Pocket Kitchen: Implementing Machine Vision for Food Management

Mimi Reyburn*
Dyson School of Design Engineering
Imperial College London

ABSTRACT

Households are the biggest contributors to UK food waste, surpassing both the production and retail sectors. While food management techniques can effectively reduce this waste, they require too much time for consistent practice in daily life. This problem presents an ideal opportunity for automation. However, physical sensing in research rarely provides true ‘automation’, requiring significant user behavioural change. Here we pioneer a machine vision method to automate domestic food inventory without requiring a change to user storage habits. We harness machine vision to integrate decay state recognition into inventory tracking, a functionality that is difficult to achieve with physical hardware. We show that our object detection model trained on a custom dataset of 2740 images can recognise 16 classes of food on a Raspberry Pi with 69.4% accuracy. We present a method that combines automated inventory with food management techniques, to inform waste-reducing interventions through our web app, Pocket Kitchen. We open-source Pocket Kitchen and its dataset as the first step towards user-oriented domestic food management with machine vision.

1 INTRODUCTION

According to WRAP (Waste Resources Action Programme), a climate action non-governmental organisation, the UK wastes 9.5 million tonnes of food every year [49]. While waste occurs throughout the food supply chain, 70% derives from households. Of this, 70% is ‘edible’, equating to 4.5 million tonnes of food - a waste of 243 litres of water per person per day and a land area the size of Wales [49]. Excluding the ‘inedible’ parts, each individual wastes an estimated £210 worth of food per year [49]. Subsequently, it is feasible that an efficient, affordable solution could pay for itself in savings. Furthermore, the nutritional value of this waste could provide the necessary nutrients and energy to an adult for 42 days [22]. With rising food bank dependence [17] and a population expected to grow to 72 million by mid-2041 [28], whether morally, environmentally, or economically, there is no shortage of incentives for ending food waste.

Households dispose of edible food for many reasons: 41% is not used in time, 28% is due to personal preference and 25% because too much food is prepared [49]. WRAP found that, despite an increase in purchasing during the first UK lockdown, food waste saw a dramatic 43% decrease[48]. 79% of citizens adopted an average of 6.7 new food management behaviours, which effectively reduced waste; some of the most significantly increased behaviours were:

(1) Checking what you have in the cupboards before shopping.
(2) Checking what you have in the fridge before shopping.
(3) Cooking creatively (e.g. trying new meals and recipes).
(4) Making a meal by combining random ingredients you happen to have.
(5) Managing the cupboards (stock and expiry dates).
(6) Managing the fridge (stock and expiry dates).[48]

Citizens also cited freezing items, shopping lists and saving leftovers as some of the most effective behaviours they adopted. However, 44% of those who stopped their new behaviours by September 2020 cited a lack of time [48]. Modern life and busy schedules are often incompatible with meal planning and carefully organised shopping trips, ultimately leading to waste [30]. Although still lower than pre-Covid, levels of food waste began to increase again as government restrictions eased [48]. *How might we use technology to reinforce helpful behaviours and encourage them post-pandemic?*

Sharing apps offer systematic means to reduce domestic food waste, but the method is remedial, not preventative. Similarly, home composting, anaerobic digestion and rapid composting products prevent biodegradable waste from reaching landfills and emitting Greenhouse Gases (GHGs). However, these services cannot process all types of food, they do not begin to return the value lost and they are not yet widely available [39].

Smart fridges have been produced in commercial and research settings, preventing waste via inventory tracking and stock-level monitoring. Interventions can provide comprehensive knowledge of fridge inventory, facilitating companion apps with on-the-go inventory information [36, 38, 43], recipe suggestions [27, 38], shopping lists [27], expiry alerts [24, 26, 42] and even real-time internal fridge images [14–16, 46]. However, commercial smart fridges are too expensive for typical households [14, 15] and still require users to perform manual labelling.

Research has explored a variety of approaches with physical sensors, such as Radio-Frequency Identification (RFID), barcode scanning and photodiodes. However, these methods fall short; they require significant user intervention or their applicable scope is limited (Appendix A). Ferrero et al. recognise that overly complex or disruptive methods provide barriers to adoption integrating Google® Assistant to reduce the learning curve for their proposed smart fridge [27]. Nevertheless, a gap persists for a system where users need not create new habits or provide significant input to obtain food inventory.

We take inspiration from successful behaviours adopted in lockdown to aid in tracking food for households. The main contributions of this paper are as follows:

(1) A first-of-its-kind end-to-end system for inventorying food in the home with machine vision and decay state recognition.
(2) A first-of-its-kind open-source custom dataset of 2740 images of 13 ingredients and store-bought food items in 16 classes in the context of the home.

*Terminology. The "camera" refers to the HQ Camera and attached 6mm Wide Angle Lens. "Food" refers to consumables, including drinks such as milk and juice.

(3) Automated food inventory via object detection model demonstrating 69.4% mean Average Precision in perceived real-time in the perspective of a user (an average 6.1fps operational frame rate).

2 RELATED WORK

2.1 Datasets and Computer Vision

Convolutional Neural Networks (CNNs) are a highly effective deep learning method primarily used in image processing. CNNs with optimised hyper-parameters obtain higher reliability in food detection than traditional support-vector machine methods with hand-crafted features [32].

Our initial testing of models trained on ImageNet found that food could be mistaken for everyday objects, from a carrot resembling a surfboard to an egg box resembling cake (see Appendix B). It follows that datasets tailored explicitly to food items are necessary for this work.

Kagaya et al. use publicly available food logging data to train their CNN detection with 93.8% accuracy [32]. However, most images feature cooked foods, meals, or items served on crockery instead of packaging. Similarly, the most significant real-world food recognition database ETH Food-101 comprises mostly of meals (see Figure 1), not individual ingredients, as you would expect to see them in domestic storage [20]. We address this problem in Section 3.4 with our custom dataset.

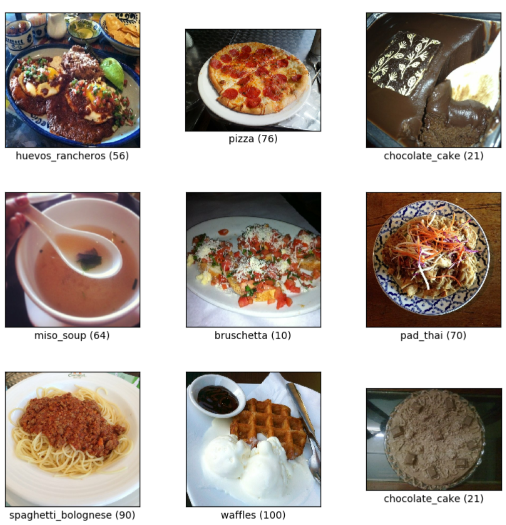

Figure 1: Examples from the Food-101 Dataset [20]

Megzec et al. introduce the NutriNet deep CNN, a modification of the AlexNet architecture, which presents a higher classification and detection accuracy for food and drinks images rooted in a 520-class image dataset from Google image searches [35]. However, the model sees dramatic losses when classifying real-world images. Megzec et al. cite overfitting, added noise and occlusion as potential sources for this inaccuracy and highlight the limitations of image classification. The single output per image from their model allows some items to go amiss. We address this problem in Section 3.5 with object detection.

Wong et al. demonstrate successful synthetic dataset generation for the modelling and recognising store-bought items (95.8% accuracy) [47]. However, data generation required significant time and specialised labour per individual item, limiting scalability. We address reducing the time and labour in data collection in Section 3.4 and propose methods for scaling in Section 5.1.8. Klasson et al. present a dataset for grocery store items with 42 coarse-grained classes for fruit, vegetables and refrigerated items for the aid of visually-impaired store customers [33]. Figure 2 shows some examples. While acceptable for their use case, images featuring bulk items and store context provide complications to labelling and add confusion to a model intended for use in the home.

Figure 2: Examples from the Grocery Store Dataset [33]

2.2 Machine Vision

Jain et al. demonstrate the Inception-V3 CNN in a trolley module for recognising items entering the fridge with item weight integration [31]. However, it is limited to fruits and vegetables and requires high user input and non-conventional behaviour to add them to a list.

