Reliable Fruit Ripeness Classification from RGB Images via Calibrated Transfer Learning

Daniel Cevallos V.
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Guayaquil, Ecuador
dacevall@espol.edu.ec

Erasmo García
Facultad de Ingenierías Aplicadas y Desarrollo Industrial
Universidad Internacional del Ecuador (UIDE)
Guayaquil, Ecuador
ergarcia@uide.edu.ec

Wilton Agila
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Guayaquil, Ecuador
wagila@espol.edu.ec

Holger Cevallos
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Guayaquil, Ecuador
hcevallo@espol.edu.ec

***Abstract*—We address nine-class fruit-ripeness recognition from commodity RGB images using a calibrated transfer-learning pipeline. EfficientNet-B0 is fine-tuned in two stages (with MobileNetV2 as a lightweight fallback) using class-weighted cross-entropy with label smoothing and Adam with decoupled weight decay (AdamW); early stopping and stochastic weight averaging (SWA) promote generalization. At inference, mild test-time augmentation (TTA; horizontal flip and central crops) is combined with post-hoc temperature scaling (TS) to improve probabilistic reliability. On the Kaggle *Fruit Ripeness* dataset (apples, bananas, and oranges across {unripe, fresh/ripe, rotten}), the model achieved a Top-1 accuracy of 0.9735 with TTA (Top-3: 0.9995), micro-averaged receiver operating characteristic–area under the curve (ROC–AUC) of 1.000 (macro: 0.999), and reduced expected calibration error (ECE) from 0.044 to 0.004. Errors concentrated in visually adjacent maturity states within the same fruit (e.g., unripe vs. fresh/ripe), and Gradient-weighted Class Activation Mapping (Grad-CAM) visualizations indicated that decisions relied on semantically meaningful regions. Implemented solely with standard TensorFlow/Keras components and commodity augmentations, the recipe provides a reproducible, deployment-oriented baseline that balances accuracy, efficiency, and calibrated confidence for food-quality applications in low-resource settings.**

***Index Terms*—Fruit ripeness classification; transfer learning; EfficientNet-B0; temperature scaling; probabilistic calibration; test-time augmentation; Grad-CAM; computer vision.**

I. Introduction

Automated assessment of fruit ripeness from images is a longstanding challenge in computer vision (CV), with direct implications for post-harvest logistics, shelf-life prediction, and food-waste reduction across the supply chain. Manual inspection remains the dominant practice in retail and distribution, yet it is subjective, labor-intensive, and difficult to scale. This paper addresses the problem of *multi-class visual classification of fruit ripeness*—distinguishing *unripe*, *ripe*, and *rotten* fruit—from commodity RGB imagery. We focus on a publicly available benchmark, the Kaggle *Fruit Ripeness* dataset [1], and study a practical transfer-learning pipeline implemented with modern convolutional neural networks (CNNs).

The task presents several technical difficulties. First, *intra-class variability* is substantial: the appearance of "ripe" can differ markedly across fruit types and lighting conditions. Second, *inter-class boundaries* are subtle: early rot, bruising, and specular highlights can mimic texture cues of unripe or ripe states. Third, real-world datasets often exhibit *class imbalance* and *spurious correlations* (e.g., background and container cues), which can bias a model's decision process and degrade calibration. These challenges motivate architectures with strong image priors (to generalize under limited or imbalanced data) and training protocols that explicitly account for imbalance and probability calibration.

To this end, we adopt pre-trained CNN backbones—EfficientNet-B0 [2] and, as a fallback under identical code paths, MobileNetV2 [3]—fine-tuned in two stages on the target dataset. EfficientNet-style scaling remains a competitive baseline for accuracy under constrained compute, while MobileNetV2 provides a lightweight alternative for edge scenarios. We position these choices within a broader landscape that includes more recent families (e.g., EfficientNetV2 [4] and ConvNeXt [5]) to situate our study with respect to contemporary CNN design, although our implementation focuses on the former two for reproducibility and deployment simplicity.

