

Food Tracker

Jaeyoon Choi, Zhengze Gong, Keaton Drebes

Use Case

- You're grocery shopping, and you forgot to make a list of things to buy.
- You don't remember whether you have milk, or eggs, or that one ingredient for that one recipe.
- You bought something and completely forgot about it, leaving it in the fridge for ages

Use Case Requirements

- Power requirements should be reasonable for a smart appliance.
 - <10 W peak
- Supported items (subject to change)
 - Apple, Banana, Orange, Milk, Eggs, Bread, Yogurt, Cheese, and Cereal
- Regular updates to web application, at least every 10 min
- Should be able to track combined inventory across multiple cabinets/locations
- CV component must have greater than 85% accuracy when identifying items
 - No individual item should have < 60% accuracy
 - Failure to identify an item is preferred to misidentification
 - Misidentification (identifying one item as something else) should happen <5% of the time
- Musts have the ability to handle unsupported items/orientations of items
 - Send photo of unsupported item to user through web app so user can identify it
- Web component should be accessible
 - Individual accounts for individual users
 - Inventory will update every time the cabinet/fridge door is opened/closed
 - will be simulated with a button press for purposes of testing
 - Users can create/Import expected normal inventory, or select a number of recipes, and generate a shopping list based on what they lack

Technical Challenges

- CV component
 - Availability of training data (If using CNN)
 - May need to manually generate a set for testing regardless
 - Computational power of embedded device
 - Getting the required accuracy
 - Communication with Web Application
- Web component
 - Communication with CV component
- Integration of components










Solution Approach

- Jetson Nano for hardware
 - Seems sufficiently powerful for our purposes
 - Meets power requirements
 - Ethernet port for communication with web app.
- Embedded camera for CV
 - Probably should be around 8 megapixels
 - All cameras should have similar resolutions
- Open CV for software
 - Using ORB for feature detection
 - May use CNN with info from imageNet if unable to get desired results with ORB
- Django for web-app
- Communication will just be posting JSON.

Testing, Verification, and Metrics

- Obtain an inventory of whatever grocery items we support, randomly arrange all possible combinations in view of the CV component, record successes and failures.
 - Orientations of items will be what you would reasonably expect, IE, you wouldn't store milk upside down, but you may store either right side up, or on it's side
- Measure the power consumption when running compute-intensive tasks: < 10W
- Measure the average power consumption for regular usage

Tasks/Division of Labor/Gantt Chart P1

		Name	Duration	Start	Finish
1		Find/make training data	7 days	2/9/22 8:00 AM	2/17/22 5:00 PM
2		Get accuracy requirements using only training data	14 days?	2/18/22 8:00 AM	3/9/22 5:00 PM
3		Get accuracy/timeframe requirements running on Jetson Nano	7 days?	3/10/22 8:00 AM	3/18/22 5:00 PM
4		Get accuracy/timeframe working for photos taken from embeded camera	7 days?	3/21/22 8:00 AM	3/29/22 5:00 PM
5		Develop MVP Web app with with one user with phony data	14 days?	2/15/22 8:00 AM	3/4/22 5:00 PM
6		Develop Web app to correctly update with when posted JSON	7 days?	3/16/22 8:00 AM	3/24/22 5:00 PM
7			14 days?	3/30/22 8:00 AM	4/18/22 5:00 PM
8		Wireframe Web app	4 days?	2/9/22 8:00 AM	2/14/22 5:00 PM
9		Write OpenCV code	7 days?	2/9/22 8:00 AM	2/17/22 5:00 PM
10		Generate library of photos similar to what will be taken from embedded ...	7 days?	2/27/22 8:00 AM	3/8/22 5:00 PM
11		Ensure OpenCV code works on Nano	5 days?	2/18/22 8:00 AM	2/24/22 5:00 PM
12		Write posting code	4 days?	2/20/22 8:00 AM	2/24/22 5:00 PM
13		Ensure posting code Works on Nano	5 days?	2/25/22 8:00 AM	3/3/22 5:00 PM
14		Ensure camera works with Nano	2 days?	2/20/22 8:00 AM	2/22/22 5:00 PM
15		Develop MVP Web app with several users with phony data	7 days?	3/7/22 8:00 AM	3/15/22 5:00 PM
16		Prep for final presentation	4 days?	4/19/22 8:00 AM	4/22/22 5:00 PM
17		Total Timeframe	53 days?	2/9/22 8:00 AM	4/22/22 5:00 PM

Tasks/Division of Labor/Gantt Chart P2

