PROJECT ABSTRACT

Tea Leaf Quality Classification using CNN

A transfer-learned ResNet/EfficientNet classifier that grades plucked tea leaves from images with Grad-CAM explanations, targeting ~85%+ validation accuracy measured by the buyer-run notebook.

01. Title

Tea Leaf Quality Classification using CNN

02. Introduction / Background

Tea quality starts at the pluck: two leaves and a bud commands a premium price, while coarse or damaged leaves drag a whole batch down. In practice, plucked leaf is graded by experienced buyers at the collection center, eyeballing each basket in seconds — fast but subjective, and impossible to keep consistent across buying days and seasons. Leaf images carry exactly the visual cues graders use: color, size, uniformity, damage marks. This project builds a CNN classifier that learns those cues: a pretrained ResNet or EfficientNet backbone, fine-tuned on public tea-leaf image collections, starts from strong general vision features and only needs to learn leaf-specific grading distinctions. Grad-CAM overlays make each prediction inspectable, showing which leaf regions pushed the model toward its grade — the visual explanation that turns a black-box classifier into something defensible in a viva.

03. Problem Statement

Buying-center leaf grading is subjective and inconsistent, yet the visual cues it relies on are learnable from images. The need is a CNN classifier that grades plucked-leaf photographs into quality classes, explains each grade with a visual overlay, tolerates real photo variation through augmentation, and evaluates itself with accuracy, F1 and a confusion matrix computed on a validation split — framed as a visual-appearance grader, never as a replacement for cupping.

04. Objectives

The objectives are to fine-tune a pretrained ResNet or EfficientNet CNN on public tea-leaf image collections; to build an augmentation pipeline for photo variation; to classify plucked-leaf photos into configurable quality grades; to explain predictions with Grad-CAM overlays; to break down class-wise performance and review misclassified samples; and to evaluate with accuracy, F1 and the confusion matrix on the validation split via the included notebook.

- Fine-tune a pretrained ResNet or EfficientNet on tea-leaf image collections.
- Build an augmentation pipeline (rotation, flip, color jitter).
- Classify plucked-leaf photos into configurable quality grades.
- Explain each grade with Grad-CAM overlays.
- Analyze class-wise performance and review misclassified samples.
- Evaluate with accuracy, F1 and confusion matrix on the validation split during the build.

05. Proposed System / Working

Plucked-leaf images are loaded, resized to the classifier's input size and normalized with OpenCV. Training applies augmentation — rotation, flips, color jitter — so the model tolerates collection-center photo variation. A pretrained ResNet or EfficientNet backbone is fine-tuned on the labeled collections, with the classifier head retrained for the grade classes. At inference, the model outputs a quality grade with a probability score; Grad-CAM overlays visualize the image regions that drove the prediction. The demo app serves grades with explanations for uploaded leaf photos. The evaluation notebook computes accuracy, F1 and the confusion matrix on the validation split and runs a misclassified-samples review with overlays.

- OpenCV loading, resizing and normalization of leaf images.
- Augmentation: rotation, horizontal flip, color jitter.
- Transfer learning with a pretrained ResNet/EfficientNet backbone.
- Grade prediction with per-class probability scores.
- Grad-CAM explanation overlays per prediction.
- Notebook evaluation: accuracy, F1, confusion matrix, misclassification review.

06. Technologies / Components Used

The project uses Python 3.10 and PyTorch for training and inference, with torchvision supplying the pretrained ResNet and EfficientNet backbones. Public tea-leaf image collections in ImageFolder layout provide the training data; OpenCV handles preprocessing and Grad-CAM overlays; NumPy and scikit-learn compute metrics and the confusion matrix; a Jupyter notebook performs the evaluation; and a Flask app serves the demo with a leaf-image upload interface.

- Python 3.10, PyTorch (CNN training and inference).
- torchvision (pretrained ResNet and EfficientNet backbones).
- Public tea-leaf image collections (ImageFolder-style datasets).
- OpenCV (preprocessing, Grad-CAM overlays).
- Jupyter notebook (evaluation with accuracy and F1).
- NumPy, scikit-learn (metrics and confusion matrix).
- Flask demo app with leaf-image upload interface.

07. Key Features

Key features include a transfer-learned ResNet/EfficientNet classifier, training on public tea-leaf image collections, configurable quality-grade classes, Grad-CAM explanations per prediction, a photo-variation augmentation pipeline, an evaluation notebook computing accuracy, F1 and the confusion matrix, a misclassified-samples review, and a demo app for leaf photos.

- Transfer-learned CNN quality-grade classifier.
- Training on public tea-leaf image collections.
- Configurable grade classes.
- Grad-CAM overlays explaining each grade.
- Augmentation for real photo variation.
- Evaluation notebook: accuracy, F1, confusion matrix.
- Misclassified-samples review with overlays.
- Demo app for uploading leaf photos.

08. Applications / Use Cases

Tea leaf grading applies to collection-center intake triage, buyer decision support, quality auditing across buying days, and planter training. It also demonstrates transfer learning, image augmentation, and confusion-matrix error analysis as viva-ready deep-learning concepts — always framed as a visual-appearance grader, never a replacement for cupping.

- Collection-center intake triage for plucked leaf.
- Buyer decision support at purchase points.
- Quality auditing across buying days and seasons.
- Planter and grader training material.
- Teaching transfer learning and error analysis.

09. Results & Scope

The expected outcome is a working leaf-grade classifier with a design target of ~85%+ validation accuracy, where the actual accuracy, F1 and confusion matrix are computed by the training notebook during the buyer's training run — the report's numbers come from that run, not from a claim. The confusion matrix names the confused grade pairs explicitly, which doubles as viva error-analysis material. Scope covers the tea varieties and imaging styles labeled in the public collections; documented limits include out-of-coverage varieties and regions, overlapping adjacent grades, and visual-only assessment — liquor taste, aroma and chemical quality need human cupping and lab analysis. Future scope includes in-field phone-camera grading, variety-specific models, and multi-grade regression scoring.

- Design target ~85%+ validation accuracy; actual metrics computed by the notebook during the buyer's training run.
- Confusion matrix naming the confused grade pairs.
- Scope: varieties and imaging styles labeled in the public collections.
- Documented limits: out-of-coverage varieties, adjacent-grade overlap, visual-only assessment.
- Future scope: phone-camera grading, variety-specific models, regression scoring.

10. Conclusion

Transfer learning turns plucked-leaf grading from a subjective eyeball task into a measurable, explainable classification pipeline, and this project delivers it as a complete auditable experiment: augmentation, fine-tuning, Grad-CAM explanations, misclassification review, and a notebook that measures accuracy, F1 and the confusion matrix on the validation split. Built to order, it ships with source code, the evaluation notebook, report, presentation, and viva preparation material.

11. References

The project cites the ResNet architecture, the ImageNet dataset, and the Grad-CAM interpretability method.

- K. He, X. Zhang, S. Ren, and J. Sun, 'Deep Residual Learning for Image Recognition,' Proc. CVPR, 2016.
- J. Deng et al., 'ImageNet: A Large-Scale Hierarchical Image Database,' Proc. CVPR, 2009.
- R. R. Selvaraju et al., 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,' Proc. ICCV, 2017.