11-2019

SmrtFridge: IoT-based, user interaction-driven food item & quantity sensing

Amit SHARMA
Singapore Management University, amit.2015@phdis.smu.edu.sg

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

Vengateswaran SUBRAMANIAM
Singapore Management University, vengates@smu.edu.sg

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

SmrtFridge: IoT-based, *User Interaction-Driven* Food Item & Quantity Sensing

Amit Sharma, Archan Misra,Vengateswaran Subramaniam, Youngki Lee*
Singapore Management University, Seoul National University*
{amit.2015,archanm,vengates}@smu.edu.sg,youngki.lee@gmail.com

ABSTRACT

We present *SmrtFridge*, a consumer-grade smart fridge prototype that demonstrates two key capabilities: (a) identify the individual food items that users place in or remove from a fridge, and (b) estimate the residual quantity of food items inside a refrigerated container (opaque or transparent). Notably, both of these inferences are performed unobtrusively, without requiring any explicit user action or tagging of food objects. To achieve these capabilities, *SmrtFridge* uses a novel *interaction-driven, multi-modal* sensing pipeline, where Infrared (IR) and RGB video sensing, triggered whenever a user interacts naturally with the fridge, is used to extract a foreground visual image of the food item, which is then processed by a state-of-the-art DNN classifier. Concurrently, the residual food quantity is estimated by exploiting slight thermal differences, between the empty and filled portions of the container. Experimental studies, involving 12 users interacting naturally with 19 common food items and a commodity fridge, show that *SmrtFridge* is able to (a) extract at least 75% of a food item's image in over 97% of interaction episodes, and consequently identify the individual food items with precision/recall values of ~ 85%, and (b) perform robust coarse-grained (3 level) classification of the residual food quantity with an accuracy of ~ 75%.

CCS CONCEPTS

• **Computing methodologies** → *Motion capture*; *Image segmentation*;

KEYWORDS

Internet of Things (IoT), smart fridge, IR sensor, food recognition

ACM Reference Format:
Amit Sharma, Archan Misra,Vengateswaran Subramaniam, Youngki Lee*. 2019. SmrtFridge: IoT-based, *User Interaction-Driven* Food Item & Quantity Sensing. In *The 17th ACM Conference on Embedded Networked Sensor Systems (SenSys '19), November 10–13, 2019, New York, NY, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3356250.3360028

1 INTRODUCTION

The notion of a *smart fridge*, which uses embedded sensors to track the usage and quantity of stored food items, is a staple part of the vision of "Connected Devices" or **IoT** (Internet of Things) [10]. An Internet-connected fridge that can automatically track the *identity* and *quantity* of items placed inside it (with this information subsequently being exposed via Web APIs) can enable several useful applications–such as allowing a consumer to ascertain commonly-used items that need to be replenished (when their quantity is low) or purchased (if they are no longer present in the fridge) while visiting the supermarket. Different sensing approaches have been proposed in recent years to track such food item attributes, with common approaches involving the use of per-object RFID tags [18], RGB camera images [22] and weight sensors [4]. Each of these has well-acknowledged adoption challenges–e.g., tagging individual food items is currently impractical, whereas in-fridge RGB cameras cannot perform quantity estimation (for opaque containers) and suffer from occlusion (when objects are stacked together).

In this work, we develop a prototype sensing system, called *SmrtFridge*, that uses a small number of commodity sensors to provide two novel features needed in an eventual smart fridge:

- *Interaction-Driven Capture of Food Items:* Using a combination of infra-red (IR) and optical camera sensors, it is able to automatically visually *isolate & extract* the specific brand/type of food item container that a user either places inside or removes from a fridge, without requiring any special per-object tags (e.g., RFID tags). By then feeding such extracted item images to "standard", state-of-the-art DNN-based object recognition pipelines, *SmrtFridge* can then identify the food item objects with high accuracy. *SmrtFridge's interaction-driven* paradigm, where the sensing pipeline is activated only during user-object interactions, is notable as it both (i) reduces the likelihood of visual occlusion (compared to prior approaches that focus on recognizing *stationary* items inside the fridge) by using multiple images and (ii) improves recognition accuracy by supplying the DNNs with cropped, *foreground* images of food items, using just a single camera[1].
- *Track Residual Fractional Quantity of Individual Containers:* By using such user-item interaction images captured by the appropriately positioned IR sensor, it automatically determines the approximate remaining amount of liquid/semi-solid content (relative to the container size) inside a food item container (whether transparent or *opaque*), whenever the container is re-inserted in the fridge.

SmrtFridge's uses two novel capabilities, namely natural interaction-driven image capture and residual quantity estimation, to perform sensing of *individual food item containers* (e.g., the amount of milk

[1]This is contrast to technologies, such as Amazon Go™, which reportedly use multiple store-mounted cameras & special product tags to identify individual items

SenSys '19, November 10–13, 2019, New York, NY, USA

ACM ISBN 978-1-4503-6950-3/19/11…$15.00
https://doi.org/10.1145/3356250.3360028

remaining in a milk carton) and requires no overt user action or additional object-level tagging.

Key Challenges: To build a smart fridge that uses such natural item-level interactions to identify food container items and their residual content, we must address several challenges:

- *How to Extract a Food Item:* Individual users interact with individual food item containers using a variety of different, transient gestures–e.g., an individual might hold a single milk container in the middle while removing it from the fridge, while another grasp the same object with two hands. A key challenge is to look at the sequence of image frames captured by a camera sensor, during such natural interaction episodes, and isolate/extract food item images, with an accuracy that is suitable for state-of-the-art image classifiers.
- *How to Overcome Varying Levels of Object Occlusion:* As a user extricates or inserts an individual food item into the fridge, it is quite likely that the camera's field of view will be obscured by the user's body parts (e.g., the user's hand or fingers) at different points of the motion trajectory, thereby occluding the food container object. An important research question thus is: How do we utilize a single static camera to robustly *recognize* an individual food item object and *reconstruct* its shape, even under occasional partial visibility?
- *How to Estimate the Content of a Container:* To generate proactive alerts for specific conditions (e.g., when a milk carton is becoming almost empty), we need a system that is capable of recognizing and tracking changes in the content of such containers. Past approaches (e.g., [4]) have suggested the use of weight or visual sensors, but these cannot handle opaque containers or track multiple items. Accordingly, we tackle the question: How do we identify the changes in the occupancy level of individual, potentially non-transparent, containers, across such natural user interactions?

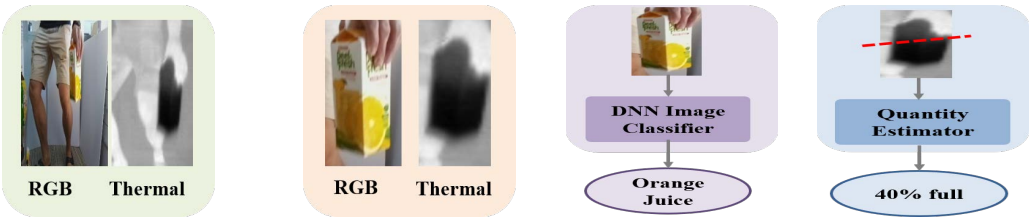
Figure 1 shows the high-level idea of *SmrtFridge*. Once the door is opened, fridge-mounted cameras capture images of user interaction (step 1). Subsequently, a combination of thermal & optical flow-based approaches is used to identify & extract the image segment corresponding to a food item (step 2). The extracted sub-image (step 3) is then fed to a Deep Neural Network (DNN) based classifier to visually identify the food item brand/type, with final item recognition based on weighted fusion of multiple images. Finally, when the user subsequently re-inserts the item back in the fridge, an ML-based pipeline operates over an IR image of the extracted food item to quantify the fraction of the container that is empty (Step 4). Note that, *SmrtFridge* currently does not attempt to (a) distinguish between multiple item instances (e.g., two identical Coke™cans), or (b) perform exact counting of food objects (such as bananas or vegetables).

