Rapid Tech Skills Program-Project Documentation (Data Science and AI)

Dedan Kimathi University of Technology

BY:

Mburu Margaret Wangari

KPSK-DeKUT-2-16

PROJECT TITLE:

CherryGrade: Coffee Cherry Ripeness Classification to Reduce Harvest Losses for Nyeri's Smallholder Farmers.

MAY 2025

1. Introduction

Smallholder farmers in Nyeri County lose 30–40% of their yields due to poor ripeness selection. This results in KES 2,000+ lost per acre annually and 78% of harvests downgraded. Visual sorting methods are subjective, inconsistent, and error-prone. Lab-grade tools (costing over KES 5,000) are unaffordable for most farmers. Climate variability makes ripeness timing even more unpredictable. Existing Agri-tech focuses on soil health and pest control, not post-harvest quality. There's a gap for affordable, AI-powered tools to support data-driven cherry grading.

Despite the economic potential of the specialty coffee market, smallholder farmers lack access to effective, scalable tools for objective ripeness assessment. Manual grading is inconsistent and prone to error, resulting in lower yields, downgraded quality, and missed market opportunities. Existing Agri-tech solutions do not solve the core problem of post-harvest quality detection, a task that requires not just precision, but real-time, field-ready intelligence. There is a pressing need for a solution that demonstrate how computer vision and object detection can be applied in practical agricultural settings.

CherryGrade is a computer vision-based tool that detects and classifies coffee cherry ripeness using object detection. Built on the YOLOv8n model, trained to identify 5 stages: unripe, ripe, semi-dry, dry, overripe. Users upload an image via a Flask-based interface; the system returns instant classification. Provides actionable feedback to guide sorting or harvesting decisions. Optimized with FP16 quantization and model caching for fast, lightweight deployment. Useful as a learning tool for AI and agriculture, with real-world dataset and application. Scalable for integration into Agri-robots, drones, or automated sorting stations.

2. Methodology

2.1) Data Collection

The data is sourced from Kaggle as Version 1.0 – Cherries on plant, with a natural background. It contains images and segmentation masks where cherries are classified based on their ripeness level.

Label Distribution:

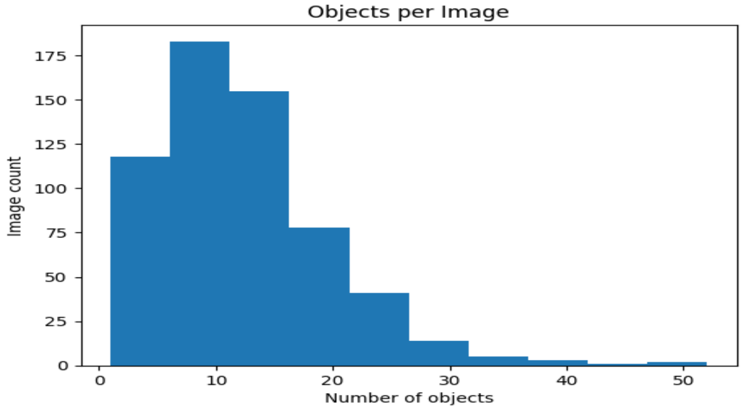
- Unripe: 8,207 - Green cherries that are still immature. (Not ready for harvest.)
- Ripe: 1,428 - Cherries that have matured properly (ready to be picked).
- Semi-dry: 874 - Cherries that have started drying but are not completely dry.
- Dry: 301 - Cherries that have dried significantly (possibly shriveled).
- Overripe: 234 - Cherries that are past the ripe stage, overmature. (Can affect coffee quality negatively.)
- Total Masks: 11,044

Dataset Split:

- - Training: 70% (600 images) - Validation: 20% (174 images)
- - Testing: 10% (85 images) - Total Images: 859

