HOUSTON
CITY COLLEGE

Fruit Freshness Analyzer
Tier 1

Benjamin LaCount II

The Problem

- **What real-world problem are you solving?**
  - A large portion of fresh fruit in US homes spoils before being eaten, making it a top source of household food waste.
- **Who cares about this problem?**
  - Consumers (households) who lose money, and anyone concerned with food waste and its environmental impact.
- **Why is it important?**
  - This wastes billions of dollars annually, costs families money, and contributes significantly to food in landfills.

A Solution

- **What will your system do?**
  - It will analyze a photo of a fruit and estimate the number of days remaining before it spoils.
- **How will it solve the problem?**
  - It gives consumers a clear, actionable "use by" timeline and provides smart storage tips (like the banana skin turning black in the fridge) to help them prioritize and consume fruit before it goes bad.
- **One sentence description:**
  - A computer vision tool that predicts a fruit's remaining shelf-life and provides storage tips from a single photo and whether the fruit is stored in the pantry or a fridge.

Technical Approach

- **CV Technique:** Image Classification
- **Model:** ResNet50 (pre-trained on ImageNet)
- **Framework:** PyTorch
- **Why this approach?**

ResNet50 is a powerful, proven architecture for classification. Using a pre-trained model (transfer learning) will help us achieve high accuracy on our specialized fruit dataset quickly.

Data Plan

- **CV Model Training Data (FruitVision):**
  - **Source:** A public dataset: "FruitVision" hosted on Mendeley Data.
  - **Size:** 81,000+ augmented images of 5 different fruit types.
  - **Labels:** "Fresh," "Rotten," and “Formalin-Infused” (pre-labeled).
- **Spoilage Timeline Data (FoodKeeper):**
  - **Source:** USDA "FoodKeeper" Dataset (data.gov).
  - **Usage:** Provides the ground-truth spoilage timelines (e.g., "Apples: 1-2 months refrigerated"), keyword maps, and contextual storage tips.

System Diagram

- **[Input 1]** User uploads fruit image.
- **[Input 2]** User selects storage: "Pantry" or "Refrigerator".
- **[Processing 1]** ResNet50 model classifies image and outputs a freshness score (e.g., "90% fresh").
- **[Processing 2]** Logic uses the CV label ("Apple") and user input ("Refrigerator") to query the FoodKeeper data.
- **[Processing 3]** System retrieves the correct "Total Days" (e.g., 42) and "Storage Tip" (e.g., "Store whole").
- **[Processing 4]** Remaining Days = Total_Days * (Freshness_Score / 100)
- **[Output]** Displays result: "Est. 38 days remaining. Tip: Store whole in the fridge."

Success Metrics

- How will you measure success?
  - By the model's ability to accurately classify the fruit's state.
- **Primary Metric:** Classification Accuracy (correctly identifying "fresh" vs. "rotten").
- **Target:** 85-90% accuracy on the test dataset.
- **Secondary Metrics:** Inference speed (processing time per image).
- **Success = 85%+ accuracy AND <1 second inference time.**

Week-by-Week Plan

Week	Tasks	Milestone
10 (Oct 30)	Finalize proposal, acquire & preprocess datasets (FruitVision, FoodKeeper).	Dataset ready
11 (Nov 6)	Set up environment, begin training baseline model.	Baseline model trained
12 (Nov 13)	Analyze baseline, tune hyperparameters, retrain.	Model accuracy improved
13 (Nov 20)	Finalize model. Parse FoodKeeper data (timelines, keywords, tips) and develop mapping logic.	Core logic complete
14 (Nov 27)	Build simple demo (e.g., Colab notebook), prepare slides.	Demo ready
15 (Dec 4)	Practice and deliver final presentation.	Presentation complete

Challenges & Backup Plans

- **Challenge:** What if the model's confidence score (e.g., 80%) doesn't reliably map to a specific number of days?
- **Plan B:** Simplify the output. Instead of "3-5 days," we will provide a "freshness" category (e.g., "Peak," "Good," "Use Soon").
- **Challenge:** What if the classification accuracy is too low?
- **Plan B:** Apply more aggressive data augmentation (color jitter, blur) and, if needed, try a more modern model architecture (like EfficientNetV2).

Resources Needed

- **Compute:** Google Colab (with free T4 GPU)
- **Tools:** PyTorch
- **Cost:** $0