Key Contributions: We shall demonstrate a practical *SmrtFridge* system that achieves our twin objectives through the 4-step process mentioned above. We make the following key contributions:

- *Dual Mode Visual Extraction of Individual Food Objects:* We demonstrate a novel segmentation technique that reliably isolates the portion of an image frame pertaining to a food item object. The segmentation technique combines two approaches: (a) a combined *IR+ visual* approach, which allows easy visual isolation of the cold part of the image (very likely corresponding to a refrigerated item); and (b) a pure visual *optical flow-based* approach, which identifies foreground food item content even when it is at the ambient (room) temperature. Real-world user studies show that, in over 97% of interaction episodes, *SmrtFridge* can extract the food item with a bounding box that contains at least 75% of the item's pixels, and achieve a median Intersection Over Union (IoU) value of 0.68 (which is higher than the 0.45-0.5 threshold required for state-of-the-art object detection frameworks [14]).
- *Accurate, Robust Object Recognition:* User studies show that a single user-item interaction episode typically lasts for 5-10 seconds, with the food item being visible in 5-10 images captured by a 30fps commodity camera. We utilize a DNN-based image recognition pipeline, which uses varying weights over this ensemble of images, to reliably identify the specific food item that is either being inserted or extracted. Experimental studies, conducted with 12 users and 19 common food items, demonstrate that *SmrtFridge* can identify the food item brand/type with 84+% precision & recall, whereas the same DNNs achieve a baseline precision of only 53% and 20% recall when supplied with the entire 'un-cropped' image[2].
- *Accurate, Robust Quantity Estimation:* We show that, across 5 different items and 3 different quantities in paper containers, the differential temperature gain rate of the container vs. the liquid results in distinct thermal zones. We show that the resulting thermal differences captured by a commodity IR sensor are discernible when the food item is placed outside the fridge for a period varying between 15 seconds to 15 minutes. By applying appropriate quantization and clustering techniques on such thermal images, we show that we can estimate the residual quantity of food items with a median and mean errors of 11% and 14% respectively (of the overall container capacity) and achieve 75% accuracy in classifying the residual quantity into three broad levels.
- *Practical* SmrtFridge *Prototype:* Using commodity sensors and embedded platforms (e.g., Raspberry PIs), we build a prototype of *SmrtFridge*, comprising 1 IR sensor, 1 camera and 1 door-contact sensor. We also empirically determine the appropriate placement of these sensors, such that they provide both good item visibility and high spatial coverage under diverse, natural, user-item interaction patterns.

We anticipate that our proposal, of using IR+visual sensing to capture user-item interactions, will strongly influence the sensor choices and sensing pipelines of future IoT-equipped smart fridges. Moreover, our core methodological innovations have applicability beyond just fridges. For instance, the approach of dynamically fusing IR+RGB sensing can be used to accurately extract segments of objects with distinct thermal signatures, e.g., (a) to automatically visually identify objects being unloaded off a refrigerated truck at a warehouse (as such object contours could be easily isolated by the IR sensor) or (b) automatically visually identify a specific component being incorrectly welded (and thus having an anomalous

[2]The development of the item recognition DNN is not the focus of our work–we expect DNN-based image recognition to improve over time



(a) Step 1: Interaction Capture (b) Step 2: Item Extraction (c) Object Recognition (d) Quantity Estimation

Figure 1: High-Level Steps in *SmrtFridge*

thermal profile) by field workers at construction sites. Similarly, IR-based sensing of thermal variations can be used for remote sensing of hot/cold objects inside containers–e.g., (a) to perform remote inspection of liquid quantities in cargo containers by simply placing them in hot/cold environments and noting resulting possible thermal variations (b) to verify the purity of unconsumed refrigerated medicines, by combining thermal based quantity estimation of such medicines with weight sensors to verify the specific density of the liquid content.

2 RELATED WORK

The related work in this field consists of both Smart Fridge-specific prototypes and systems, as well as broader work in the area of IR/visual camera-based sensing. The desired capabilities of a smart fridge are often motivated by studies on food wastage [3, 6] which found that 48% of food wastage is due to items that have passed their expiration date, with 36.5% of the cases arising from food left untended (without the user's awareness) inside a fridge.

Smart Fridge prototypes: A widely adopted approach for tracking the contents of a fridge involves the use of RFID tags attached to individual food objects. Noutchet [18] and Gu & Wang [8] propose attaching RFID tags to each product, with an RFID reader scanning each tag whenever an item is placed in or removed from the fridge. While this approach may be useful once all packaged items are universally tagged, its use at present would require extensive manual effort in labelling each object before inserting in the fridge and removing such tags before eventual discard. The Pervasive Fridge system [20] also envisions a system that tracks food items and their expiry dates (a feature that has been reported [5] to be extremely desired by users) via a manual process that uses multiple modalities (barcode scanning, audio input or text input) to explicitly add items and their attributes (e.g., expiry dates).

The CloudFridge prototype [22] is one of the first systems to build and evaluate a sensor-based prototype to track user-fridge interactions in real-time and retrieve current and historical states of food items. Similar to our approach, CloudFridge applies video-based recognition to identify individual food items and uses multiple additional proximity (IR) sensors to keep track of each item's location inside the fridge. Evaluations performed using full-frontal images of the objects achieve precision values of ≈ 70%. However, CloudFridge does not directly address the real-world problems of extracting food item sub-images from videos of real-world human interaction or of estimating the residual food amount in a container. Along similar lines, the *PerFridge* system [17] augments a refrigerator with various sensors such as proximity (IR) and magnetic sensors to track various forms of 'wasteful' behavior, such as leaving a fridge door open for an excessively long duration or stacking multiple items in one corner (resulting in improper air flow). *PerFridge* does not, however, automatically identify food items or their residual quantity, relying instead on a touch-screen interface for explicit human input.

Several commercial smart fridge products have also recently been announced. An example is the *Liebherr* smart fridge [2], which uses an interior-mounted camera to classify food items inside the fridge and a voice recognition system to process voice commands (such as ordering food items). At present, we are, however, unable to quantify the performance of such commercial systems under densely-packed, occluded scenarios.

Analysis of Food & Other Content: There has been a variety of innovative work, employing different sensing modalities (e.g., visual, weight and RF), to infer various attributes of container-based food items. In the most recent and relevant work, Jiang et al. [11] employ a CNN (convolutional neural network) based approach to estimate four discrete levels of content inside a glass or *transparent* bottle. Although the CNN is trained with various coloured plastic/glass bottles, a purely visual sensing approach does not work for non-transparent containers, e.g. paper cartons. A while back, Chi et al. [4] had demonstrated a method for estimating the type of food ingredients and their quantity using a combination of weight sensors and camera-based identification of ingredients (on a specially instrumented countertop). More recently, Wang et al. [24] has shown how phase/RSSI information from RFID tags mounted on containers can help distinguish between different liquids in containers with accuracy as high as 94%. We believe that our use of an IR sensor to identify the quantity of liquids/semi-solids inside a container is a novel approach that exploits the temperature differential between a fridge's interior and its ambient surroundings. IR-based thermal tracking has been proposed in [19] to monitor the quality of vacuum-packed food containers. This approach, however, monitors the *whole* container and does not attempt to use thermal variations for residual quantity estimation.

Visual Analysis of Food Items: A different line of work has explored the use of automated techniques to identify food items based on image analysis. Nutrinet [16] applied a CNN-based approach to recognize 520 commonplace food and drink items, typically captured by a smartphone camera, with an accuracy of ≈ 87%, whereas Kagaya et al. [13] previously demonstrated how CNNs provided better food recognition accuracy (using a public food blogging dataset) than shallow classifiers, such as SVMs. More recently, the Annapurna system [23] addressed the problem of identifying and

extracting images of plated food items captured by a smartwatch-embedded camera. Most such approaches are based on close-up photos of food content that is assumed to constitute the foreground. In *SmrtFridge*, we explicitly tackle this challenge of extracting out the food object from images of natural human interaction, captured by a fridge-embedded camera.

3 MOTIVATING SCENARIO & DESIGN GOALS

To motivate the capabilities of *SmrtFridge*, we envision a Smart Fridge operating as follows:

- To prepare her breakfast, Alice opens the fridge and grabs a juice carton, which she then proceeds to pour into a glass on the kitchen countertop. *During this operation,* SmrtFridge *is triggered when Alice is retrieving her juice carton and infers the retrieved item: Juice Carton Product A.*
- Subsequently, Alice reaches into the fridge and grabs two pouches of yoghurt, which she then empties into her breakfast bowl. *As before,* SmrtFridge *should be able to track the new food items that Alice has retrieved—2 pouches of Yogurt Product B.* She proceeds to mix her cereal into the breakfast bowl, for about 1-2 minutes.
- At this point, Alice places the juice carton back in the fridge. SmrtFridge *monitors this act of inserting a food item, identifies that the item is Juice Carton Product A, and also estimates that the carton is now only 25% full.* (This quantity estimation can be transmitted to a back-end portal, which can asynchronously trigger relevant actions–e.g., generating a 'Low Juice' alert.)
- Finally, Alice also inserts a can of her favourite beverage in the fridge, before closing the fridge door. SmrtFridge *tracks this object insertion, identifying the objects as "can of Beverage Product C", and thereby updates the repository of the fridge's content.*

It is important to note that this entire workflow is based on a user's *natural* interactions with the fridge: at no point is Alice required to perform any specific additional action (e.g., scanning an item's barcode on a reader, tagging an item, annotating an image) to aid *SmrtFridge's* operation. While labels are needed to train image-based item recognizer, this can be performed a-priori e.g., by external companies that survey available food products.

