Fridge Vision – Colab Notebook Journey

From data loading to YOLOv8 training, evaluation, and inference.

Notebook Overview

- Google Colab notebook powering the Fridge Vision project

- Uses YOLOv8 for object detection on grocery/fridge items

- Includes sections for setup, data prep, training, and testing

Environment & Setup

1

- Mount Google Drive and link to dataset

2

- Install/import Ultralytics YOLOv8 and other libraries

3

- Configure file paths and runtime (GPU)

Data Preparation

- Load images and label files
- Organize train/validation/test splits
- (Optional) apply data augmentation
- Visually inspect sample labeled images

YOLOv8 Training

- START FROM A PRE-TRAINED YOLOV8 CHECKPOINT

- POINT TO A CUSTOM DATASET YAML

- SET EPOCHS, BATCH SIZE, IMAGE SIZE, AND LEARNING RATE

- RUN TRAINING AND MONITOR TRAINING/VALIDATION LOSS

Evaluation & Metrics

- Evaluate on validation/test sets

- Look at precision, recall, and mAP

- Examine detection examples to check quality

- Identify confusion cases (e.g., apples vs tomatoes)

Inference & Demos

- Run the trained model on new fridge images

- Display images with bounding boxes and labels

- Save example outputs for use in reports and slides

Challenges Captured in the Notebook

- MODEL STRUGGLED TO TELL TOMATOES FROM APPLES

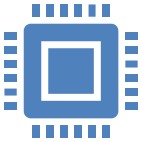
- LIMITED TOMATO TRAINING DATA AND VISUAL SIMILARITY

- TRAINING/DEBUGGING TOOK LONGER THAN PLANNED

- REAL FRIDGE IMAGES WERE MESSIER THAN EXPECTED

Lessons & Reflections

- High-quality, balanced data is essential



- Real-world environments need robust models

- Time estimates for training must include debugging

- The notebook became a central log of experiments

Conclusion

- The Colab notebook documented the full ML lifecycle for Fridge Vision, from raw data to trained YOLOv8 model and final demos.
- Working through the project reinforced practical skills in computer vision, experimentation, and iterative improvement.