Identifying bottles and canned drinks with object detection has been explored, with Telegram notifications for stock management [44]. We adopt this method for notifying users in Section 3.6.3. Although Soh et al. achieve high accuracy, the process is too demanding for microprocessors, resulting in a more invasive and costly intervention. We address this in our design objectives, Section 3.2. Avinash et al. propose a fridge redesign using a CNN for recognition and an android app for food management; however, they present no evidence of implementation [18].

Shweta uses image classification trained on self-collected data to identify a small variety of vegetables in a fridge vegetable tray [43]. Using image histograms, classification based on colour, texture, shape, and size finds the best match with 96.55% accuracy. Rouillard also utilises image recognition in the Pervasive Fridge but finds barcode technology more efficient [42].

2.3 Decay recognition

CNNs and support vector machines can also identify the degradation of select food items. Billah et al. attain 90% accuracy in classifying the age of bananas [19], and Zhu et al. utilise the You Only Look Once (YOLO) algorithm to identify defected areas on their peel accurately [50]. There is also potential for further work as Das et al. offer a large dataset for grading the freshness quality of tomatoes [25].

2.4 Mobile Apps

The Kitche app provides inventory management and recipe suggestions from scanned receipts with estimated expiry reminders based on food type [8]. However, as one user review states, it "won't work if someone else occasionally raids your fridge!"[9]. The NoWaste app provides a solution, allowing users to share inventory and shopping lists [11]. Both NoWaste and CozZo offer the capability to add items by scanning barcodes [4]. Nevertheless, users must always manually add expiry dates as the available databases do not provide individual product expiry data. Similar apps, such as 'BEEP', an expiry date tracking app, find negative user reviews regarding this user requirement [1].

2.5 Discussion of literature

Food datasets (2.1) in the literature progress towards food item recognition but focus on bulk items, which do not apply to domestic inventory management. Machine vision applied to inventory (2.2) has been too computationally expensive and fails to make full use of ML capabilities. Neither datasets nor vision present examples of integrated decay state recognition in inventory management (2.3). Finally, mobile apps in the literature(2.4) require too much manual input and user behavioural change and lack the benefits of automation.

With a suitable dataset and model, machine vision provides accurate image recognition for food items with potential expansion to identifying degradation. Furthermore, it requires little-to-no user input to take food inventory, promoting easy adoption and the potential to recognise any item. While computer vision sees previous success in food and drinks recognition, ingredient recognition is unprecedented in a domestic context. It also provides a significant opportunity to integrate decay states into inventorying domestic food automatically.

3 METHODOLOGY

Literature review informs the opportunity for a novel distributed system. The research objective is to validate the feasibility of an end-to-end food management system facilitated by machine vision. We will move beyond approaches in the published literature, integrating decay states and introducing a dataset for ingredients and store-bought items in the domestic context. Focusing on the fridge environment provides an appropriate starting point, considering the appliances store most perishables. Likewise, limiting the scope of detection to less than 20 classes will create a detailed model without breaching the system requirements of a lightweight computer. Selecting ten recipes will demonstrate recipe suggestions based on the capped items contained in the classes.

3.1 Project Approach

We divide the research into five sections: Hardware, Dataset, Back-end: Vision, Back-end: Web-App and Front-end. We adopted a hybrid approach during the project. First, a waterfall approach catered for planning, initial script development and establishing an initial 'bare-bones' integration of the project divisions, including hardware. We then adopted an agile approach to make stepwise improvements to each section. In addition, we take a minimum-viable-product approach to prototyping hardware.

3.1.1 Design Approach. We favour a double diamond design approach. Appendix C provides further details of the design approach followed and how it furthered the impact of this project.

3.1.2 Use Case. We consider the prospective end-users - the food purchaser and the home cook - to create an appropriate and affordable system proposal. This project aims to develop a working proof of concept for such a system.

3.2 Design Objectives

The ultimate objective of this research is to present a first-of-its-kind end-to-end food management system with integrated decay state recognition enabled by machine vision. Our objectives for a technical system design are as follows:

(1) *Affordable*
Considering that food waste costs the average family £720 per year [49], we can be confident that a solution costing between 0.5 and 1 times this will be affordable, especially if produced at scale.
(2) *Accessible*
Design for retrofitting reduces unnecessary electronic and plastic waste and overall cost. In addition, we assume the fridge to be an essential kitchen item in the home. Therefore, no new kitchen appliances are required, and we demonstrate the system's capability in other kitchen storage areas.
(3) *Small and non-invasive*
We wish to minimise user disturbance, whether this disruption stems from extensive user input or invasive hardware. However, the literature review showed that interventions could be large and expensive. Therefore, we require a non-intrusive system design for easy, subtle integration into the existing home environment. Consequently, we design for the use of single-board computers (SBCs).
(4) *Minimal user-input automated inventory tracking*
Awareness of food inventory correlates with reduced household food waste. Integrating stock management seamlessly into users' lives will promote preventative food waste practices. Minimal user intervention offers the fewest barriers to entry and, therefore, the highest likelihood of successful adoption.
(5) *Responsive*
The detection of food items and their addition to the database should occur in a time frame close to real-time, as a user would perceive it.
(6) *Educate users on waste streams*
We choose not to propose landfills as a waste stream to the user. Since education encourages behavioural change, a GUI

will prompt users to divert excess to responsible streams, such as food sharing systems, and waste to home composting or anaerobic digestion.

(7) *Open-source dataset for store-bought food items*
A minimum of 1000 images will comprise a dataset for store-bought food items and decay state recognition in the home context, where some of the most wasted food items will be targeted [49].

(8) *Decay state recognition*
We will explore the detection of decay states for three food items. Alerts to decay states will support waste reduction at the most common waste stream (41% not used in time).

We detail metrics for benchmarking design objectives in Appendix D.

3.3 Prototyping Approach: Hardware

The hardware (see image) consists of a 4GB Raspberry Pi 4, Raspberry Pi High-Quality camera and a 6mm Wide Angle Lens. The internal fridge environment can interfere with electronic components and is dark unless the door is open. Consequently, we mount the camera on the outside of the appliance, obtaining a birdseye view.

3.3.1 Raspberry Pi. Raspberry Pi provides a small, affordable computer ideal for devices in the Internet of Things, Sensing and Control domains, and attaching to a fridge. Furthermore, its many ports provide opportunities for expansion and modular design.

3.3.2 Coral AI Accelerator TPU. Object detection is computationally expensive and negatively impacts frame rate. With Raspberry Pi, the Google Coral AI Accelerator dramatically improves the inference speed of object detection, enabling more accurate tracking of moving items within the camera frame.

3.3.3 Rapid Prototyping. Additive manufacturing provides an affordable means of rapidly refining and obtaining desired parts for an experimental set-up. The mount design was improved in iterations to be more versatile than the initial screw-based iteration.

Figure 3 shows three prototypes for mounting the Pi to the fridge. The first was made from mount-board and screwed into place. The second was 3D-printed for strength. While it has holes for screws, clamps are necessary to secure it to a different fridge style. Noting this we refined the final design for clamps, since they offer more adaptability than screws. The final design features two long clamps and a mounting frame designed to work with them.

Figure 3: Prototypes 1-3

We use 3D prints to mount the HQ camera and lens to the pi (see Figure 4), creating a concise protected part and a familiar camera shape recognisable to a user. The frame is made custom fit to the camera in Autodesk Fusion 360 to secure it in place whilst retaining adaptability. This method can retrofit many fridges and cupboards. The clearance of the fridge door is dependent on the mount's height. While this may vary across different appliances, clamps allow the addition of spacers. Figure **??** shows an overview of the final system prototype.