3.1 Design Goals

The observations above drive the following *SmrtFridge* design goals.

- *Identification of Product Labels: SmrtFridge* must be able to identify and label the individual food products with which a user interacts. Generic item-agnostic alerts would be of the form 'item was extracted from the fridge at time *t*' and are useful only for tracking fridge usage. In contrast, an alert of the form 'Juice Product X, with approx. 40% content remaining, has been in your fridge for the past two weeks' provides a user targeted, actionable feedback.
- *No Additional Human Effort:* To support unobtrusive tracking, *SmrtFridge* must not require the user to perform any additional actions, beyond what she presently does with a conventional fridge. This implies that *SmrtFridge* cannot employ approaches such as barcode scanners [20] or manually-entered product logging [15] to obtain additional insights. Moreover, *SmrtFridge's* image-based item recognition pipelines must work in-the-wild, i.e., with images of items that are not necessarily centred or placed vertically.
- *Need to Estimate Residual Amount in a Container:* Past studies [12] have shown that users who are aware of the amount of unconsumed food items in their fridge make less wasteful consumption decisions. To support such insights, *SmrtFridge* must be able to estimate the fraction of remaining liquids inside specific containers. From a practical perspective, it is imperative to perform such content estimation when the user is inserting an item back into the fridge (as the user would have typically consumed some fraction of the existing content) so that the user can subsequently track (without inspecting the refrigerator) the residual quantity of food.

3.2 *SmrtFridge*: Out of Scope

As consumers can certainly desire additional capabilities from a smart fridge, we establish upfront the functions that *SmrtFridge does not* currently support. Very specifically:

- *No Product Expiration: SmrtFridge* does not have any notion of detecting the possible expiration dates of individual food items. While *SmrtFridge* can provide the duration for which an item has been residing in the fridge (and a rule-based back-end may trigger alerts when a specified period has been exceeded), more precise expiration tracking will require coupling our mechanisms with alternative approaches (e.g., OCR-based parsing of expiration dates on containers).
- *No Support for Unlabeled Food Items: SmrtFridge's* operational logic is based on extracting visual images of a food container or discrete food items (e.g., fruits), and then performing DNN-based recognition of the product. Accordingly, *SmrtFridge* cannot presently support recognition of unlabeled food items (e.g., home cooked foods such as salads or curries), although future versions can integrate ongoing deep learning work on recognition of cooked foods (e.g., FoodAI [1]).
- *Approximate Support for Quantity Estimation: SmrtFridge's* IR sensing techniques help to provide coarse-level estimates of residual food quantity in containers (e.g., less than 25% remaining). However, *SmrtFridge* does not aim to measure such food quantity precisely (e.g., 30 mg of juice). It is likely that the addition of high-resolution weight sensors might enable more precise quantity estimation of food items.
- *No Tracking of Specific Item Instances: SmrtFridge's* visual sensing effectively recognizes specific food types or brands (e.g., a can of *Coke*), rather than specific individual item instances.

4 SMRTFRIDGE SYSTEM OVERVIEW

As mentioned before, *SmrtFridge's* key novelties (compared to prior work) are in developing processing pipelines to (a) *extract* a food item's sub-image (a bounding box) from individual RGB image frames (so that it can then be recognized using state-of-the-art DNNs), and (b) *estimate* the residual food quantity in such food containers. Before detailing these key functions in Sections 5 and 6 respectively, we first present the overall functioning of *SmrtFridge*,

as illustrated in Figure 2. *SmrtFridge's* sensing substrate includes: (a) an RGB camera that visually captures the item-level interactions that an individual performs with the fridge; (b) an infrared (IR) camera that is used to both aid in food item extraction and quantity estimation; and (c) a magnetic reed switch, attached to the door, which detects the opening and closing of the fridge door (and thus triggers the sensing pipeline).

The *SmrtFridge* workflow consists of the following steps:

(1) *Episode Segmentation:* The door contact sensor helps to identify the *start* and *end* of a single interaction *episode*–an episode may involve the user retrieving or inserting one or more (and possibly even none) items from/in the fridge. This sensor acts as a trigger to the IR and RGB camera pipelines–these cameras start capturing images whenever the door is open until the user closes the door subsequently.
(2) *Item Image Extraction:* This process concurrently executes two different pipelines. The first pipeline uses only the visual (RGB) camera data to first compute object motion vectors, followed by clustering and thresholding of such vectors to extract the image of the food item. The second pipeline uses the thermal (IR) camera to obtain the relevant coordinates of objects that are significantly colder than the ambient thermal values, and then obtain the image of the food item by extracting the *corresponding* coordinates in the RGB camera.
(3) *Image-based Food Item Recognition:* The extracted RGB images (or sequence of images), ideally corresponding to a food item, is then passed through a CNN-based recognizer. The CNN is pre-trained by an external entity (e.g., an image analytics company) with a, preferably large, corpus of representative images of various food items. For each image frame, the CNN then outputs the likely label (along with the confidence values). Because the item-specific user interaction (within an episode) lasts for several seconds, the extraction process retrieves a sequence of multiple (typically 30-40) images, of which 5-10 contain the food item. This series of CNN output labels are then further fed through a classifier that uses the frequency of occurrence and associated confidence levels to output the food item label with the *highest likelihood, above a minimum threshold.*
(4) *Residual Food Quantity Estimation:* In parallel to the above process, IR images are also fed through a quantity estimation pipeline. This pipeline works on the principle of *differential heating of the container vs. the food content inside* and is thus triggered only when the user is *inserting* items into the fridge. (Such a temperature differential is absent when the user is taking out a currently refrigerated item.) The extracted food item's IR image is fed through an unsupervised classifier that demarcates the container pixels into two spatially contiguous clusters. The partial area of the *colder cluster* (corresponding to the non-empty portion of the container), relative to the area of the overall container, is then used to estimate the residual food quantity (by volume percentage).
(5) *Additional Workflow Steps:* Once we have determined the food item and its remaining quantity, *SmrtFridge* can appropriately update a repository of refrigerated food contents. Similar to prior work, such changes ('bottle of product A, 30% full' inserted) may be pushed to a Web server, which can be instrumented to generate relevant alerts (e.g., "send an SMS if a container with residual quantity ≤20% has been sitting in the fridge without any user interaction for more than a week"). Note that such a Web back-end has not been implemented in the present *SmrtFridge* prototype.

5 VISUAL IDENTIFICATION OF FOOD ITEM

SmrtFridge's process for extracting food items includes two alternative pipelines (one purely using visual images vs. another fusing IR and visual images), coupled with a CNN-based item classifier.

5.1 IR-driven Image Extraction

In this relatively more straightforward approach, we utilize the insight that a refrigerated item will typically be *much colder* than either the interacting human's body or the ambient temperature. Accordingly, an IR camera should be able to easily isolate such a cold item, as the food item's pixels will be much darker than other ambient objects. Accordingly, we attempt to use a *pixel intensity-based segmentation* approach. In this approach, during an ongoing interaction episode, we extract the cold item from the thermal image (a frame with timestamp t) by selecting all pixels below a threshold value Th_{temp}. We compute the Cartesian coordinates of all the selected pixels, thus segmenting the cold item from the thermal image. We then calculate a bounding box (i.e., the smallest rectangular region that *contains* the entire contour area) to represent the segmented object.

Once the item's bounding box in the IR camera's coordinates is identified, we utilize the fact that both the IR and RGB cameras *concurrently* and continuously record images/videos, albeit with different *FoV* (field of view), during an interaction episode. Because the two cameras are fixed, we transform the IR camera's coordinates into the RGB camera's frame of reference using an a-priori computed *transformation matrix*. However, empirically (because our Raspberry Pi-based implementation does not support a real-time OS), we observe that the frames of the two cameras are not always synchronized. Accordingly, we select all the RGB frames that have a timestamp $(t - \Delta, t + \Delta)$, where Δ represents the time offset: for each of these frames, we extract the sub-image corresponding to the (transformed) RGB bounding box coordinates. Separately, the optical flow approach (Section 5.2), applied to selected RGB frames $(t \in \{t - \Delta, t + \Delta\})$, provides another set of candidate images. Each of these "potential food item" images (two per frame) is then sent to the downstream item recognition DNN classifier.

