IntelliStorage

D3: Siyuan Li, Jason Kim, Yuma Matsuoka
18-500 Capstone Design, Spring 2024
Electrical and Computer Engineering Department
Carnegie Mellon University

Product Pitch

In today's fast-paced world, efficient management of groceries remains a significant challenge as many struggle to keep track of items and result in unneeded waste.

IntelliStorage is an innovative solution designed to simplify the way individuals and families manage their groceries. With just a quick scan, the item's brand, expiration datem and other important information will be detected and stored into our network of nodes across the home. It will give daily recommendations on what items to use, whether it be old items or items nearing expiration, personalized to your preferences.

System Architecture

The Scanner Module first takes in the Barcode and images of the Expiration date via its precepts. The barcode is looked up in a UPC database to figure out the item information, and the images are evaluated to give a high-confidence expiration date via OCR. The UPC barcode format is mainly for US items, and this is a design limitation of the product. The data will be confirmed with the user for no mistakes, and will be relayed to the Central Module.

The Central Module is a Raspberry Pi that acts as a database for the overall system. In case of failures on scanner modules or itself, it is able to recover data using the RAFT distributed consensus protocol. It gathers information from all the edge nodes and provides users with recommendations on items to use via a helpful User Interface.

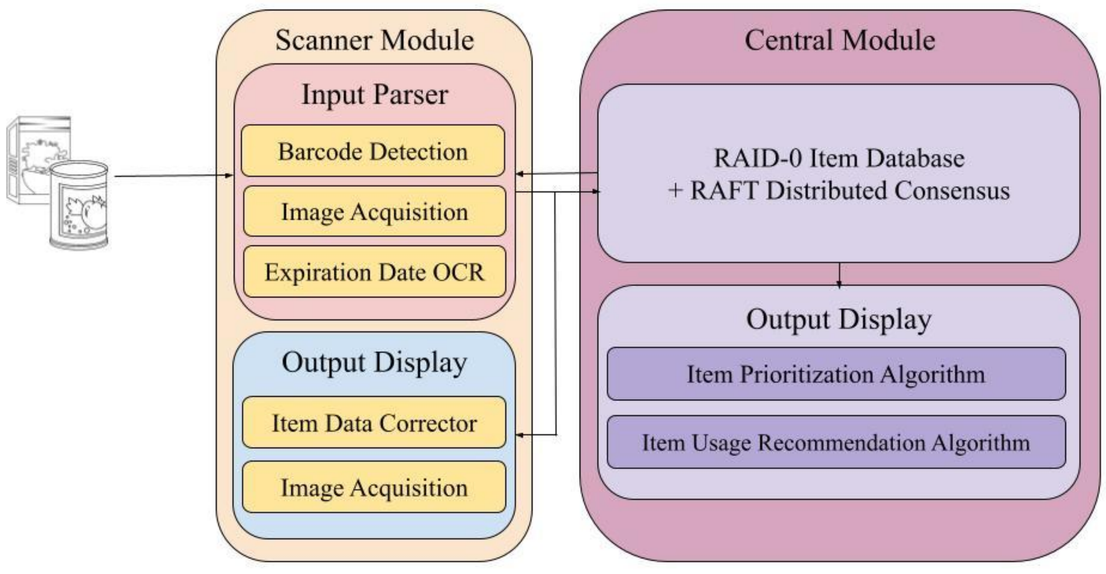

Figure 1: System Architecture

Conclusions & Additional Information

Scan the QR Code for more information!

http://www.ece.cmu.edu/~ece500/projects/S24-teamD3

Overall, we have demonstrated a solid proof of concept. Although this design's efficacy is limited by the U.S. UPC barcode format, resulting in recognition of only U.S. items, it is expandable in future works. Cascading API calls to various regional databases such as Japan's EAN barcode would help alleviate this regionality issue.
During the process we have learned and experienced the troubles of integrating various people's works. We have learned that deciding the signature of functions before implementation would help alleviate this issue. Additionally, we learned that a good user interface is hard to make, as designing and implementing a working MVP was already a bottleneck to the project.

System Description

Software:

- **UI Application**
  - Handles user recommendation requests and switch register/deregister modes
- **Optical Camera Recognition (OCR)**
  - Recognizes expiration date via customized algorithm and extracts via regex
- **Distributed Consensus Algorithm**
  - Determines shared and agreed global state even through dropped messages and node failures

Hardware:

- **Raspberry Pi 4/5**
  - Computing unit for both modules
- **Barcode Scanner**
  - Item barcode percept for item information
- **Arducam Camera**
  - Camera percept for expiration date
- **5-7" Touchscreen**
  - Interface for manually overriding item information, requesting recommendations.

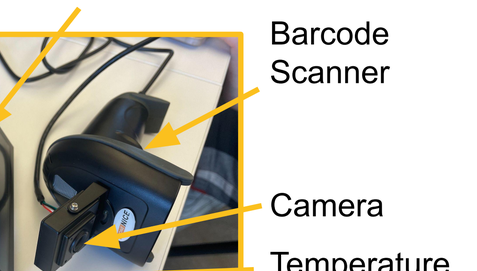

Figure 2: OCR Detection Workflow

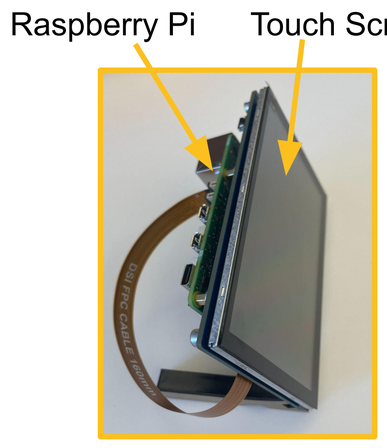

Figure 3: Central Module

Figure 4: Scanner Module

System Evaluation

- **Single Module Testing**
  - Mimic use case and scan items in continuously at boundary conditions
- **Multi Module Testing**
  - Scale single module testing and turn off modules to mimic power loss
- **Usability Testing**
  - Check if interface hits use case lag upper bound

Requirement #1 (Item Registration & Tracking)	Requirement #2 (Scaling)	Requirement #3 (Ease/Accessibility of Use)
Target: **>90%** read-in accuracy Result: **73%** accuracy Cause: Blurry saturated photos from shaky hand ❌	Target: **40 items** per storage space Result: **>100 items** ✅	Target: Store information within **1 sec** of scanning Result: **100ms** ✅
Target: **30 degree** scanning angle Result: max **35 degrees** ✅	Target: **3 storage spaces** per network Result: ✅	Target: Display info within **500 ms** Result: ✅
Target: **15-25 cm** scanning distance Actual: **5-30 cm** ✅	Target: **10 sec** Synchronization Result: **500ms** ✅	Target: Daily report of expiring item Result: ✅
Target: **4 sec** registration time Actual: **3 sec** ✅	Target: Data consistency Result: ✅	Target: **<5 min** setup node Result: **4 min** ✅

Table 1: Quantitative Results

Electrical & Computer ENGINEERING
Carnegie Mellon