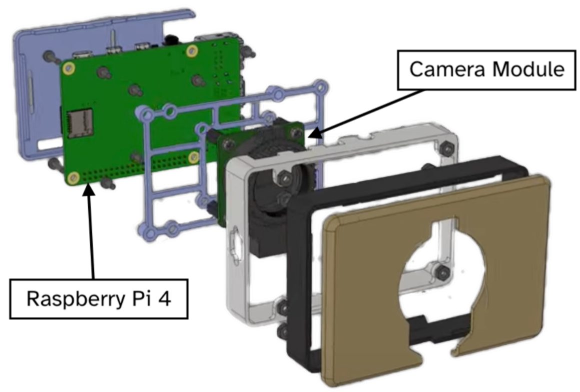

Figure 4: 3D Printed Camera Case Assembly [10]

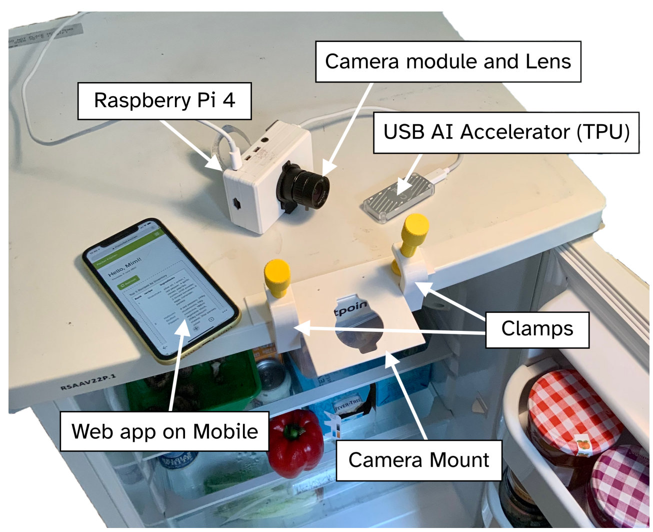

Figure 5: Hardware Prototyping Overview

3.4 Prototyping Approach: Dataset

3.4.1 Dataset. Machine vision facilitates generalised sensing. Internet-collected datasets often see poor accuracy in real-world testing, signifying the importance of accurate data [35]. Some of the most extensive datasets also lack images from a domestic context [20]. Despite demonstrating realistic outputs, creating synthetic datasets

Class	Number of Images
Banana	331
Bread	151
Carrot	76
Cream	191
Juice	309
Milk	239
Mushrooms	364
Pepper	278
Potatoes	146
Salad	310
Soft-Fruit	273
Yogurt	142

Table 1: Class Sizes for the Dataset

can require specialised technical knowledge and significant labour per item for accurate food recognition [47].

We establish an image dataset for ingredients and store-bought items in the home environment. We use the HQ camera to capture images both in and out of context. Discussion with experienced researchers determined a minimum of 1000 images to achieve acceptable results. We use a Python script, 'photo.py', to add file names in bulk and a button to capture images, speeding up data collection and aiding labelling later on. While Google Cloud Storage proved helpful for data storage, transferring files across the local network sufficed and was advantageous for its lack of cost.

We give individual labels to decay states, simplifying their inclusion in the dataset. We experiment with three food items: peppers, mushrooms and bananas. Figure 6 provides an example.

Figure 6: Dataset example labelled 'pepper-bad' (left) and 'pepper' (right)

3.4.2 Dataset v2.0. improve the dataset by removing some ambiguos images and evening out class sizes, Table 1 describes the composition of final 2740 image datatset.

We open-source the dataset on Github for community contribution and furthering the field of food recognition in domestic environments:
https://github.com/myPocketKitchen/PocketKitchen-Dataset

3.5 Prototyping Approach: Vision

3.5.1 Labelling. 'LabelImg' is an open-source Python program for image labelling with bounding boxes (bboxes). Image classification models can miss food items if there are multiple in one image [35]. Object detection, however, requires bboxes to identify multiple items in one frame and their locations, enabling the tracking of items through the frame. With the benefit of our overhead view, we can store bbox coordinates to determine whether items enter or leave their storage environment. Labels are in PascalVOC format for TensorFlow. All images were labelled manually.

3.5.2 Object Detection. We select the EfficientDet model architecture for object detection. Figure 7 shows that EfficientDet offer high mean average precision (mAP) for reduced computation against other standard architectures [45].

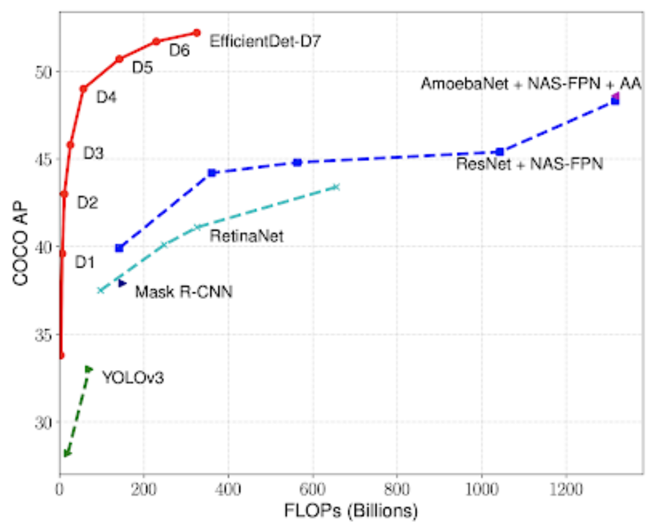

Figure 7: Tan et al. find EfficientDet models are 4x-9x smaller and use 13x-42x less computation than previous detectors for the same accuracy during testing [45].

TensorFlow EfficientDet models are convertible to TFLite models, compatible with Raspberry Pi, where standard object detection models are often too large. Furthermore, unlike other models, such as tiny-YOLO, TensorFlow offers easy adaption for TPU acceleration with the Coral AI Accelerator (Edge TPU), reducing inference time and improving our ability to track items through the camera frame in a small, non-intrusive prototype.

EfficientDet has several architectures programmable with TensorFlow and Edge TPU. We select our model by balancing system specifications, latency and accuracy. The Coral AI Accelerator has approximately 8MB of SRAM. Breaching this requirement will increase inference times as model parameters will be fetched from host system memory. We prioritise reducing inference times to

Model architecture	Minimum TPUs	Recommended TPUs
EfficientDet-Lite0	1	1
EfficientDet-Lite1	1	1
EfficientDet-Lite2	1	2
EfficientDet-Lite3	2	2
EfficientDet-Lite4	2	3

Table 2: Recommendations for the number of Edge TPUs to use with each EfficientDet-Lite model

Model architecture	Size(MB)*	Av. Recall**	Av. Precision ***
EfficientDet-Lite0	5.7	54.6%	64.5%
EfficientDet-Lite1	7.8	62.2%	64.7%
EfficientDet-Lite1 (Dataset v2.0)	6.0	66.1%	69.4%

Table 3: The Performance of each trained EfficientDet-Lite Model Compared to each other
*** Size of the integer quantized models complied for TPU with metadata.**
**** Average Recall is the mAR (mean Average Recall) given one detection per image.**
***** Average Precision is the mAP (mean Average Precision).**

work towards perceived real-time performance. Since we expect future work expanding the dataset can improve accuracy, we trade off accuracy to reduce latency as well as restricting our model to less than 8MB. We have access to one Coral AI Accelerator; Table 2 shows that EfficientDet-0 and 1 offer the most reliable service with our system specifications.

Table 3 shows iterative improvement on our model during the agile phase of the project. We evaluate our models with the COCO (Common Objects in Context) evaluation metrics [3]. These metrics are sector standard tools for evaluating and comparing the accuracy of different object detection algorithms.

We refine to the EfficientDet-Lite1 model trained on Dataset v2.0. Figure 8 shows an example detection by our model during testing.

.

3.5.3 Detect.py and MongoDB Atlas. Figure 9 shows the flowchart for processes on the Raspberry Pi. We add the bottom bounding box coordinate to a dictionary with the recognised item as the key. Once list length passes a threshold, we determine if the item moves down or up the frame (in or out of the fridge, respectively). Then, we add or remove the item from the database and clear the values in the dictionary.

MongoDB Atlas is our database of choice; it is a free, fast and well-documented NoSQL database storing JSON format documents.

3.6 Prototyping Approach: Back-end - Web App

3.6.1 Calculating Expiry Dates and recipe suggestion. Figure 10 shows the flowchart for back-end processes for the web app, performed both client and server side. Upon adding a new item, we prompt the user for an expiry date, which we upload to our database.