5.2 Purely Visual Extraction

The IR-driven approach is not applicable when a food item's temperature matches the ambient room temperature–e.g., an item bought from a grocery store is being inserted for the first time. To provide an alternative means of food item tracking under this broader set of conditions, we utilize the fact that a user's interaction with a food item (either removing or inserting into the fridge) involves a directional motion either away from or towards a fridge-mounted camera. The approach, illustrated in Figure 3 first applies the principal of *optical flow* to identify the image segments that are moving (across consecutive frames), thereby eliminating the parts of the

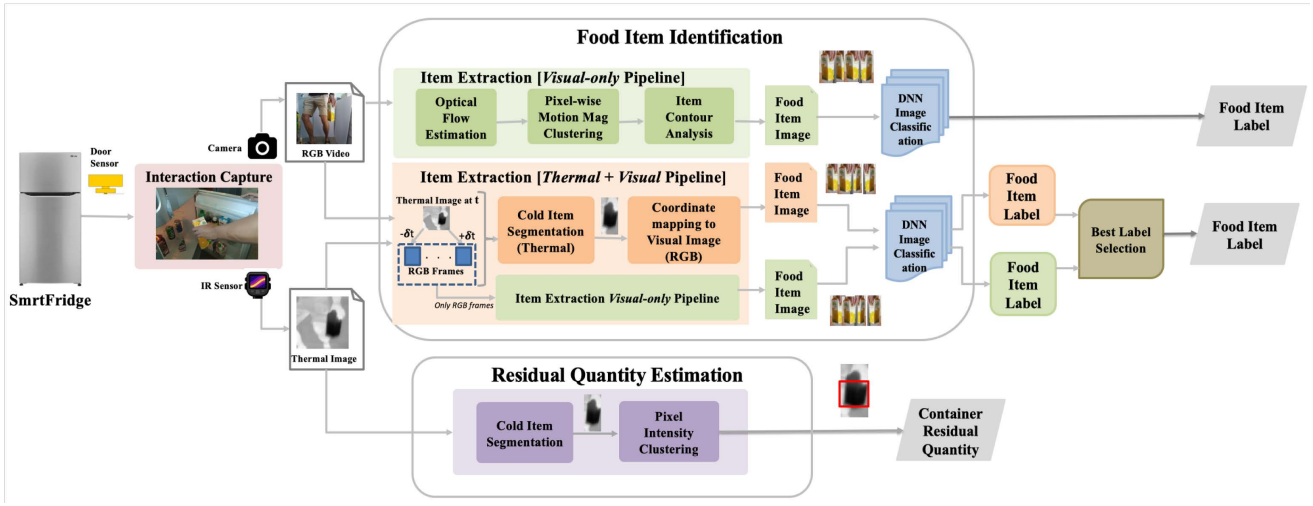


Figure 2: Overview of *SmrtFridge's* Workflow.

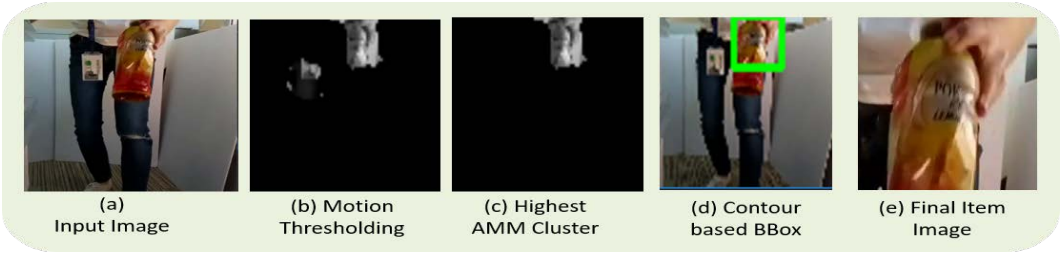


Figure 3: Visual Item Extraction Steps

image that correspond to a static background. Such optical flow estimation identifies motion vectors (direction and displacement magnitude) for each pixel in an image.

We then identify the parts of an image with *significant movement*–i.e., with motion magnitude higher than a minimum threshold Th_M. The resulting pixels (Figure 3(b)) are likely to contain the food item, as well as other moving objects captured in the camera's field of view (FoV), such as the user's limbs, the moving fridge door and even background movement (e.g., an animal moving in the background). The static background portions of the image (e.g., parts of the fridge door) are first removed through standard background subtraction techniques. To then isolate the food item from additional movements, we first employ spatial clustering. More precisely, we create a feature vector where each pixel's feature consists of its coordinates, as well the magnitude and direction of its motion vector–i.e., {x,y, motion-mag, motion-dir}. We employ the K-Means clustering technique to cluster the pixels into distinct, spatially disjoint, *motion clusters*, and then pick the cluster with the highest average motion magnitude (AMM) value (Figure 3(c)). This is based on our intuition that the food item of interest is usually *the moving object closest to the camera*, and thus extremely likely to have the largest displacement magnitude from the camera's perspective.

The resulting cluster (Figure 3(c)) consists of both the food item, as well as possibly additional background pixels. To better isolate the image segment corresponding to the food object, we then execute the *Canny* edge detection algorithm, followed by standard morphological operations (e.g., erosion and dilation) to help connect some of the disconnected edges. The resulting edges are then passed through a contour detection algorithm to obtain an outline of the food item, before fitting a bounding box (Figure 3(d)) over this contour to represent the image. As this bounding box image is from a scaled-down version of the initial RGB frame (the down-scaling was initially performed to speed up the computation), we finally scale-up the bounding box coordinates, using template matching, to select the high-resolution sub-image (Figure 3(e)) that represents the extracted food item. As before, each such 'food item' is sent to the downstream item recognition DNN classifier.

5.3 The Food Item Recognition Process

Given the extracted item, we then use a well-known CNN-based deep learning classifier, the *ResNet v2 (152 layers)* [9] to classify the food item. Note that (see Figure 2) that this classifier receives *multiple possible* food item images. Specifically, the combined IR-Visual pipeline provides one coordinate-transformed image for each RGB frame with a timestamp within Δ of an IR frame, whereas the Motion-thresholding approach provides an image for every RGB frame with a foreground cluster exceeding the motion threshold.

The item recognition process consists of the following steps:

(1) Given K different classes of food items, we first train a multi-class CNN classifier that outputs $K+1$ labels: each of the K food items + a *null* class (corresponding to a 'non-food' classification).
(2) During the test phase, each interaction involves a sequence of say S image frames, provided by both the IR+MV and MV-only methods. Each frame was then individually passed through the classifier, generating a probability/confidence value for each of the K+1 labels. Let $p_i^k \;\; k = \{1, \ldots, K+1\}, \;\; i = \{1, \ldots, S\}$ represent the probability of the k^{th} class for the i^{th} frame.
(3) For each such frame, if the highest likelihood class is $K+1$ (the non-food or background class), then we discard the corresponding frame (this occurred ~50% of the time).
(4) For the remaining L frames, we compute the cumulative likelihood of each of the K food item classes using the **FREQ-CONF** method: for each class, we compute the frequency of identification, as well as the sum of confidence values (across L frames) within the episode, and then select the most-frequent class that has the highest frequency probability/likelihood across the L frames.
(5) Finally, we select the food item label that has the highest cumulative likelihood value across all the frames. An alternative strategy of just using the classification output from a single 'randomly-selected' frame may reduce the energy consumption but has much lower accuracy (Section 8.3).

6 RESIDUAL FOOD QUANTITY ESTIMATION

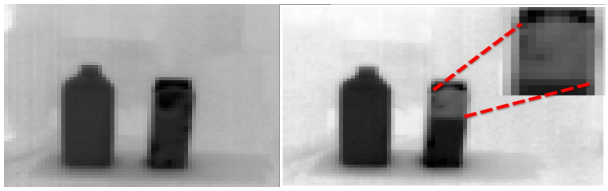
Besides identifying the food item removed or replaced, *SmrtFridge* also quantifies (at a coarse granularity) the quantity of residual food inside the identified container. Quantifying such content is vital for several possible applications e.g. informing users if the quantity of juice in a container falls below a minimum threshold (e.g., 20%), or if a close-to-full container has been lying inside the fridge for a very long duration. For our exposition, we estimate quantity as a fraction of the container volume e.g., if a 1000 ml juice container presently has $\frac{3}{5}^{th}$ (600 ml) of juice remaining, the quantity should be ideally estimated to be 60%.

