

Fruit Quality Grading using CNN

An EfficientNet-B0 CNN grading fruit quality (fresh/unripe/rotten) on the Fruits-360 dataset, with a grading-station demo and batch sorting. Delivered built-to-order with report, PPT and viva kit.

01. Title

Fruit Quality Grading using CNN

02. Introduction / Background

Packing lines still grade fruit largely by human eyes — slow and inconsistent across shifts. A camera plus a classifier grades in milliseconds with perfect consistency. The Fruits-360 dataset (90,483 images, 131 classes including fresh and rotten variants) is the ideal training ground. This project fine-tunes EfficientNet-B0 with colour-jitter augmentation for lighting robustness, analyzes per-group accuracy and confusion pairs honestly, documents a TFLite edge-export path, and wraps it all in a grading-station demo.

03. Problem Statement

Tutorial builds report one accuracy number on clean images and ignore the deployment realities: lighting variation, the bruised-vs-rotten boundary, and edge hardware limits. The project addresses this with lighting-robust augmentation, per-group and confusion-pair analysis, an efficiency-first backbone choice, and a documented TFLite export for the conveyor camera unit.

04. Objectives

- Fine-tune EfficientNet-B0 on Fruits-360 with colour-jitter augmentation.
- Evaluate with overall, per-group accuracy and confusion-pair analysis.
- Build a grading-station demo with grades, probabilities and lane sorting.
- Provide batch grading with automatic lane assignment.
- Export the model to TFLite for edge deployment.
- Deliver the complete student kit: notebook, weights, demo, report, PPT and viva Q&A.;

05. Proposed Methodology

The pipeline has three stages. (1) Data: Fruits-360 at 224×224 with the published split; augmentation with colour jitter, random crop and flip. (2) Training: ImageNet pre-trained EfficientNet-B0 fine-tuned with Adam for 30 epochs, early stopping on validation accuracy. (3) Evaluation and demo: single test evaluation with per-group metrics, TFLite export, and wiring to the grading-station demo.

06. System Architecture / Working

The notebook exports trained weights and the class-to-grade mapping. The single-file demo lets the student pick a fruit sample, runs grading with a scan animation, and shows the grade (A / B / reject) with confidence and the full class-probability distribution plus quality notes. Batch mode grades 8 fruits and sorts them into premium / ripening / reject

lanes. Analytics views show training curves and per-group accuracy.

07. Hardware / Software Requirements

- Python 3, PyTorch/TensorFlow, torchvision
- EfficientNet-B0 (ImageNet pre-trained)
- NumPy, Matplotlib, scikit-learn (metrics)
- Jupyter Notebook
- Fruits-360 dataset (Mureman)
- HTML5, CSS, JavaScript (demo)
- GPU recommended for training

08. Expected Outcomes

- A trained grader with design-target test accuracy 97.8%.
- A grading-station demo with batch lane sorting.
- Per-group accuracy and confusion-pair analyses.
- A complete report, PPT and viva Q&A; on efficient CNNs for industry.

09. Applications

- Teaching transfer learning and efficient architectures.
- Demonstrating augmentation for lighting robustness.
- Academic study of edge AI export paths.

10. Future Scope

- On-line fine-tuning with packing-line photos.
- Multi-fruit conveyor tracking.
- Internal-defect (non-visual) sensor fusion.
- Ripeness-day regression as an extension.