Figure 8: An example detection by our model during testing

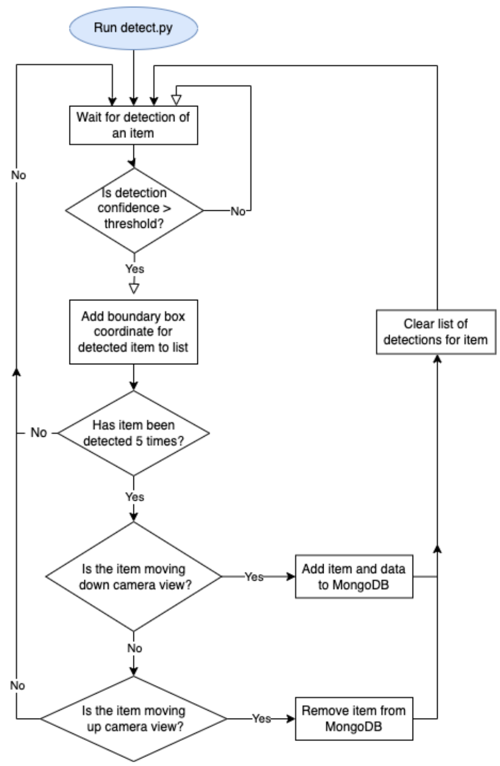

Figure 9: A flow chart for processes on the Raspberry Pi

From the expiry date, we calculate a countdown in the client-side javascript. These enable different nudges based on proximity to expiry as well as recipe suggestions based on using foods most at risk of going to waste.

Modals inform users of responsible waste streams, should an item become inedible, and encourage users to maximise their food use.

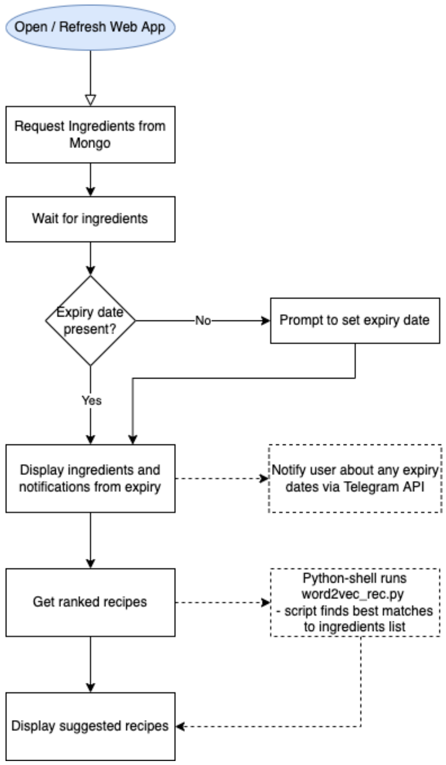

Figure 10: A flow chart for back-end processes on the Web App

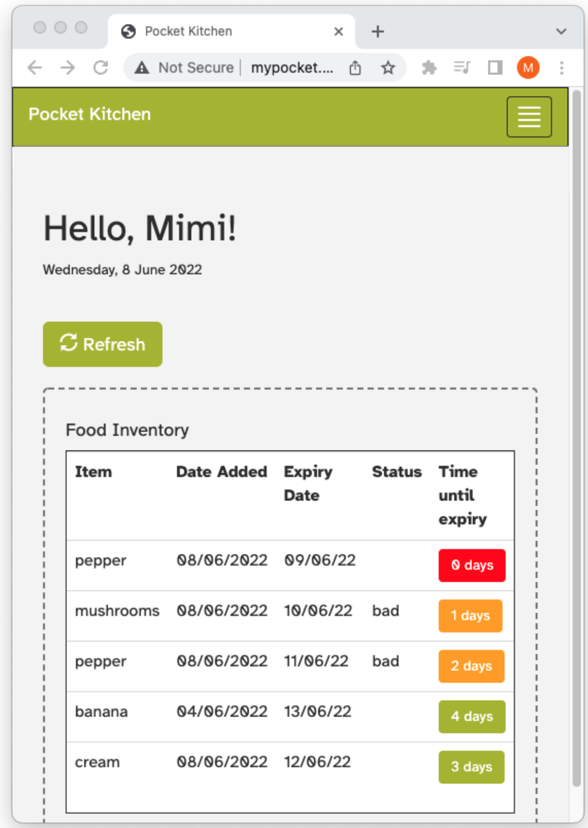

Figure 11: Pocket Kitchen Graphical User Interface - Mobile. Avaliable at www.mypocket.kitchen

3.6.2 Recipe Suggestion with Wordnet. Recipe suggestion from a string is already successful [12], and we incorporate an established method into the web app. The method uses machine learning (ML) and word vectorisation to improve suggestions and is scalable. The original author's application is built upon two-thousand web-scrapped recipes and provides the means to expand our ten recipe demonstration with ease [7, 34].

3.6.3 Telegram Notifications. Users can subscribe to Telegram notifications from Pocket Kitchen from the Web app. Users receive alerts notifying them of foods close to expiry to encourage their use.

We host the web app on Heroku with Express for access anywhere with an internet connection.

3.7 Prototyping Approach: Front-end

3.7.1 Web App Interface. The web interface is coded with Bootstrap 3 - a mobile-first web interface package - and presents three main features: Food Inventory, Recipe Suggestion and Expiry Notification. Fig 11 shows the GUI created for Desktop and Mobile use. Mobile accessibility is vital to reducing household waste, allowing users to always check their inventory before shopping.

3.7.2 Food Inventory. Requests to a Mongo database retrieve a list of items identified to be within the fridge. We also record the date and time of entry to the fridge. New items prompt the user to suggest or enter an expiry date, which we return to the database. Upon the approach of an item's expiry, we notify users via a Telegram message. Design psychology informs nudges with a traffic light system to increase general awareness of item expiry.

3.7.3 Recipe Suggestion. We demonstrate ranking according to two distinct criteria: what a user has and what is most at risk of going to waste. This encourages the food management technique, "making a meal by combining ingredients you happen to have." [48]

3.7.4 Education. Positive waste streams are widely encouraged. When items pass or approach expiry, we encourage users to judge for themselves if an item is edible, as recommended by WRAP and Government advice [2, 5]. Should an item be wasted, we propose responsible disposal practices and never landfills.

3.7.5 Ethics and Privacy. Mongo mediates between the Pi and the global network (Heroku). It permits the separation of sensitive data on the Pi from the global network.

4 OUTCOMES

Previous literature regarding food inventory management consistently demands high levels of user input. More autonomous systems

Class	Mean Average Precision
Banana-Ripe	49.4%
Banana-Overripe	79.5%
Banana-Unripe	22.1%
Bread	74.2%
Carrot	66.0%
Cream	78.0%
Juice	79.9%
Milk	83.2%
Mushrooms-Good	70.7 %
Mushrooms-Bad	88.8%
Pepper-Good	74.6%
Pepper-Bad	53.0%
Potatoes	66.7%
Salad	74.8%
Soft-Fruit	80.4%
Yogurt	69.6%

Table 4: Mean Average Precision for Classes in the EfficientDet-Lite1 Model

provide fewer barriers to entry but are rarely thoroughly investigated. Machine vision facilitates such a system; food inventorying that asks less of the user and provides more from their food.

4.1 System

Pocket Kitchen provides a charming and informative title to a human-oriented system. The distributed system is outlined in Figure 12. The system is comprised of hardware, software and a cloud-based database connected over the global network. We facilitate access on the go with a Heroku-based web app:
www.mypocket.kitchen

4.2 Dataset Creation and Model

Object detection provides multiple outputs per frame, recognising items that image classification may miss. Bounding boxes also allow us to track items through the frame, determining whether items are being added or removed. Models were trained in Google Colab and on a local computer with TensorFlow. We perform object detection on the edge (on the Raspberry Pi) and improve frame rate by 84.8% by enabling an Edge TPU. Making trade-offs between speed and accuracy to create a functional and efficient system was crucial. Table 4 shows our promising results for each class. We use the sector standard COCO tools to evaluate our model performance [3]. Averaged across all classes, we achieve a mAP of 69.4% .

4.3 Testing

Physical testing with the camera viewport aided the selection of a model with the designed system. Our agile approach allowed steady improvement of the model and refinement of the dataset.

The prototype incurred 3 days of testing. Figure 13 shows the testing setup.

4.4 Benchmark Evaluation Table

Ahead of prototyping, we set several design objectives. Upon completion of the end-to-end system prototype, we return to our goals to measure achievements against measurable benchmarks in the context of our design intentions, provided in Appendix E. We provide a discussion of benchmark success in Section 5.1

4.5 Open Source

This project would not have been possible without open sourcing. Community aids technological development, and this project is no exception. We open-source this project to encourage further research into technical interventions for domestic food waste to promote machine vision to reduce barriers to entry and to make our dataset of domestic food items globally available.