6.1 The IR-based Approach

SmrtFridge uses a non-intrusive quantity estimation technique that is both robust to different ambient lighting conditions and the *opaqueness* of the food container. In this technique, an inexpensive relatively low-resolution IR camera is used to record and extract the food item's *thermal profile*, when the user is re-inserting an item back inside the fridge. The technique is motivated by a fundamental observation on *differential specific heat properties* of a container and the item that it contains. In particular, whenever a currently refrigerated food item is removed and placed outside, its temperature will start to increase as it absorbs ambient heat (assuming room temperature is higher than the item temperature). For a full container, all parts of the container (containing the solid or liquid food item) will gain heat at a similar rate, whereas in a partially filled container, there will be a difference between the rates at which the empty & filled portions of a container warm. Table 1 lists the specific heat capacity of some of the common liquid/solid food items and typical container material. In general, we see that the food items have significantly higher specific heat than typical container material: intuitively, the part of the container in direct contact with the food item (liquid or solid) will share its acquired heat conductivity with the item, and thus become cooler than the empty portion (which will heat faster). Moreover, the *larger the specific heat of the food item*, the higher the difference between itself and the container and thus the larger the expected differential between the thermal intensity of the empty vs. filled parts of the container.

Food Item		Container Material	
Juice	3.4	Plastic	0.4
Milk	3.93	Glass	0.2
Water	4.18	Paper	0.33
Yogurt	3.52	Air	0.718 (C_v)

Table 1: Specific Heat of Substances (KJ/kg/ C)



(a) Cold items at t=0 **(b) Cold items at t = 20 secs**

Figure 4: Thermal Intensity Differential after 20 seconds

Our hypothesis is that the thermal camera can utilize this temperature difference to estimate the remaining quantity inside the container. Such differentiation will, of course, depend on the thermal resolution of the IR camera; we found that commodity cameras (e.g., the Raspberry PI compatible Bricklet camera[3]) typically have resolution of 0.1°C or lower. As an illustration of this hypothesis, Figure 4 shows two thermal images, each containing two cold containers (the left one being full and the right one partially filled). Image 4a is a thermal image when both the containers were just taken out from the refrigerator ($t = 0$) whereas Image 4b shows the thermal image of the same containers after they were kept outside for $t = 20$ seconds. We see that the thermal image of the partially filled container shows two regions of different pixel intensities, with the empty region having higher temperature values (less dark pixels) and the 'filled' region having lower temperature values (darker pixels). We shall leverage this difference in pixel intensities to estimate the size of the empty portion of the container and thereby derive the quantity of the food item inside the container.

6.2 Processing Pipeline

Figure 5a shows the thermal camera installed inside our test refrigerator, while Figure 5b shows two representative thermal images captured when two different food items are being kept inside the refrigerator. After the thermal images are captured, each image is passed through the following processing pipeline (illustrated in Figure 6):

[3]https://www.tinkerforge.com/en/shop/thermal-imaging-bricklet.html

(a) Camera setup inside the fridge (b) Sample thermal images taken by thermal camera

Figure 5: Thermal Camera Sensing

- *Partial Capture Check*: Due to the continuous capture of images during the user-item interaction episode, the thermal camera will generate multiple images of the food container. Because of the underlying motion dynamics, some images will capture the item only partially, while others will obtain a larger, clearer view. To eliminate *partial captured* images (which can be ignored for estimating quantity), we check to see if the container's contour intersects with the boundary of the captured image. If so, the container has likely been captured only partially; we thus discard the image.
- *Cold Item Segmentation & Noise Removal*: Given the thermal image, we follow the pixel intensity based segmentation steps outlined in Section 5.1 to extract the image segment corresponding to the food item container, which may contain additional extraneous pixels (often due to heat leakage around the cold item container, whereby pixels that are *near* the cold container have an intermediate temperature value that is lower than the ambient temperature). To remove these neighboring intermediate pixels, we use clustering and contour detection. First, we cluster all the segmented cold points into two clusters, one containing the intermediate neighborhood pixels (and empty part of container) and another containing the "filled-part" of the container itself. Second, we find contours from both the clusters, labeling the contour with the larger perimeter value as the *outer contour* (this contains all the neighborhood intermediate cells) and the other as the *inner contour*. To selectively discard only the neighborhood pixels, we first obtain the top-most point (highest y coordinate) of the inner contour. Because the empty part of the container is always *above* the filled portion (due to gravity), we then discard those pixels from the outer contour that lie below this top-most point (i.e., have smaller y coordinates) and combine the remaining pixels (which we anticipate to correspond to the empty portion of the container) with the pixels of the inner contour to obtain the *container's contour*.
- *Occluded pixels:* Depending on the interaction pattern, some part of the container can be occluded by the user's hand. This occlusion is also evident (as high brightness pixels) in the thermal image, and can cause an under-estimation of the container volume. To overcome this occlusion, we use an interpolation strategy, where we first extend the detected contour to a more regular (often rectangular) shape. The occluded pixels within this extended contour are then given an *estimated* thermal value, computed as the median of the neighboring non-occluded pixels.

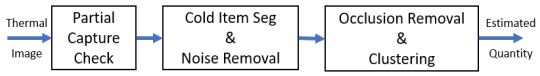


Figure 6: Quantity Estimation Processing Pipeline

- *Clustering*: Finally, we apply clustering on the pixel values of the extended container contour obtained from the previous step. Intuitively, if the item container is full, then there should only be a single cluster, whereas a partially filled item should be separable into two clusters. We use the Silhouette Coefficient method [21] to resolve between these two alternatives. If the number of preferred clusters is 2, we compute the fractional quantity of the food item by dividing the pixel count of the "food item" (lower temperature) cluster by the total pixels in both the filled and empty clusters.
- *Averaging:* Finally, given multiple valid images for a given interaction episode, the final quantity estimate is obtained by averaging the fractional estimates of each image.

6.3 Controlled Study & Validation

We performed *preliminary controlled studies* using the *SmrtFridge* prototype (which we shall describe next in Section 7) to understand the basic feasibility of this IR-based quantity estimation process. In particular, we experimented with a paper container that was filled to 60% of its capacity with 3 different liquids and initially placed inside the fridge. The container was then brought out of the fridge and placed outside for a variable duration, before being re-inserted into the fridge. The IR-based quantity estimation technique was then applied to the images captured during the user's interaction during this re-insertion phase. We studied two distinct questions:

- *How does estimation accuracy vary with different food items?* To address this question, we experimented with 3 distinct liquids {*juice, milk, water*} placed inside the container.
- *How long does a container need to placed outside for the thermal differentiation to be discernible?* Intuitively, if this ambient exposure time is too short, the thermal difference would be too negligible to permit proper clustering; conversely, if the duration was too long, then both the empty and filled portions of the container would reach (or be close) to the ambient temperature and be indistinguishable. To address this question, each of the 3 liquids was placed outside the fridge for a duration T_a that varied between {0,5,15,30,60,90,150,200,450, 800,1100,1800} seconds.

Figure 7 plots the estimation error for all 3 liquids, as a function of the ambient exposure duration T_a. We see that:

- The quantity estimation error is typically less than 15-20% for all liquids, indicating *our IR-based approach provides good* ***coarse-grained*** *quantity discrimination capability.*
- This error is relatively insensitive to the ambient duration (T_a), as long as this duration varies between 5 secs–15 minutes. (The vast majority of daily user interactions with refrigerated items should involve keeping the item outside for at least 5 secs, and no more than 15 minutes.) Our results thus suggest that *our IR-based approach is applicable to a very wide variety of user interaction patterns, even though its accuracy*

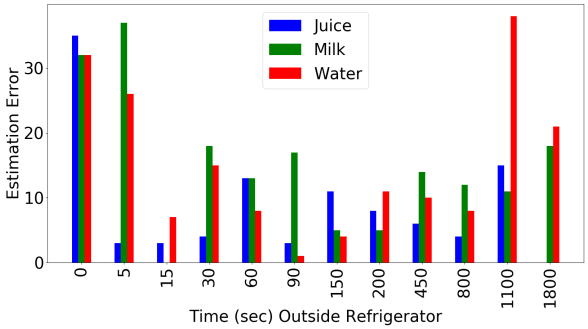


Figure 7: Difference in Estimated and Actual Quantity (%) wrt Item Exposure Time outside Refrigerator

would degrade if a container was left outside too briefly (<5 secs) or for too long (>20 mins).

7 SMRTFRIDGE PROTOTYPE

To empirically demonstrate the feasibility of our techniques for IR+visual based food item identification and IR-based quantity estimation, we have built and tested a *SmrtFridge* prototype. The prototype (costing less than USD 300) was built using a commodity fridge (Toshiba GR-R20STE, double door with 184L capacity), with the following sensors controlled by a Raspberry Pi 3 model B:

- *Visible Light Camera sensor*: Raspberry Pi camera module V2
- *IR/Thermal Camera sensor:* Thermal imaging bricklet [4]. One important property is that the IR sensor has a relatively low resolution (80 by 60 pixels). While higher-resolution IR sensors might offer better accuracy, they were significantly more expensive.
- *Door Contact sensor*: Normally open magnetic reed switch.