2.2) Data Preprocessing

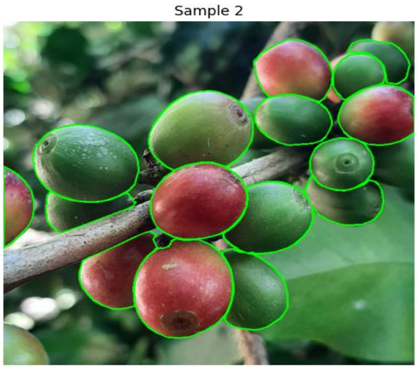
Images were formatted into YOLO-compatible structure. Duplicates and corrupt labels were cleaned. YOLOv8 auto-handles normalization and resizing, but manual inspection of annotations and class balance was conducted as shown in the diagram. The graph shows an Image-level Object Count Imbalance, the variation in the number of labeled objects per image.



- The majority of images contain 5 to 15 objects, with a peak around 8–10 objects.
- Some images have over 30 objects, but those are rare.
- This distribution is right-skewed, meaning most images have fewer objects, with a few having very many.

2.3) Exploratory Data Analysis (EDA)

The dataset was analyzed to understand the distribution and characteristics of coffee cherry maturity stages. Sample visualizations of annotated images confirmed the presence of correctly labeled bounding boxes for various classes such as ripe, semi ripe, overripe, and dry. While YOLOv8 internally handles resizing and normalization, further analysis of class distribution and bounding box sizes can provide insight into potential class imbalances and model focus areas. Future iterations may include quantitative class balance charts and bounding box statistics for deeper insights.



2.4) Modeling Approach

The model used is YOLOv8n.pt (YOLOv8 nano) for lightweight, fast object detection. Epochs: 100, Batch size: 16, Image size: 640. Loss functions: Box loss, classification loss, and DFL loss. I trained the model on CPU (Collab), later moved to GPU for improved performance. Output model: best.pt. which I later exported to use in deployment.

Sample of how the model trained and run, as well as the output model best.pt.

```
                 all        174       2365      0.697      0.709      0.746      0.707

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     99/100      12.5G     0.1959     0.1631     0.7968        108        640: 100% 38/38 [00:46<00:00,  1.22
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:05<00:
                   all        174       2365      0.689       0.72      0.738      0.702

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
    100/100      12.5G     0.1954     0.1648     0.7959         59        640: 100% 38/38 [00:46<00:00,  1.23
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:05<00:
                   all        174       2365      0.695      0.712      0.736        0.7

100 epochs completed in 1.591 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 136.7MB
Optimizer stripped from runs/detect/train/weights/best.pt, 136.7MB
```

2.5) Evaluation Metrics

YOLOv8 provides precision, recall, and mean Average Precision (mAP@0.5).

Performance Metrics summary:

P (Precision) - % of predicted boxes that are correct
R (Recall) - % of ground-truth boxes correctly detected
mAP50 - Mean Average Precision at 50% IoU (basic accuracy)
mAP50-95 - Averaged over IoU thresholds 0.5 to 0.95

Class	Images	Instances	P	R	mAP50	mAP50-95
all	174	2365	0.701	0.743	0.768	0.721
dry	22	64	0.840	0.822	0.854	0.787
overripe	13	31	0.413	0.419	0.428	0.387
ripe	45	317	0.704	0.839	0.833	0.779
semi_ripe	41	181	0.655	0.735	0.773	0.737
unripe	154	1772	0.892	0.901	0.950	0.916

Unripe class is performing best (high precision and recall).

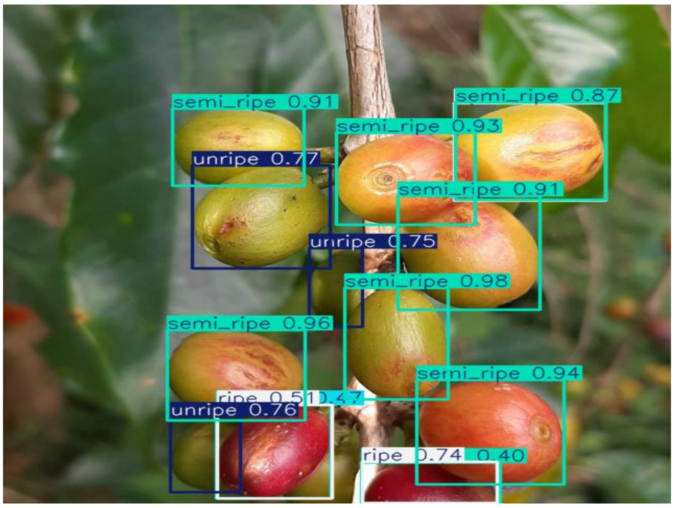
Overripe is weakest — possibly due to few examples (only 31).