5 DISCUSSION

5.1 A Reflection on Design Objectives

5.1.1 Affordable and retrofitted. Efforts toward domestic food management in the literature regularly require hardware beyond the budget of the typical intended user [44]. Excluding ‘inedible’ parts, families waste an estimated £720 worth of food per year [49]. Likewise, individuals waste an average of £210 per year. Since the project costs came to £187.99, it is feasible that this intervention could pay for itself in savings.

While prototyping enjoys feeless server and storage usage, this is not possible on a widespread intervention. In addition, increased database security is necessary at scale; food inventory contains valuable and sensitive personal data. In contrast, components bought in bulk are cheaper. Raspberry Pi provides surplus capabilities for this use case. Future work could explore custom components tailored to the project needs and optimised for inference speed reduction.

Retrofitting all cupboards and fridges poses a substantial challenge. Modules for appliance adaption in the literature are frequently too large for subtle integration into the home [18, 44]. We make a reasonable attempt with 3D-printed clamps and a mounting frame. In addition, we find that matching the appliance colour allows for a discreet modification. Still, this does not cater for all appliances and home installations. Future work could facilitate a universally applicable mount design.

5.1.2 Small. As previously mentioned, interventions observed in the literature repeatedly offer invasive hardware for the home environment. For example, one even offers a whole trolley module to accompany your appliance. Instead, we present a minimalist fridge or cupboard attachment that could fit comfortably inside a shoebox. Detection runs on the Coral AI Accelerator, supported by a lightweight computer - the Raspberry Pi 4.

5.1.3 Accessibility. We use hobby components and non-specialised parts to construct the hardware set-up. While some parts are 3D printed, equivalent low-cost commercial parts are available. Furthermore, the custom mount could be hand made from any arbitrary materials as in the initial prototype.

System benefits are redeemable from anywhere with internet access. We make the web app available on the global web to permit the benefits of inventory management beyond the confines of the home and embed it into our newly returned ‘normal’ lives. However,

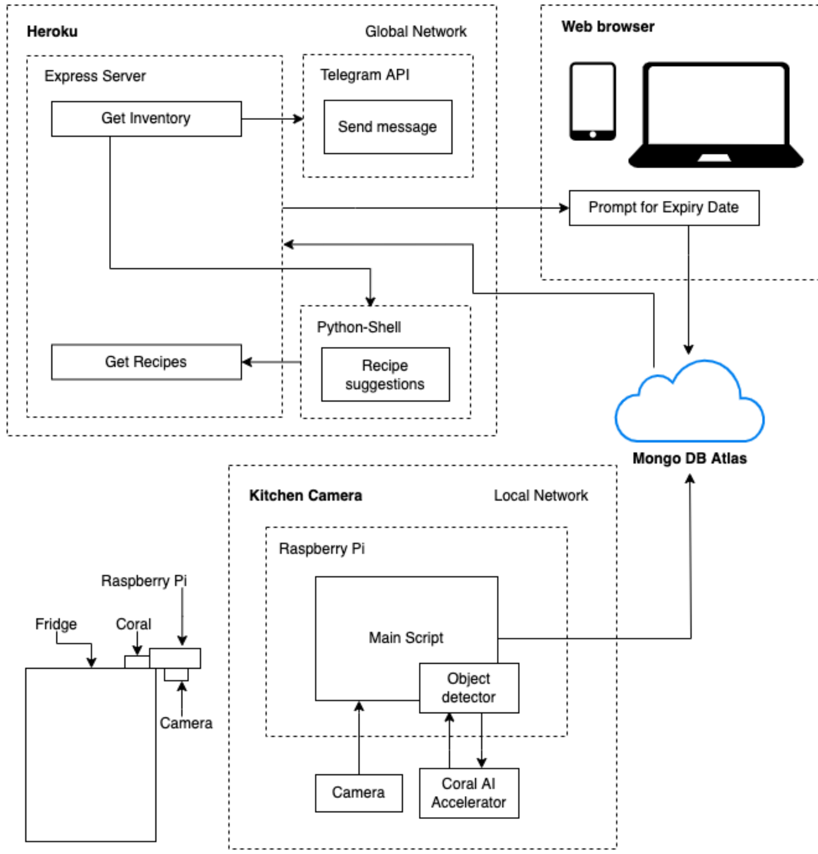

Figure 12: Pocket Kitchen System Architecture

without an internet connection, the system cannot be accessed. Future developments could see the integration of offline capabilities into a mobile application.

5.1.4 Minimal user input automated inventory tracking. Methods for inventory tracking seen in the literature consistently require significant user input, particularly since WRAP finds users struggle to maintain effective behaviours due to a lack of time [48]. We successfully implement object detection to identify and track food items with minimal user input. Our TFLite Object Detection model achieves an average accuracy of 69%, with more than half of the classes achieving over 70% accuracy (see Table 4). Given that we use 16 classes, this provides better than chance performance, and further work could continue to improve accuracy.

However, machine vision does have some limitations. Firstly, each new item added to the dataset requires the generation of a new model. However, transferring ML processes to the Cloud with AutoML could minimise interference. Secondly, items smaller than the human hand, therefore concealed by it, will be missed altogether. Using a second camera may help but not completely solve the problem. Therefore, users must make a specific effort to 'show' each item to the camera. Since this minor adjustment facilitates the system, we deem it feasible and believe users can adjust naturally over time. A fridge-mounted GUI could aid the process in future.

Since the camera cannot automatically focus, the field of view is not all in sharp focus. However, an improved, broader dataset or data augmentation could help accommodate this characteristic.

We use prompts to obtain expiry dates. Even if users estimate a date, it adds value, allowing the system to encourage users to eat their stock before expiry. However, further work could implement a second model to recognise expiry dates for select items.

We successfully demonstrate automatic inventorying of food items; however, we cannot determine how much of an item we possess. We could obtain this data by fitting load cells to the fridge or cupboard shelves to determine the change in mass for each item. A further dataset could aid this process with packaging weights for standard items. In addition, this could aid another limitation; all items entering the storage environment are identified as new, even those being returned there. Finally, accurate weight sensing could facilitate the identification of returned items. We offer a rudimentary demonstration of combining weight and vision sensing during previous work [6, 13].

Figure 13: Prototype Testing Set Up

5.1.5 *Responsive.* Our model makes detections in near real-time in the perception of a user with an average frame rate of 6.1s. Detections are made and added to the database in a matter of seconds (on average, 110.3ms). As a result, the detection rate performs sufficiently well for maximum usability and user convenience.

5.1.6 *Educate users on waste streams.* Multiple methods in the literature, apps, in particular, apply several user interventions in one design, maximising effectiveness and impact. We adopt this technique in our distributed system.

Design psychology informs the use of nudging. For example, during front-end web development, we employed the strategy to urge the user towards positive food management practices and waste streams where necessary; a traffic light system communicates risk to the user and conveys urgency. As well as this, Telegram supplies friendly prompts to promote the use of 'at-risk' items.

5.1.7 *Decay States.* In our model, both unripe and ripe bananas face low recognition rates (see Table 4). Therefore, we can infer that the similarity of the classes creates confusion in the model. However, it is worth noting that this may align with human recognition of bananas since it is often hard to distinguish ripeness by sight. Moreover, determining ripeness often relies on personal preference. Therefore, improving the distinction between the classes and refining their definition will reduce ambiguity.

Literature explores specific models for decay state recognition tailored to each item of food [19, 50]. In future, we may also wish to consider using a separate model tailored to determine each item's state of decay.

5.1.8 *Open-source dataset for store-bought food items.* Our dataset contains 16 coarse-grained classes with seven fine-grained classes across 2740 images. Compared to the literature, this is approximately half the size of the Grocery Store Dataset, but we provide significantly more images per class. There is substantial room for improvement here, given more time, but our model provides an acceptable demonstration of the benefits of machine vision.

While mounting to the top of the appliance allows for application across the kitchen environment and avoids the problems raised by the internal fridge environment, an overhead view can provide challenges for detection. Moreover, a limited and atypical view limits the benefit of external image data. A larger dataset produced for our needs will help address these issues.

Several project limitations are solvable with more images. APIs for step-wise image data collection are in research and development [21]. Adapting these APIs could support data collection in the future. Open-sourcing the project and inviting community contributions using such APIs could rapidly scale dataset growth and improve future models' accuracy and reliability.