7.1 Placement of Sensors

One of the important empirically-determined choices relates to the placement of the sensors. In particular, the IR and RGB camera sensors need to be appropriately positioned to support multiple concurrent objectives: (a) *maximize gesture coverage*–i.e., support the video based capture of user-item interactions performed in a variety of ways, across different shelves of the fridge; (b) *minimize occlusion*–i.e., ensure that the food item is maximally visible within individual frames (to aid proper computation of the residual quantity); (c) *maximize visible frames*–slightly different from the above objectives, the goal here is to have the item be visible in the maximum number of possible frames (to maximize the chances of correct food item classification).

We empirically experimented with various positions, of which the three most choices are illustrated in Figure 8. (The figure also shows sample images captured from each of these positions.) Note also that we explicitly chose positions where the sensors were an integrated part of the fridge frame/body–accordingly, we did not consider choices that involved placing the sensors externally.

[4]https://www.tinkerforge.com/en/shop/thermal-imaging-bricklet.html

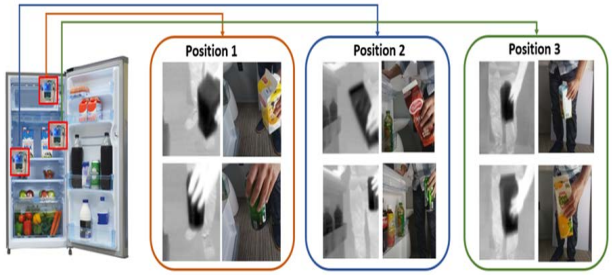


Figure 8: Potential camera deployment positions. Position 3 is preferred due to {greater coverage, lower occlusion}.

On analyzing sample video frames (obtained from our controlled studies) we observed the following characteristics:

- *Position 1*: Here the camera (IR+ RGB) sensors are installed on top of the refrigerator, thus providing a top view of the items while they are being added/removed from the fridge. Although this view is likely to capture most of the item interactions, it is often unable to capture the height of the containers properly (see Figure 8) especially when the containers are picked from the lower racks, leading to lower accuracy of quantity estimation.
- *Position 2*: Here the thermal and visible light camera sensors are deployed on the left side (closer to the door) of the refrigerator. In this case, the captured items often include items kept in the trays mounted on the fridge door. While such images can possibly be eliminated by optical flow techniques, the presence of such cold items is likely to increase the error of the thermal segmentation process.
- *Position 3*: This is the case when both thermal and visible light camera sensors are deployed on the right side (away from the door hinge) of the refrigerator. From our sample observations, we found that the vast majority of interactions (across a variety of 'removing' or 'inserting' gestural patterns) were visible with this placement, with the camera's field-of-view (FoV) primarily capturing the user-item interactions. Furthermore, occlusion of the food items was also very rare. Accordingly, we have used Position 3 as the preferred placement in our prototype.

We believe that, while these observational insights can benefit future fridge design, additional model-specific studies would be needed to determine optimal placement in other scenarios (e.g., single vs. double door fridges).

7.2 Object Recognition DNN

To identify the food item objects, we utilize the well-known *ResNet v2* Model (152 layers) with pre-trained ImageNet weights. The classifier is trained, using *TensorFlow* on an Intel Core i7-7700 CPU @ 3.60GHz with 64GB RAM & NVIDIA GeForce GTX 1080 Ti GPU, The classifier had 19(objects) + 1(background) class and 2000 images per class. To generate the training set (in a commercial setting, such training data could be crowd-sourced directly from food manufacturers), we (a) used a camera to record videos of the food items under different conditions, such as varying zoom levels, object rotation, background lighting and occlusion levels; and (b)

S.No	Parameter	Values
1	Liquid Types	Juice, Milk, Water
2	Content Quantity	100%, 60%, 30%
3	Container Material	Paper
4	Time Outside Refrigerator	20 Seconds

Table 2: Quasi-Controlled Study Specs (Quantity Estimation)

downloaded corresponding Web images using Google's Custom Search engine. Also, for the 'null' (background) class, we shot videos of various indoor lab settings. From this dataset, we utilized 80% for training, 10% for validation and 10% testing, achieving a test accuracy of 97.6%. Our training dataset didn't include *in-fridge* videos of any item.

8 PERFORMANCE ANALYSIS

We now study the real-world performance of the various components of the *SmrtFridge* system.

8.1 Data Collection & User Studies

Our results are based on two separate studies:

- **Naturalistic User Study:** In this study, conducted with an explicit institutional IRB approval, 12 different users (members of the general public) initially performed natural fridge-based interactions with 15 different & common food items– e.g., chocolate milk, orange juice, guava juice, etc. Users were asked to insert and remove such items from the fridge multiple times, without any restriction on how long the item could remain outside. In a subsequent phase, 7 new users participated in an expanded study, which included 4 additional fruit & vegetable items (oranges, broccoli, green peppers, eggplant). This user study is used principally to study the efficacy of the *item identification* process.
- **Quasi-Controlled Micro Study:** The goal of this separate study (detailed in Table 2) was to ascertain the accuracy of *item quantity estimation*, under varying quantity levels, different vertical angles and for different liquid food items. In this study, 7 users perform *natural-like* interactions with different items, but with explicit instructions on (a) the items to be kept inside or removed from the fridge and (b) how long the items were kept outside (the ambient exposure time).

8.2 Item Extraction

We first evaluate the performance of *SmrtFridge's* item extraction pipelines. We use two principal metrics:

(a) Intersection Over Union (*IoU*), which evaluates the relative overlap between the (manually annotated) ground-truth bounding box of the item (BB_{GT})and the bounding box (BB_{Est})computed by the automated *SmrtFridge* pipeline. It is computed as $\frac{BB_{GT} \cap BB_{Est}}{BB_{GT} \cup BB_{Est}}$.

(b) Item Coverage *ICov* (= $\frac{BB_{GT} \cap BB_{Est}}{BB_{GT}}$), which computes the ratio of the intersection area of the ground-truth and computed bounding boxes to the ground-truth bounding box.

Figure 9 plots the fraction of extracted images (across all episodes in the user study) whose IoU exceeds the specified threshold. We compute the IoU scores separately when using (a) just the RGB motion vector pipeline (mv_only), (b) just the IR-driven mapping to RGB coordinates (ir_driven _mv(thermal)) and (c) the proposed IR-driven motion vector (ir_driven _mv) pipeline that utilizes the best of both pipelines. We see that the combined approach provides the best extraction performance: over 80% of images have IoU values greater than 0.6 (object detection frameworks typically require IoU values higher than 0.45-0.5). In contrast, the pure RGB motion vector-based approach performs the poorest, achieving IoU values greater than 0.6 in less than 20% of the images.

Pipeline	*ICov* >=**95%**	*ICov* >=**75%**
MV only	82.4	97.3
IR assisted MV	83.3	97

Table 3: Percentage of Episodes vs. *ICov*

Approach	15 class Classifier		19 class Classifier	
	Precision	**Recall**	**Precision**	**Recall**
Motion Vector (MV) Only	0.82	0.79	0.83	0.81
IR driven MV	0.80	0.78	0.81	0.79
MV+IR Merged Pipelines	0.83	0.83	0.84	0.84

Table 4: DNN-based Item Identification accuracy (per Episode)

To further understand the importance of high IoU values (i.e., ensuring that the extracted image faithfully captures the food item), Figure 10 plots the precision/recall values for *DNN-based item identification* for those images whose IoU value exceeds the corresponding x-axis value. We observe that the item identification accuracy increases with IoU, reaching 95+% when the IoU value exceeds 0.7.

Figure 11 plots the distribution of *ICov* values, for both the combined and the RGB motion-vector only methods. We see that the combined technique achieves *ICov* values of 0.8 or higher in 80% of the interaction episodes. The higher *ICov* values observed for the "RGB motion-vector only" occur because this approach typically extracts a larger fraction of the image but also includes a disproportionately larger 'background' component (hence, the lower IoU score). As we shall show in Section 8.3, the presence of a larger background leads to poorer performance of the DNN-based item identifier. To further illustrate the preciseness of *SmrtFridge's* item extraction process, Table 3 quantifies the number of episodes (out of a randomly selected 20% of the total episodes) that contain at least 1 extracted item image with *ICov* values higher than {75%,95%}.

8.3 Item Identification

We now study item identification accuracy, based on the extracted images (from an initial study with a 15-item classifier & 12 individuals, followed by a 19-item classifier with additional fruit & vegetable items & 7 users), achieved by our ResNet-based DNN.