From an optimization standpoint, we combine AdamW (Adam with decoupled weight decay) [6] with label smoothing [7] to stabilize training under limited supervision. We address class imbalance via class-weighted loss and mild class-specific boosting. To enhance reliability at inference time, we employ test-time augmentation (TTA)—ensembling predictions over simple geometric and crop transforms [8]—and post-hoc temperature scaling for probability calibration [9]. The latter is crucial because well-calibrated confidence estimates enable trustworthy human–AI collaboration; recent large-scale analyses have revisited calibration failure modes and evaluation practices in modern deep networks [10]. We further apply stochastic weight averaging (SWA) to widen optima and

improve generalization [11]. Although our pipeline uses TTA rather than adaptive test-time training (TTT) strategies, we discuss TTT as a complementary, more recent line of work that adapts models under distribution shift during inference [12].

Beyond accuracy, we quantify performance with Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC), as well as Precision–Recall (PR) analysis [13], [14]. We also report Expected Calibration Error (ECE) and reliability diagrams to assess probabilistic fidelity, which is particularly relevant when predictions gate downstream actions (e.g., automatic sorting or triage). To improve transparency, we employ Gradient-weighted Class Activation Mapping (Grad-CAM) [15] to visualize spatial evidence supporting each decision, helping to diagnose spurious attributions (e.g., background dominance) and to communicate model behavior to stakeholders.

This paper makes the following practical contributions:

1) A reproducible transfer-learning recipe for fruit-ripeness recognition on the Kaggle dataset [1], comprising two-stage fine-tuning of a pre-trained CNN (EfficientNet-B0 or MobileNetV2) with class-weighted, label-smoothed optimization under AdamW.
2) A lightweight reliability stack at inference time—test-time augmentation (TTA) and temperature scaling— that maintains or slightly improves top-k accuracy while substantially improving probability calibration.
3) A diagnostic and interpretability suite (ROC/PR, ECE, confusion analysis, and Grad-CAM) that surfaces per-class failure modes and supports responsible deployment.
4) An implementation aligned with deployment constraints, using only commodity augmentations and standard TensorFlow/Keras components, to ease replication and porting to edge devices.

While recent architectures (e.g., EfficientNetV2 [4] and ConvNeXt [5]) may yield further gains, our results indicate that carefully tuned baselines—augmented with principled calibration and ensembling—offer an attractive accuracy–efficiency trade-off for food-quality applications. The proposed pipeline aims to be immediately useful in low-resource settings (e.g., smartphone or single-GPU environments) where operational simplicity and confidence quality are as important as raw accuracy.

The remainder of this paper is organized as follows. Section II details the Methodology, including data preparation, model architecture, optimization, inference-time ensembling, and calibration. Section III presents the Results, with ablations and interpretability analyses. Section IV concludes the paper.

II. Methodology

This section details the end-to-end pipeline employed to classify fruit ripeness using a CNN. We first describe the dataset and split policy, followed by data preparation and augmentation, the transfer-learning architecture, optimization and regularization strategies, inference with TTA, post-hoc probability calibration via TS, and the evaluation protocol. All experiments were implemented in TensorFlow/Keras with deterministic seeds set to 42 for NumPy, Python, and TensorFlow.

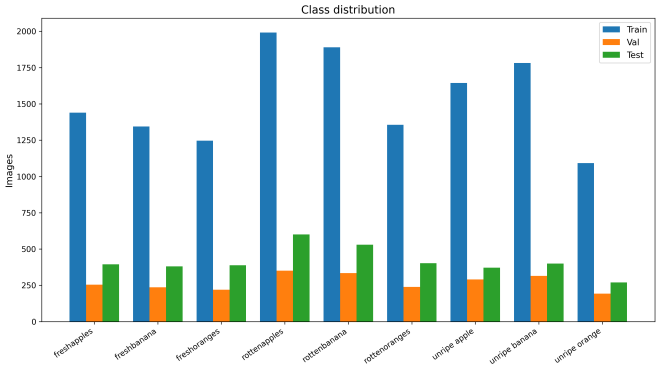


Fig. 1. Class distribution across Train/Validation/Test after stratified splits.

