Neo Jun Wei, William Chen, Steven Shi

Fridge Genie

Use Case

Problem

- Keeping track of items in your fridge is tedious and inefficient
- Though smart fridges exist with some similar function, they cost up to $1000 more than regular models

Solution

- Our product will include a mobile application linked to moveable (actuated) cameras and computer vision models, allowing users to effortlessly monitor what's in their fridge
- The app will also have a suite of functions(recipe recommendations/ grocery list generation)

Areas

- Software Systems(Mobile App)
- Hardware Systems (Embedded Devices)
- Signals and Systems(Computer Vision/ Wireless Communication)

Requirements

Detection and classification accuracy

- 90%
- Correct and reliable item tracking requiring minimal user intervention

Image capture and CV inference pipeline latency

- 15s total
- Minimize time required to generate list of ingredients detected from when image capture is remotely prompted

Recipe retrieval time

- **3 relevant recipes** within **2 seconds** of request, with **>=70%** of ingredients being currently available
- Quick, relevant meal suggestions help reduce food waste and saves time

Technical Challenges

Image quality

- Extrinsic to the CV model quality affecting inference
- Affected by
  - Camera quality
  - Image resolution
  - Lighting
  - Camera position
- We will need to install our own light source when capturing the current fridge contents, especially when the fridge is closed

Model suitability

- A performant CV model will be necessary to achieve high accuracy of image detection
- The model will need to be generalizable across variations in
  - Fridge layouts
  - Package design variations
  - Item orientations
  - Lighting
  - Item background complexity

Image capture and CV pipeline latency

- Scanning and inferring the fridge contents is time-intensive
- Potential chokepoints:
  - Wireless transmission to microcontroller to activate camera and lighting
  - Physical camera movement in capturing panorama
  - Image transmission latency for cloud-based inference

Solution Approach - CV Model

- Use object detection to classify various ingredients
- We plan to utilize a deep learning CV model written in PyTorch, with cloud-based inference
- OpenCV to handle image processing
- YOLOv5 model trained on existing datasets of fridge ingredients captured under various conditions
- Might require location-specific datasets given regional variations in packaging
    - E.g., butter wrapped in translucent packaging in the U.S. vs foil in Europe

Solution Approach - Motorized Camera Module

- HD1080p cameras with stepper-motor based mounts
- Cameras placed on each shelf at the back of the fridge
- Physically scans the fridge's interior through translating across the fridge width (marked in red)
- Captures images at sufficiently high resolution while minimizing blind spots and item occlusion
- Single camera on motorized mount reduces hardware costs compared to multiple fixed cameras at different locations

Solution Approach - Embedded Device

- Local embedded device, likely a Raspberry Pi
  - Real-time camera control
  - Data transmission to the cloud
  - Can also perform CV inference locally if cloud transmission presents unacceptable delay

- Performs data compression to reduce transmission delay

Solution Approach - Mobile App

- Platform for user input to interact with the embedded system

- Provides other key features
  - User-prompted CV inference
  - Recipe suggestions
  - Grocery list generation
  - Notifications(scan completion/ expiration updates)

- Ensures user convenience and quick data verification

Testing, Verification & Metrics

- We will adopt a unit-testing methodology, testing individual components and modules