Table 4 plots the item classification results (for episodes involving the original 12 users who interacted with the original 15 food item classes), for both the 15-class classifier and the subsequent 19-class classifier. We see that the combined pipeline results in the highest and identical precision/recall values (of ~0.84). Moreover, the results are fairly stable over the 15-class and 19-class classifiers. As a point of comparison, the food item precision/recall is 74% and

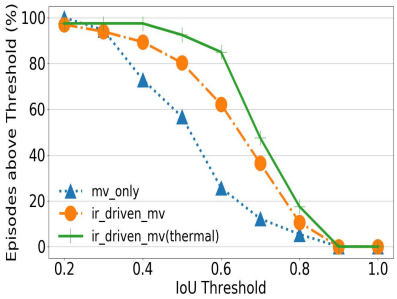


Figure 9: Relationship between IoU scores and percent of episodes above it

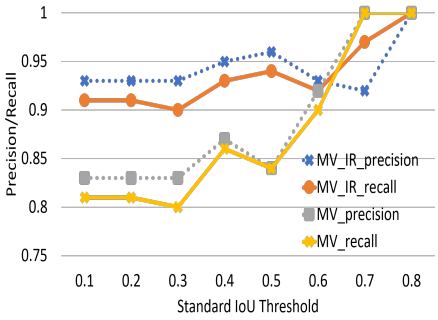


Figure 10: IoU score vs image identification precision recall

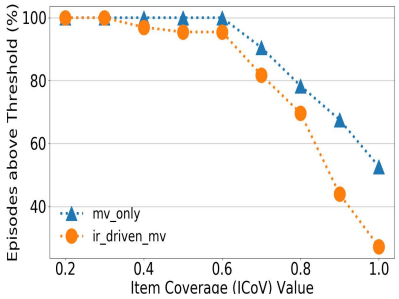


Figure 11: % of Episodes vs. Minimum *ICov*

72% respectively, for the episodes involving the 7 new users, who interacted solely with the 4 new fruits & vegetable items.

We observe that:

- The overall item recognition accuracy is high but not as high as the 97%+ accuracy reported on the externally curated training data. In large part, this is due to the lack of sufficient *relevant* training data for our classifiers. In particular, the training corpus consists entirely of images of items extracted from the Web or shot in close proximity by a video camera. These training images are quite distinct from the partial views of items captured by the *SmrtFridge* RGB+IR sensors. We fully anticipate that the accuracy will improve as the corpus is continuously expanded in the real world (similar to approaches used by consumer ML-based devices such as Amazon's Alexa™) to include more such *in-the-wild* images.
- The accuracy is lower for the newer episodes that involved the 4 new food items. This was principally due to the lack of sufficient *appropriate* training images—unlike canned items, fruits and vegetables have greater diversity in shape and color, and thus require more diverse training data.

Alternative Classification Strategies: To further underline the importance of accurate sub-image extraction, we computed the accuracy of a baseline where the DNN classifier operated on full-HD images (containing both the food item and miscellaneous background content). The DNN classifier then performed very poorly, achieving precision and recall values of only 0.53 and 0.20. Similarly, if the classification is performed only on 1 extracted image (as opposed using the highest cumulative likelihood across all frames), the item identification accuracy drops to 0.48.

8.4 Quantity Estimation

We then use the quasi-controlled study to evaluate *SmrtFridge's* coarse-grained quantity estimation technique. Figure 12 plots the estimated quantity for 3 different liquids {juice, milk, water}, and 3 different fractional quantities {30%,60%,100%}. The plot shows that these 3 levels are distinguishable (distinct mean values, with low overlap between 5/95% confidence intervals). However, the estimates are significantly more noisy for juice when the container is only 30% full). Studies with additional semi-solid items {yogurt, ketchup, peanut butter} show that the estimation error remains within 10-20%, indicating the robustness of our technique.

Coarser Estimation/Classification: While fine-grained quantity estimation is challenging for certain (liquid, container) combinations, coarser-grained estimates are acceptable for many applications. For example, an application that generates alerts (when the food quantity becomes very low) may just need to know when the quantity drops below, say, 20%. Accordingly, we now study the accuracy of the coarser-grained classifier that assigns the captured IR image into one of 3 bins/classes: 30|60|100%. For this ternary classification problem, we achieve a classification precision of 75% and recall of 71%. Overall, our results suggest that IR-based technique may be useful for obtaining coarse-grained quantity estimates (average error of ~15%).

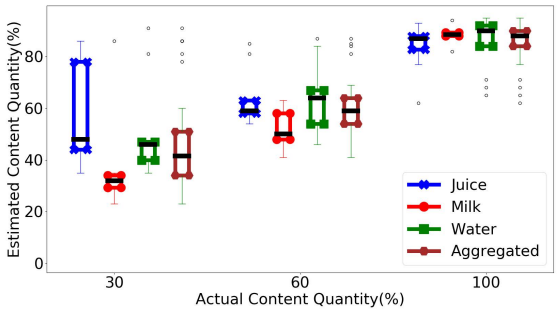
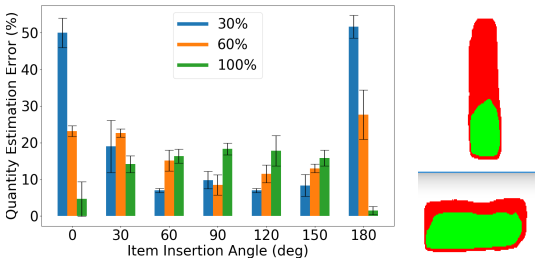


Figure 12: Median Accuracy of Quantity Estimation

Item Insertion Angle vs Accuracy: We also studied whether the estimation accuracy depends on the container's inclination angle. Figure 13a shows mean quantity estimation error, as % of whole container, when a juice container was put inside at 7 different horizontal angles (via a controlled study) ranging from θ = 0-180° (vertical $\rightarrow$ θ = 90°). We see that the estimation error is usually within 10-25% (and thus sufficient for coarse-grained resolution), unless the container is horizontal $\theta = \{0, 180\}°$. As an intuitive explanation, note that most food containers are taller and narrower. The same residual quantity thus results in a larger empty *height* when the container is vertical, and a much smaller empty height when horizontal. Moreover, we observed that even modest hand



(a) Quantity Estimation Error vs. Item Angle

(b) Thermal Clusters for $\theta = 90°, 0°(content = 30\%)$

Figure 13: Food item inclination angle performance

movements during the interaction can cause the liquid to splash vertically inside the container and 'contaminate' the empty portions "above". Given the relatively low spatial resolution of our IR camera, the clustering error (illustrated in Figure 13b) is thus much larger when the container is horizontal, than when vertical.

8.5 Additional Performance Characteristics

Supporting Multiple Items: While the *SmrtFridge* pipeline supports the user's concurrent interaction with multiple items, we observed that such interaction (e.g., retrieving a milk carton and a yogurt container together) is very unusual (never occurred in our Naturalistic study). To understand the performance of *SmrtFridge* under such possible multi-item interactions, we collected data for 8 episodes, where 2 users were explicitly instructed to retrieve 2 items concurrently. In this admittedly small sample, *SmrtFridge's* clustering technique reliably identified 2 distinct items and extracted them with IoU values between 0.63-0.71. However, more detailed studies are needed, as such concurrent retrieval may give rise to other non-obvious usage artifacts (e.g., occlusion of one items).

Energy Consumption: As per surveys [5], households tend to interact with their fridge about 15-25 times per day. Via measurements with 40+ distinct episodes, we found that average energy consumption is 7.5mWh/episode for the quantity estimation pipeline, and 90mWh/episode for the item extraction & recognition pipeline. In contrast, the yearly average energy consumption of a typical fridge (e.g. Toshiba GR-R20STE 185L), is 566 kWh. Accordingly, *SmrtFridge* is expected to impose an additional overhead of only **0.15**% on a fridge's energy consumption.

9 DISCUSSION

Privacy Concerns: *SmrtFridge's* use of an outward-facing camera can raise privacy concerns: consumers may be wary of devices that not only capture images of food items but also, potentially, that of the individual and the residence's background, *even if all image processing is performed locally on the fridge.* We believe that commercial products can address this issue via appropriate design and placement of cameras, while utilizing *SmrtFridge's* interaction-driven paradigm. In particular, instead of the outward-facing camera setup, we can deploy multiple narrow-FoV (field-of-view) cameras on the rim of the fridge, such that they are capable of only taking 'sideways' images of the fridge. However, such a setup can increase occlusion (at least on one side). Accordingly, we may need to modify the image extraction pipeline to accommodate multiple simultaneous images (of varying occlusion) from multiple cameras.