5.2 A Reflection on Methods

We find that our design approach produced an appropriate and technically benchmarked solution to a well-defined research problem. In addition, our hybrid approach to minimum-viable-product prototyping facilitated the delivery of a complete working prototype comfortably within the project timeframe, allowing for elaborate features and exploration. Abiding by the critical path proved essential to the project outcomes and achievements, and we strive to continue this approach in future.

Henceforth, we aim to approach machine learning with more rigorous evaluation tools. For example, coding the entire TensorFlow model or using Google's AutoML service would unlock the full benefits of Tensorboard's model analysis tools.

5.3 Next steps

There are several technical developments necessary to widen the impact of our system. A larger dataset, improved modelling, and hardware are the first steps towards reliable integration into a home environment. Where expiry dates are not specified, we could achieve further autonomy by estimating them automatically. Physical sensors could offer their benefits in conjunction with our system. For example, integrating load cell sensors could identify when users open packaged items and how much of an item is left, or gas sensors could identify the prevalence of particles released during biological decay. In addition, we could apply the benefits of machine learning to other aspects of the system. For example, a model could learn user shopping and waste habits over time to reduce prevent waste from overbuying, applying the full advantages of machine learning to ideas seen in the literature.

We consider how we might scale image dataset creation. APIs are in development, tailored to rapid image data collection. Applying these methods with but a handful of users will quickly gather a wealth of data. While manual voluntary labelling is always possible, Amazon Mechanical Turk is a standard commercial and research tool for faster, more scalable labelling at an affordable price.

5.4 Impact

We show that machine vision makes minimising the entry barriers for food inventory management methods feasible. Furthermore, our system is the first of its kind, providing users with more comprehensive data via integrated decay recognition. The individual wastes an estimated 69kg of edible food yearly; our system could aid the reduction of this waste to zero. This reduction represents a saving of £210 per individual per year and the reduction of methane in landfills, a Greenhouse Gas with 72 times the warming potential compared to carbon dioxide over 20 years [29]. Moreover, this minimises the waste of the embodied energy within the edible food we throw away. Unlike commercial smart fridges, this intervention is affordable, and non-intrusively builds upon appliances users already possess.

6 CONCLUSION

There is no shortage of reasons to end food waste. While the UK has the largest percentage of under 15-year-olds living in severe food insecurity in Europe[40], each UK household wastes approximately £720 worth of food yearly[49]. Households are responsible for 70% of the total 9.5 million tonnes of annual UK food waste, and 70% of the food they dispose of is 'edible' [49]. In addition, food waste makes up an immense 9% of global anthropogenic GHG emissions [23]. Food management behaviours that previously reduced domestic waste during lockdown now take too much time to be maintained in regular life [48]. Although technological interventions for domestic food management exist, they consistently require significant user input (see Appendix A Table 5) , causing a high barrier to entry and lowering the likelihood of adoption.

We propose the use of machine vision to create interventions built upon users' lives instead of insisting they change how they store their food. This paper presents a first-of-its-kind system for taking food inventory from discovery to delivery. The literature review highlighted the value of CNNs for machine vision in food tracking and the opportunity it presents for decay state recognition. We take inspiration from the successful behaviours adopted in lockdown to reinforce methods for minimising waste and present our interventions in a user-friendly globally-available web application. In contrast to the state-of-the-art, we prioritise minimal user input in taking inventory. Using an initial waterfall approach to minimum-viable-product prototyping, we conquered challenges on the critical path before taking an agile approach to iterative improvement and development. As a result, we offer users a breadth of resources, from Telegram notifications to advanced recipe suggestions. Our system is the first prototype designed for prospective end-users - the food purchaser and the home cook.

We recognise 13 food items with a mean average precision of 69.4% and an average frame rate of 6.1fps. We can recognise three of these items in various states of decay. A dataset of 2740 labelled images provides the foundations for our EfficientDet-Lite1 model, and we open-source it to encourage community contributions. To advance machine vision for domestic food inventory management, we open-source this project on GitHub. The Pocket Kitchen web app presents the food inventory, gathered with minimal user intervention, to the user. Promoting effective food management techniques, we urge users to check expiry dates, manage their fridge or cupboard, and cook creatively based on their ingredients. We present these encouragements on the web app in the form of nudges, recipe suggestions and Telegram alerts.

In the future, further additions to the dataset could create more precise food recognition models. In addition, exploring new hardware designs could offer tailored product specifications to machine learning and a more adaptable kitchen environment integration. More extensive integration of machine learning could fine-tune interventions to user behaviour. For example, it is feasible that the system could learn about the individual user's waste habits and intervene ahead of their occurrence. The system could also offer nutritional information for health monitoring and dieting. Adjusting this system design could deliver machine vision to solve other supply and demand problems or handle food management in new settings, such as offices or even supermarkets.

7 ACKNOWLEDGEMENTS

We thank Dr David Boyle for his advice, support and mentorship throughout the course of this project.

We also wish to thank Dongmyoung Lee, Ranya Aloufi and Kiyohito Kunii for their time and advice.

REFERENCES

[1] [n.d.]. BEEP - Expiry Date Tracking, Google Play Store. https://play.google.com/store/apps/details?id=com.bgpworks.beep&hl=en_GB&gl=US. Accessed: 2022-06-03.

[2] [n.d.]. Best before and use-by dates: Food.Gov. https://www.food.gov.uk/print/pdf/node/271.

[3] [n.d.]. Common Objects in Context Detection Metrics. https://cocodataset.org/#detection-eval.

[4] [n.d.]. CozZo App. https://cozzo.app/features/receipt-barcode-scanner/. Accessed: 2022-06-03.

[5] [n.d.]. DATE LABELS – WHAT DO THEY MEAN?: WRAP.

[6] [n.d.]. GitHub: VegPi Project Repository. https://github.com/mimireyburn/SIOT, note = Accessed: 2022-06-03.

[7] [n.d.]. GitHub: What's Cooking. https://github.com/jackmleitch/Whatscooking-, note = Accessed: 2022-06-03.

[8] [n.d.]. Kitche App. https://kitche.co/. Accessed: 2022-06-03.

[9] [n.d.]. Kitche on the App Store. https://apps.apple.com/gb/app/kitche/id1521215203. Accessed: 2022-06-03.

[10] [n.d.]. MakerBot Thingiverse: Raspberry Pi HQ Camera Case. https://www.thingiverse.com/thing:4420496, note = Accessed: 2022-06-03.

[11] [n.d.]. NoWaste App. https://www.nowasteapp.com/. Accessed: 2022-06-03.

[12] [n.d.]. Streamlit App: What's Cooking. https://share.streamlit.io/jackmleitch/whatscooking-deployment/streamlit.py, note = Accessed: 2022-06-03.

[13] [n.d.]. VegPi YouTube Demo Video. https://youtu.be/JMFF1UoZxIk, note = Accessed: 2022-06-03.

[14] 2021. LG InstaView Door-in-Door™ Fridge Freezer. https://www.lg.com/uk/instaview-door-in-door-fridge-freezers

[15] 2021. Samsung Smart Fridge Freezers. https://www.samsung.com/uk/refrigerators/family-hub-fridge-freezers/

[16] 2021. *Smarter FridgeCam - Smart Fridge Camera with Wi-Fi Voice Activated.* https://smarter.am/products/smarter-fridgecam

[17] 2021. *State of Hunger: Building the evidence on poverty, destitution, and food insecurity in the UK, Year Two Main Report.* Technical Report. Trussel Trust. https://www.trusselltrust.org/wp-content/uploads/sites/2/2021/05/State-of-Hunger-2021-Report-Final.pdf

[18] N J Avinash, Renita Pinto, Sowmya Bhat, R Chetan, and H Rama Moorthy. 2020. Smart Fridge for Global Users Based on IOT Using Deep Learning. In *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS).* 1–4. https://doi.org/10.1109/ICPECTS49113.2020.9337016

[19] M Raki Billah, Sayed Ahmed, and Rabeya Kamal Surovi. 2019. Degradation Analysis of Banana Using Comparative Approach. (2019).

[20] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101 – Mining Discriminative Components with Random Forests. In *European Conference on Computer Vision.*

[21] Bennet Cobley and David Boyle. 2020. OnionBot: A System for Collaborative Computational Cooking. *arXiv preprint arXiv:2011.05039* (2020).