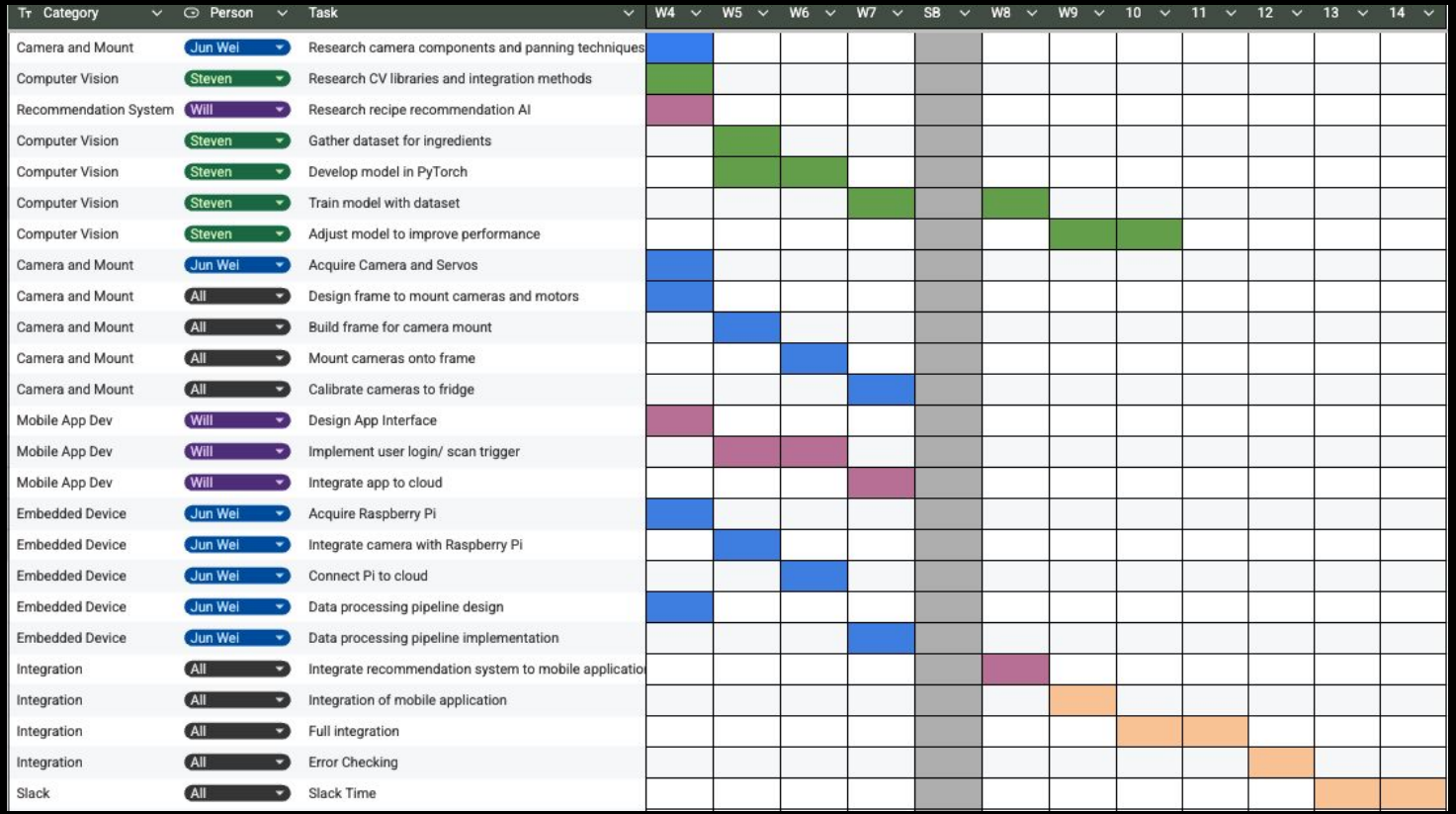
- After integrating the various components, we will conduct validation testing with a fridge filled with ingredients

- Robustness testing:
  - Simulate use across a given period of time
  - Varied fridge access patterns
    - Adding and replacing existing items
    - Removing items
  - Placing items in various parts of the fridge
  - Varying object orientations

Tasks and Division of Labor

Team Member	Responsibilities
Jun Wei	Camera, actuator, and microcontroller integration
Will	Mobile app and app features
Steven	CV models for ingredient detection and classification
Everyone	Unit testing and integration of their respective modules

Schedule



Category	Person	Task	W4	W5	W6	W7	SB	W8	W9	10	11	12	13	14
Camera and Mount	Jun Wei	Research camera components and panning techniques												
Computer Vision	Steven	Research CV libraries and integration methods												
Recommendation System	Will	Research recipe recommendation AI												
Computer Vision	Steven	Gather dataset for ingredients												
Computer Vision	Steven	Develop model in PyTorch												
Computer Vision	Steven	Train model with dataset												
Computer Vision	Steven	Adjust model to improve performance												
Camera and Mount	Jun Wei	Acquire Camera and Servos												
Camera and Mount	All	Design frame to mount cameras and motors												
Camera and Mount	All	Build frame for camera mount												
Camera and Mount	All	Mount cameras onto frame												
Camera and Mount	All	Calibrate cameras to fridge												
Mobile App Dev	Will	Design App Interface												
Mobile App Dev	Will	Implement user login/ scan trigger												
Mobile App Dev	Will	Integrate app to cloud												
Embedded Device	Jun Wei	Acquire Raspberry Pi												
Embedded Device	Jun Wei	Integrate camera with Raspberry Pi												
Embedded Device	Jun Wei	Connect Pi to cloud												
Embedded Device	Jun Wei	Data processing pipeline design												
Embedded Device	Jun Wei	Data processing pipeline implementation												
Integration	All	Integrate recommendation system to mobile applicatio												
Integration	All	Integration of mobile application												
Integration	All	Full integration												
Integration	All	Error Checking												
Slack	All	Slack Time												