Extending to Other Food Types: The experimental results presented here focused primarily on discrete container-enclosed items, as discrete food items (e.g., oranges & eggplant). However, the quantity estimation of such discrete items is currently unexplored and will require newer approaches–e.g., thermal segmentation is unlikely to be able to distinguish between 1, 2 or 3 bananas.

Additional Sensors for Finer-grained Sensing: *SmrtFridge's* current visual recognition pipeline recognizes only food item *types/brands*, and not instances. For example, if a fridge has 2 Coke cans (both 50% full), the system cannot distinguish between them if one of them is removed and returned (with 30% residual content). Additional sensor types may help overcome such limitations. For example, explicit weight sensors (load cells), can help provide fine-grained estimates of changes in the fridge's weight, which can then be used to discriminate between multiple identical items. A single 100 lb (≈ 45 kg) Futek LSB200 sensor[6] can detect load changes as small as 10 grams. Other novel sensors may enable additional functionality, such as detection of expired food items. For example, Goel et al. [7] have applied hyper-spectral imaging to infer the aging of food items such as fruits.

10 CONCLUSION

We have demonstrated the *SmrtFridge* prototype, which innovatively combines infra-red (IR) and visual (RGB) camera sensors to provide two unique smart fridge capabilities: (i) food item identification and (ii) residual quantity estimation. *SmrtFridge's interaction-driven* sensing paradigm utilizes the clear images of a food item, captured during the transient period when the user either removes it from or places it into the fridge. By combining IR-based extraction of the cold portions of the image with an optical flow-based segmentation technique, we show that *SmrtFridge* can provide cropped images that contain ≥75% of a food item's pixels in over 95% of interaction episodes. This precise extraction helps a DNN-based classifier identify the food item with over 84% accuracy (which can be further improved with a larger corpus of relevant food item images). In a parallel process, the minor variations in thermal intensity between the filled and empty portions of a container can be used to achieve coarse-grained classification of the residual food content (between three levels: 30|60|100%) with ~75% accuracy. We believe that our work lays the foundation for utilizing multiple such IR and RGB sensors as part of a platform for highly accurate, unobtrusive monitoring of food item consumption.

11 ACKNOWLEDGEMENT

We thank the anonymous reviewers and our shepherd for detailed, constructive suggestions on improving the paper. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore partially under its IDM Futures Funding Initiative and partially under its International Research Centres in Singapore Funding Initiative, as well as partially by the National Research Foundation of Korea (NRF) grant (No. 2019R1C1C1006088).

[5] https://www.housebeautiful.com/uk/lifestyle/storage/news/a2110/fridge-food-cupboard-habits/

[6] http://www.futek.com/files/pdf/Product%20Drawings/FSH00091.pdf

REFERENCES

[1] [n. d.]. Deep Learning based system to recognize cooked food. https://foodai.org/. ([n. d.]). [Online; Last accessed 10-April-2019].

[2] [n. d.]. Introducing the liebherr smart refrigerator in cooperation with microsoft. https://blog.liebherr.com/appliances/sg/liebherr-smart-refrigerator-microsoft/. ([n. d.]). [Online; Last accessed 15-Feb-2019].

[3] Manuele Bonaccorsi, Stefano Betti, Giovanni Rateni, Dario Esposito, Alessia Brischetto, Marco Marseglia, Paolo Dario, and Filippo Cavallo. 2017. âĂŸHighChestâĂŹ: An Augmented Freezer Designed for Smart Food Management and Promotion of Eco-Efficient Behaviour. *Sensors* 17, 6 (June 2017), 1357. https://doi.org/10.3390/s17061357

[4] Pei-Yu (Peggy) Chi, Jen-Hao Chen, Hao-Hua Chu, and Jin-Ling Lo. 2008. Enabling Calorie-Aware Cooking in a Smart Kitchen. In *Persuasive Technology*, Harri Oinas-Kukkonen, Per Hasle, Marja Harjumaa, Katarina Segerst\a ahl, and Peter ÃŸhrstrÃ¥m (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 116–127.

[5] Angela Cusack, Arthur Fox, Audrey Hiscock, Marcus VanOirschot, Emeka Ogueji ofor, P Eng, and Mr Paul Dioron. 2012. Refrigeration revolution project proposal. (2012).

[6] Silvia Gaiani. 2013. *Lo spreco alimentare domestico in Italia: stime, cause ed impatti.* Ph.D. Dissertation. alma.

[7] Mayank Goel, Eric Whitmire, Alex Mariakakis, T Scott Saponas, Neel Joshi, Dan Morris, Brian Guenter, Marcel Gavriliu, Gaetano Borriello, and Shwetak N Patel. 2015. HyperCam: hyperspectral imaging for ubiquitous computing applications. In *In Proc. of UbiComp*. ACM.

[8] Hanshen Gu and Dong Wang. 2009. A content-aware fridge based on RFID in smart home for home-healthcare. In *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, Vol. 2. IEEE, 987–990.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity Mappings in Deep Residual Networks. *CoRR* abs/1603.05027 (2016). arXiv:1603.05027 http://arxiv.org/abs/1603.05027

[10] Avi Itzkovitch. 2013-09-18. The Internet of Things and the Mythical Smart Fridge. (2013-09-18). https://uxmag.com/articles/the-internet-of-things-and-the-mythical-smart-fridge.

[11] Yijun Jiang, Elim Schenck, Spencer Kranz, Sean Banerjee, and Natasha Kholgade Banerjee. 2019. CNN-Based Non-contact Detection of Food Level in Bottles from RGB Images. In *MultiMedia Modeling*, Ioannis Kompatsiaris, Benoit Huet, Vasileios Mezaris, Cathal Gurrin, Wen-Huang Cheng, and Stefanos Vrochidis (Eds.). Springer International Publishing, Cham, 202–213.

[12] Juliane JÃűrissen, Carmen Priefer, and Klaus-Rainer BrÃďutigam. 2015. Food Waste Generation at Household Level: Results of a Survey among Employees of Two European Research Centers in Italy and Germany. *Sustainability* 7, 3 (2015), 2695–2715. https://doi.org/10.3390/su7032695

[13] Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa. 2014. Food Detection and Recognition Using Convolutional Neural Network. In *ACM Multimedia Conference*. https://doi.org/10.13140/2.1.3082.1120

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[15] Suhuai Luo, Jesse Jin, Jiaming Li, et al. 2009. A smart fridge with an ability to enhance health and enable better nutrition. *International Journal of Multimedia and Ubiquitous Engineering* 4, 2 (2009), 69–80.

[16] Simon Mezgec and Barbara Korouić Seljak. 2017. NutriNet: A Deep Learning Food and Drink Image Recognition System for Dietary Assessment. In *Nutrients*.

[17] Satoshi Murata, Shota Kagatsume, Hiroaki Taguchi, and Kaori Fujinami. [n. d.]. Perfridge: An augmented refrigerator that detects and presents wasteful usage for eco-persuasion. In *Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on* (2012). IEEE, 367–374. http://ieeexplore.ieee.org/abstract/document/6417317/

[18] Amavi Djimido Noutchet. 2013. Novel User Centric RFID Fridge Design. *Computer and Information Science* 6, 2 (April 2013). https://doi.org/10.5539/cis.v6n2p151

[19] Alexandru Popa, Mihaela Hnatiuc, Mirel Paun, Oana Geman, D. Jude Hemanth, Daniel Dorcea, Le Hoang Son, and Simona Ghita. 2019. An Intelligent IoT-Based Food Quality Monitoring Approach Using Low-Cost Sensors. *Symmetry* 11 (2019), 374.

[20] JosÃľ Rouillard. [n. d.]. The Pervasive Fridge. A smart computer system against uneaten food loss. ([n. d.]), 7.

[21] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53 – 65.

[22] Thomas Sandholm, Dongman Lee, Bjorn Tegelund, Seonyeong Han, Byoungheon Shin, and Byoungoh Kim. 2014. CloudFridge: A Testbed for Smart Fridge Interactions. *arXiv:1401.0585 [cs]* (Jan. 2014). http://arxiv.org/abs/1401.0585 arXiv: 1401.0585.

[23] Sougata Sen, Vigneshwaran Subbaraju, Archan Misra, Rajesh Balan, and Youngki Lee. 2018. Annapurna: Building a Real-World Smartwatch-Based Automated Food Journal. In *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 1–6.

[24] Ju Wang, Jie Xiong, Xiaojiang Chen, Hongbo Jiang, Rajesh Krishna Balan, and Dingyi Fang. 2017. TagScan: Simultaneous Target Imaging and Material Identification with Commodity RFID Devices. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, MobiCom 2017*.