Team Purple Mountain Majesty

Design Specification

The goal of our project is to provide a solution for food management. We aim to track food that users purchase, and helps users reduce the amount of waste that they produce. Through researching consumer patterns, we have discovered that the main sources of food waste in families come from overestimating the amount of food to purchase, as well as forgetting to cook or use the food before it expires. Our solution to these issues is to build a smart fridge shelf to automatically track and detect wasted food.

The shelf consists of an array of sensors to collect various datapoints on user interactions with the fridge. The sensors include: four load cells to capture changes in mass, a grid of infrared emitters/receivers to detect occupied positions, a barcode reader to read UPC barcodes on food products, and a fridge door trigger that detects user interactions with the fridge. Each sensor's data is fed to a Photon microcontroller, and the data is aggregated and sent to the AWS cloud through a connected Raspberry Pi.

The scale we designed must be able to detect weight changes within a second of any item being put on it. This means the scale must be well balanced so the load is evenly spread, and the time for the weight to settle is minimized. We chose to use four load cells because it would disperse the weight of the items, and we could balance the shelf on top of each load cell in the corner. Our goal is to get accuracy of weight deltas down to within 1 gram. We are purchasing load cells that meet this specification because in order to predict food usage, high accuracy is necessary. Another limitation is that the scale must be able to support heavy loads. We selected load cells that can support over 20 pounds, so that fridge overload will not cause issues.

The infrared array is used to track which locations on the shelf are occupied. The transmitter sends out an infrared signal upwards, which will bounce off of any objects placed direct over it. The signal is then picked up by the receiver directly next to the emitter, with a wall separating the two. The array is limited to a 3.3 V power supply, which is the voltage that can be supplied by a Teensy 3.6. We will also need to connect the data lines to the GPIO of the Teensy. The goal is to have the sensors detect items in a range of 3mm, and we will cover the shelf with infrared sensors to fit this range.

The barcode scanner will be hooked onto the side of the fridge so that users can scan items easily, as they put them in the fridge. The barcode should be able to be read within 1 second, and the data is decoded and sent to the Teensy. The cloud backend is connected to a database of UPC barcodes, which provides accurate information about what item was just interacted with.

The fridge door trigger simply sends a signal whenever the door is opened. We will use this to trigger the power supply for the sensors in order to save power. The sensors do not have to be active while the door is closed because there is no interaction happening.

In the backend, all the data is streamed to AWS, where we will host the food prediction and analytics components. All the user data is stored in DynamoDB because it is easy to integrate with Alexa. The Alexa SDK provides the tools we need to interact with the user through voice, and we define specific keywords and voice lines to trigger database commands. The SDK also provides tools to send notifications to the user through Alexa, which is useful for spoilage notifications. We have a database of food spoilage timers that can be used to estimate what time the food will go bad. The user will get a notification before a spoilage event is about to occur so that they can use the food and not let it go to waste.

The mobile UI displays a list of items in the fridge that users can see when they are grocery shopping. This prevents situations where people buy items they already have and helps reduce spoilage and saves money. It also features a shopping list that users can add to through either the app or hands-free through Alexa. This allows users to plan their next shopping trip, while also giving our food prediction algorithm more data to classify items.

The prediction algorithm will use data from mass changes and location on the shelf to search previous events. We will do a nearest neighbor search of these features to previous items, to generate a list of possible candidates. This takes into account where the user usually places the item, and how heavy it is. Once the candidate list is generated, we will use the shopping list and previous purchase history to assess the confidence of each candidate. The candidate item with the highest confidence will be selected. If the confidence measure is not high enough to accurately predict an item, Alexa will be triggered to prompt the user to fill in what the food was. This provides the algorithm with a corrective opportunity, and can be used to train the weights of the features. We also have the barcode scanner to pinpoint accurate data on whatever item is scanned, and this will also be used for training.

The algorithm should improve accuracy over time with user interaction. We aim to reduce user interaction as much as possible in order to make this food management system as easy to use as possible. The goal is to reduce the number of interactions necessary to less than 15% of the number of fridge transactions that occur. This translates to an 85% prediction accuracy for the machine learning model. The goal is for the model to improve itself over time, and we will be refining the details of the algorithm over the next semester.