[22] Karen A. Cooper, Tom E. Quested, Helene Lanctuit, Diane Zimmermann, Namy Espinoza-Orias, and Anne Roulin. 2018. Nutrition in the bin: a nutritional and environmental assessment of food wasted in the UK. *Frontiers in nutrition* 5 (2018), 19.

[23] Monica Crippa, E Solazzo, D Guizzardi, F Monforti-Ferrario, FN Tubiello, and AJNF Leip. 2021. Food systems are responsible for a third of global anthropogenic GHG emissions. *Nature Food* 2, 3 (2021), 198–209.
[24] Abhishek Das, Vivek Dhuri, and Ranjushree Pal. 2020. Smart Refrigerator using Internet of Things and Android. *arXiv preprint arXiv:2012.10422* (2020).
[25] Sikha Das, Samarjit Kar, and Arif Ahmed Sekh. 2020. FGrade: A Large Volume Dataset for Grading Tomato Freshness Quality. In *International Conference on Computer Vision and Image Processing*. Springer, 455–466.
[26] Matthew Edward, Kanisius Karyono, and H. Meidia. 2017. Smart fridge design using NodeMCU and home server based on Raspberry Pi 3. In *2017 4th International Conference on New Media Studies (CONMEDIA)*. IEEE, 148–151.
[27] Renato Ferrero, Mohammad Ghazi Vakili, Edoardo Giusto, Mauro Guerrera, and Vincenzo Randazzo. 2019. Ubiquitous fridge with natural language interaction. In *2019 IEEE International Conference on RFID Technology and Applications (RFID-TA)*. IEEE, 404–409.
[28] Government Office for Science. [n.d.]. Trend Deck 2021: Demographics. (Jun., 21 [n. d.]).
[29] Piers Forster, Venkatachalam Ramaswamy, Paulo Artaxo, Terje Berntsen, Richard Betts, David W Fahey, James Haywood, Judith Lean, David C Lowe, Gunnar Myhre, et al. 2007. Changes in atmospheric constituents and in radiative forcing. Chapter 2. In *Climate change 2007. The physical science basis*.
[30] Eva Ganglbauer, Geraldine Fitzpatrick, and Rob Comber. 2013. Negotiating Food Waste: Using a Practice Lens to Inform Design. *ACM Trans. Comput.-Hum. Interact.* 20, 2, Article 11 (may 2013), 25 pages. https://doi.org/10.1145/2463579.2463582
[31] Pulkit Jain and Paras Chawla. 2021. Smart Module Design for Refrigerators based on Inception-V3 CNN Architecture. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 1852–1859.
[32] Hokuto Kagaya, Kiyoharu Aizawa, and Makoto Ogawa. 2014. Food detection and recognition using convolutional neural network. In *Proceedings of the 22nd ACM international conference on Multimedia*. 1085–1088.
[33] Marcus Klasson, Cheng Zhang, and Hedvig Kjellström. 2019. A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.
[34] Jack Leitch. [n.d.]. Using BeautifulSoup to help make beautiful soups. *Medium* ([n. d.]). https://jackmleitch.medium.com/using-beautifulsoup-to-help-make-beautiful-soups-d2670a1d1d52
[35] Simon Mezgec and Barbara Koroušić Seljak. 2017. NutriNet: a deep learning food and drink image recognition system for dietary assessment. *Nutrients* 9, 7 (2017), 657.
[36] S. Miniaoui, S. Atalla, and K. F. Bin Hashim. 2019. Introducing Innovative Item Management Process Towards Providing Smart Fridges. In *- 2019 2nd International Conference on Communication Engineering and Technology (ICCET)*. 62–67. https://doi.org/10.1109/ICCET.2019.8726900 ID: 1.
[37] S. Miniaoui, S. Atalla, and K. F. Bin Hashim. 2019. Introducing Innovative Item Management Process Towards Providing Smart Fridges. In *- 2019 2nd International Conference on Communication Engineering and Technology (ICCET)*. 62–67. https://doi.org/10.1109/ICCET.2019.8726900 ID: 1.
[38] Amavi Djimido Noutchet. 2013. Novel user centric rfid fridge design. *Computer and Information Science* 6, 2 (2013), 151.
[39] Defra Press Office. 2020 [Online]. Household food waste to be collected separately by 2023 and 50,000 city trees to be planted in Urban Tree Challenge Fund. Defra in the media Blog. https://deframedia.blog.gov.uk/2020/02/10/household-food-waste-to-be-collected-separately-by-2023-and-50000-city-trees-to-be-planted-in-urban-tree-challenge-fund/
[40] Audrey L Pereira, Sudhanshu Handa, and Goran Holmqvist. 2017. Prevalence and correlates of food insecurity among children across the globe. (2017).
[41] S. B. Prapulla, G. Shobha, and T. C. Thanuja. 2015. Smart refrigerator using internet of things. *Journal of Multidisciplinary Engineering Science and Technology* 2, 1 (2015), 1795–1801.
[42] José Rouillard. 2012. The Pervasive Fridge. A smart computer system against uneaten food loss. In *Seventh international conference on systems (ICONS2012)*. pp. 135–140.
[43] A. S. Shweta. 2017. Intelligent refrigerator using ARTIFICIAL INTELLIGENCE. In *2017 11th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 464–468.
[44] Zainal Hisham Che Soh, Jaafar Ag Ku Izzul Hamzi Ag, Siti Noraini Sulaiman, Syahrul Afzal Che Abdullah, Mohammad Nizam Ibrahim, and Bakar Aslina Abu. 2021. Fridge load management system with AI and IOT alert. In *IOP Conference Series: Materials Science and Engineering*, Vol. 1088. IOP Publishing, 012062.
[45] Mingxing Tan, Ruoming Pang, and Quoc V Le. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10781–10790.
[46] Jessica Velasco, Leandro Alberto, Henrick Dave Ambatali, Marlon Canilang, Vincent Daria, Jerome Bryan Liwanag, and Gilfred Allen Madrigal. 2019. Internet of things-based (IoT) inventory monitoring refrigerator using arduino sensor network. *arXiv preprint arXiv:1911.11265* (2019).
[47] Matthew Z Wong, Kiyohito Kunii, Max Baylis, Wai Hong Ong, Pavel Kroupa, and Swen Koller. 2019. Synthetic dataset generation for object-to-model deep learning in industrial applications. *PeerJ Computer Science* 5 (2019), e222.
[48] WRAP. 2020. *CITIZEN RESPONSES TO THE COVID-19 LOCKDOWN – FOOD PURCHASING, MANAGEMENT AND WASTE.* Technical Report. WRAP. https://wrap.org.uk/sites/default/files/2020-10/WRAP-Citizen_responses_to_the_Covid-19_lockdown_0.pdf
[49] WRAP. 2020. *Food surplus and waste in the UK – key facts.* Technical Report. WRAP. https://wrap.org.uk/sites/default/files/2020-11/Food-surplus-and-waste-in-the-UK-key-facts-Jan-2020.pdf
[50] Lili Zhu and Petros Spachos. 2021. Support vector machine and YOLO for a mobile food grading system. *Internet of Things* 13 (2021), 100359.

APPENDIX

B PHYSICAL SENSING APPROACHES TO FOOD INVENTORY MANAGEMENT

Table 5: Review of Physical Sensing

Sensing Type	Advantages	Disadvantages
Radio-Frequency identification	Simplify tracking many items, Pre-tagged bags reduce user workload but still require new habits [37]	Requires user to manually tag items, Privacy risks [38]
Barcode scanning	Gather extra data when tagging	Significant user involvement, expiry dates not integrated.
Photodiodes [26], light-dependent resistors [41] infrared sensors [24] pressure sensors [41]	Identify stock levels for solids and liquids without the privacy risks of RFID.	Sunlight interferes with photodiodes whilst the fridge is open, and transparent liquids cannot be detected [26]. Furthermore, the internal fridge environment can disrupt regular component operation [46].