Overall mAP50-95: 0.721 → solid performance for a small dataset.

3. Results and Analysis

Model learned to distinguish cherry ripeness stages effectively despite class imbalance. Unripe and ripe classes showed high confidence. Overripe and dry classes had slightly lower precision due to limited data.

Sample results:



4. Deployment

The trained best.pt model was deployed via a Flask app. Users upload images through a web interface; YOLOv8 model returns classified images with bounding boxes. Optimized for speed using model caching and FP16 quantization. Future-ready for integration into Agri-robots, mobile apps, or drone feeds.

5. Conclusion and Recommendations

This project has demonstrated how AI can reduce post-harvest losses using real-time object detection.

Recommendations:

In future this project can be scaled more to perform much better: E.g.
- Gather more balanced data across ripeness stages
- Deploy model to edge devices such as mobile or Raspberry Pi
- Collaborate with cooperatives to field-test in Nyeri

6. Appendices

Link to the Dataset:

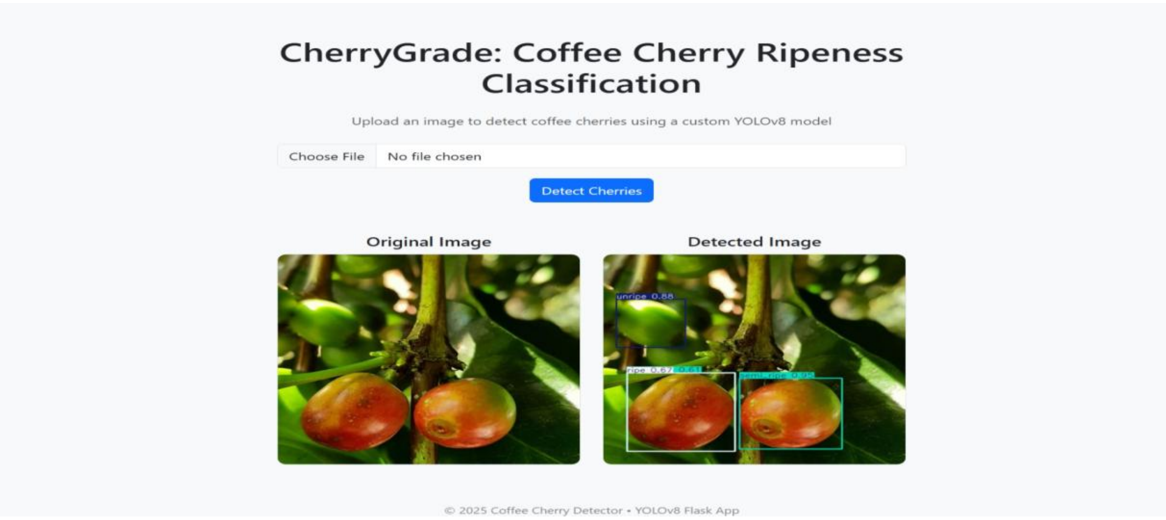
Kaggle Dataset: https://www.kaggle.com/datasets/cienciacafeto/coffee-fruit-maturity

Notebook Link:
https://colab.research.google.com/drive/1UspxZAUtAFo7kqSbefvWmCH4Odahnon2#scrollTo=lRUqI5OFue2k

GitHub Link to the source code:

https://github.com/MaggieGeorges/Cherry-Grade

Flask Interface Screenshot:



Ministry of Investments Trade and Industry
State Department for Industry
THE WORLD BANK
Spineberg
E4IMPACT FOUNDATION