PROJECT ABSTRACT

Fruit Ripeness Detection using CNN

A built-to-order final-year project that predicts fruit type and ripeness stage (unripe, ripe, overripe) from a photo with a multi-task MobileNetV3 CNN, built on Fruits-360 plus a curated ripeness-labelled set, with training notebook, TFLite export, demo and documentation.

01. Title

Fruit Ripeness Detection using CNN

02. Introduction / Background

Fruit grading on packing lines and in mandis is still done by eye: slow, subjective and inconsistent between graders and shifts. Ripeness is fundamentally a vision problem — colour distribution, spotting and surface texture change predictably as fruit ripens. A lightweight CNN can make the same grading decision on every fruit, and if it is small enough it can run on a phone in the field. This project trains a MobileNetV3-Small backbone with two output heads — fruit type and ripeness stage — combining the published Fruits-360 dataset (131 classes) with a curated, ripeness-labelled banana and mango set.

03. Problem Statement

Single-task classifiers answer only 'which fruit'; commercial grading needs 'which fruit and how ripe'. Student builds rarely handle the labelling discipline ripeness needs (annotator agreement, ambiguous boundaries) or the deployment constraint (phone-sized model). This project addresses both with a multi-task setup and a TFLite export.

04. Objectives

- Label ~3,600 banana/mango photos unripe / ripe / overripe with two annotators.
- Train a multi-task MobileNetV3-Small: fruit-type head (12 classes) + ripeness head (3 classes).
- Evaluate both heads with accuracy and confusion matrices on a held-out split.
- Export a ~3.4 MB TFLite model for on-device grading.
- Deliver an interactive grading demo with sample photos.

05. Proposed Methodology

A shared MobileNetV3-Small backbone feeds two softmax heads trained with a weighted multi-task loss. Inputs are 224×224 normalised images. Augmentation uses rotation and cropping; colour jitter is kept mild because colour carries the ripeness signal — the report includes this ablation reasoning. AdamW with early stopping is used for training.

06. System Architecture / Working

- Input: fruit photograph.
- Preprocessing: resize 224×224, normalise.
- Model: MobileNetV3-Small → head 1: fruit type (12); head 2: ripeness (3).

- Output: fruit type + ripeness stage with probabilities; demo adds a shelf-life hint.
- Deployment: TFLite export for Android on-device inference.

07. Hardware / Software Requirements

- Hardware: laptop/PC; free cloud GPU for training; any Android phone for the TFLite model.
- Software: Python 3, TensorFlow/Keras, scikit-learn, Jupyter Notebook.
- Demo: any modern web browser.

08. Expected Outcomes

- Multi-task model with design-target ripeness accuracy ~93% and fruit-type accuracy ~97% (reported honestly after training).
- Per-head confusion matrices showing where ripe/overripe boundaries blur.
- TFLite model, demo, report, PPT and viva Q&A.;

09. Applications

- Packing-line and mandi grading assistance.
- Consumer app for checking fruit ripeness while shopping.
- Classroom demo of multi-task learning.

10. Future Scope

- More fruit types and internal-quality (sugar/firmness) sensor fusion.
- Video-based ripeness tracking in storage.
- Integration with a sorting actuator.