C OBJECT DETECTION WITH IMAGENET IN PRACTICE

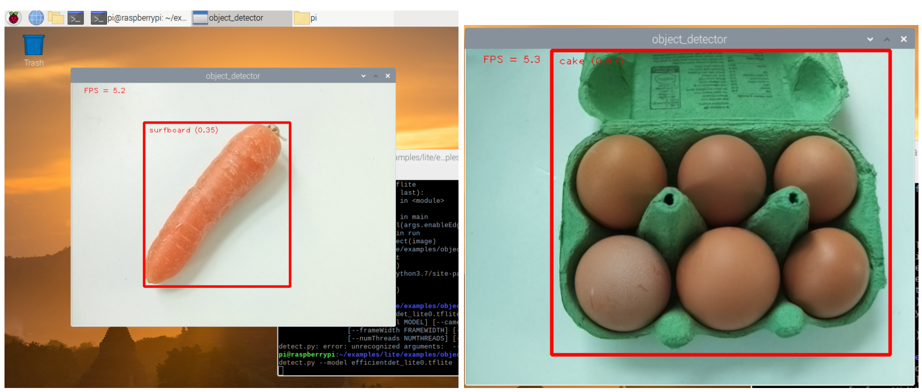

Figure 14: EfficientNet-Lite0 trained on ImageNet predicts a carrot to be a surfboard and an eggbox to be a cake

D DESIGN APPROACH

Figure 15 shows the Double Diamond approach taken to design.

E TANGIBLE BENCHMARKS FOR DESIGN OBJECTIVES

Table 6 shows design criteria and measurable benchmarks.

F RESOURCE MANAGEMENT

F.1 Data Management

We store data securely on Mongo DB Atlas.

F.2 Version Control

GitHub ensures version control. Since editing on Raspberry Pi and on a cloned GitHub repository can result in conflicting versions, the Pi will only be operated headless via ssh and VNC Viewer. All editing will be made with Visual Studio Code in a repository on a separate computer and pushed to the Pi. Although a somewhat tedious method, it offers reliable version control, as well as fast editing on a preferred GUI.

Table 6: Design Benchmarks

Design Goal	Benchmarks	Derivation	Result
Affordable	No more than £720	The annual cost of waste for a household	£187.99. We provide a breakdown of costs here.
Accessible	Designed for retrofitting to fridges and cupboards	Requires no new appliances	Clamp-oriented 3D-printed 3-component design but requires a 'lip' for attachment. Further work needs to achieve a truly universal design.
Minimal user-input automated inventory tracking	mean Average Precision (mAP) >200%/(number of classes)	At least twice as good as guessing	mAP = 69.4% Lowest classwise mAP = 22.1% Highest classwise mAP = 88.9%
Minimal user-input automated inventory tracking	0 unplanned human interventions	Minimum user input required, maximum system autonomy desired.	3 unplanned interventions in 24-hours. Higher accuracy will be achieved via dataset improvement
Responsive	Time from recognition to database addition <3s	Perceived real-time addition to the database for users	Average time from recognition to database update: 110.3ms
Small and non-invasive	Fridge/ Cupboard top form factor	<250mm^2	160mmx130mmx120mm
Small and non-invasive	Use a single-board computer whilst maintaining functionality	Reduces prototype size	Prototyping hardware comprised of Raspberry Pi 4 4GB and Coral AI accelerator.
Diversions to responsible waste stream at the point of potential expiry/waste	Preventing the use of landfills where it can be prevented	Nudges at the approach of expiry promote the use of sharing apps if the user does not intend to use the item. Nudges at the point of expiry divert users to local food waste and home composting.	Educate users on waste streams
Open-source dataset for store-bought food items	1000+ Image Dataset on GitHub	Research professionals inform the minimum quantity of images. Industry-standard finds datasets stored on GitHub	2740 labelled images
Decay state recognition	Three food items	Three accessible food items with distinct decay states	Bananas: Ripe, Unripe, Overripe. Peppers: Good, Bad. Mushrooms: Good, Bad.
Decay state recognition	mean Average Precision >100%/(number of classes) mean Average Precision >200%/(number of classes)	Better than chance	16 classes offers a 6.25% chance of accuracy when guessing. Our decay states achieve the following mean average precision. Banana-Ripe: 49.5%, Banana-Unripe: 22%, Banana-Overripe: 79.5%, Pepper: 74.6%, Pepper-Bad: 53.0%, Mushrooms, 70.7% and Mushrooms-Bad: 88.9%. We discuss these results in Section 5.1.7.
Decay state recognition	Decay states influence recipe suggestion	We wish to integrate decay state recognition into system functionality. We wish to encourage the use of items visually approaching expiry, as well as approaching predicted expiry.	Parsed ingredient state influences recipe suggestion by expiry. Negative ingredient state increases rank of respective recipe suggestions

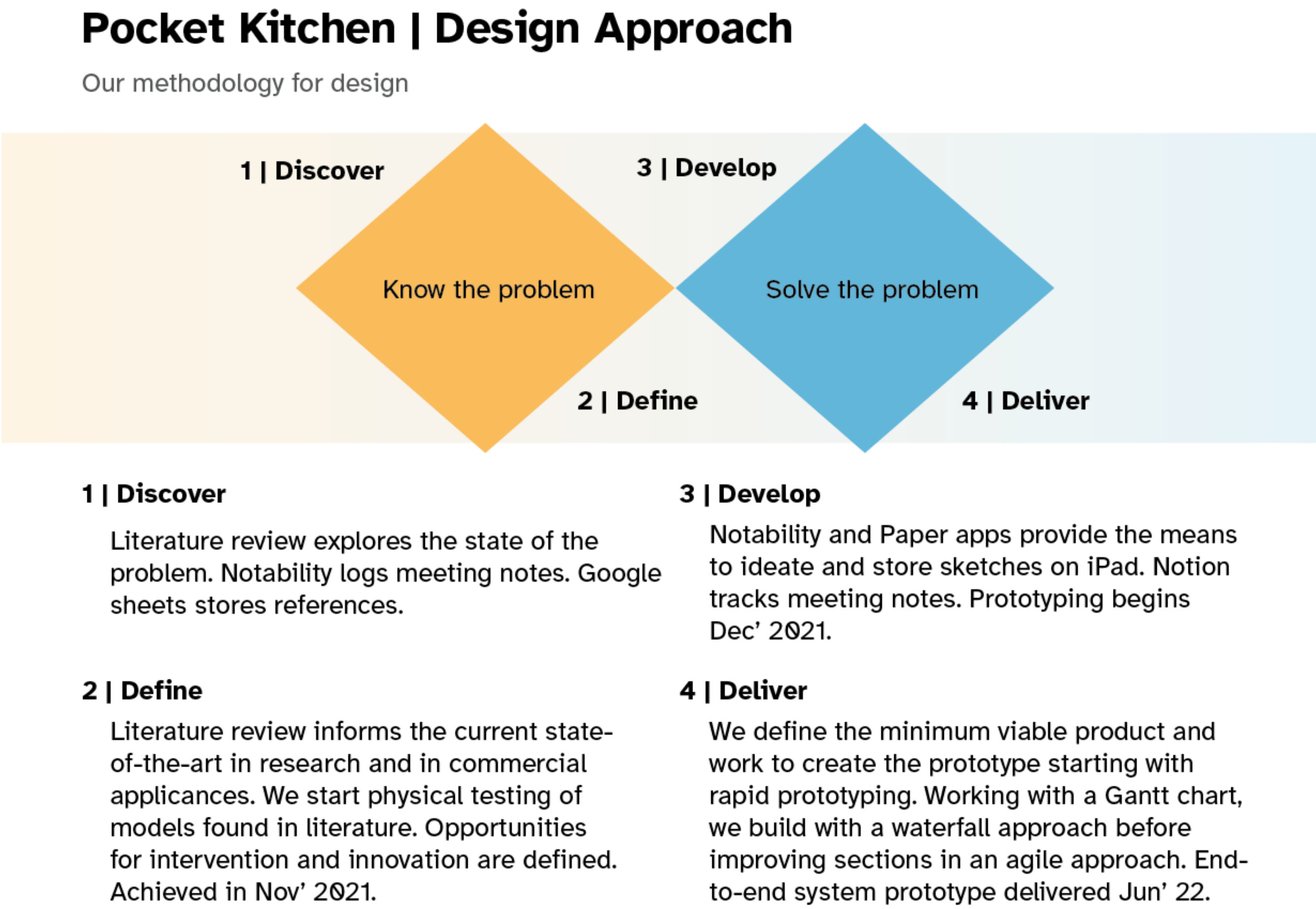

Figure 15: Project Design Approach