A. Dataset and Split Policy

We use the public "Fruit Ripeness: Unripe, Ripe, and Rotten" dataset from Kaggle [1], which contains nine visual categories spanning three fruit types (apple, banana, orange) at three ripeness stages (unripe, fresh/ripe, rotten). Images provided in the official `train/test` folders were parsed programmatically; when a test split was not present, we created a stratified pseudo-test split by sampling 15% of the training pool, and a stratified validation split of 15% from the remaining training pool (Fig. 1). Class names are inferred from directory names to avoid manual label coupling. Optional extra images under `dataset/` are merged into the training pool when available.

B. Data Preparation and Online Data Augmentation

Images are lazily streamed with the `tf.data` API, decoded as RGB, resized to 224×224, and batched by 32. To improve robustness and reduce overfitting, we apply light, label-preserving stochastic augmentations within the graph: horizontal flip, rotation up to $\pm 10\%$ of a full turn, zoom $\pm 10\%$, and contrast jitter $\pm 10\%$. Backbone-specific preprocessing (mean/scale) is injected through the corresponding `preprocess_input` function.

C. Network Architecture and Transfer Learning

We adopt transfer learning with *EfficientNet-B0* [2] as primary backbone, falling back to *MobileNetV2* [3] if ImageNet weights are unavailable (identical head in both cases). The backbone is initially frozen and followed by global average pooling, a dropout layer (rate 0.35), and a dense softmax classifier over $K = 9$ classes. Stage 1 trains only the classification head; Stage 2 performs fine-tuning by unfreezing the top 30% of backbone layers while keeping early layers frozen to preserve general features.

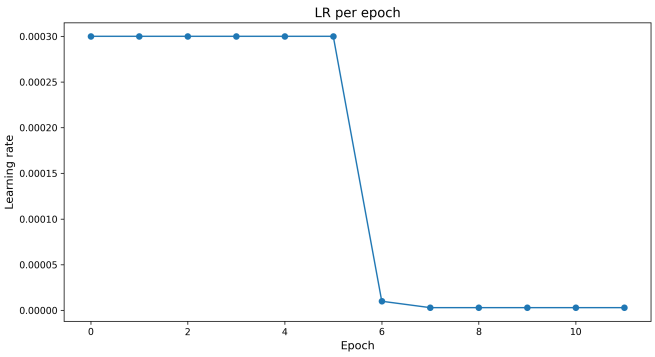


Fig. 2. Learning-rate (LR) schedule recorded during both stages.

D. Loss, Class Rebalancing, and Label Smoothing

We minimize a weighted cross-entropy with label smoothing. Given one-hot labels $\mathbf{y}_i \in \{0,1\}^K$ and model probabilities $\hat{\mathbf{p}}_i$, we use the smoothed target

$$\tilde{\mathbf{y}}_i = (1-\varepsilon)\mathbf{y}_i + \frac{\varepsilon}{K-1}\left(\mathbf{1} - \mathbf{y}_i\right), \quad \varepsilon = 0.05,$$

and per-sample loss $\ell_i = -\sum_{k=1}^{K} \tilde{y}_{ik} \log \hat{p}_{ik}$ [7]. To address imbalance, we compute class weights w_c using inverse-frequency reweighting (scikit-learn), and apply a light domain-informed boost for rare confusions (*unripe apple*, *unripe orange*, factor 1.10). The final objective is

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} w_{y_i}\,\ell_i.$$

E. Optimization, Scheduling, and Regularization

We train with AdamW [6] (Stage 1: learning rate 3×10^{-4}, weight decay 5×10^{-5}; Stage 2: 1×10^{-5} and 1×10^{-5}, respectively) and gradient clipping at ℓ_2 norm 1.0. A Reduce-on-Plateau scheduler monitors validation loss with factor 0.3 and patience of two epochs (minimum LR 10^{-6}). Early stopping (patience 5, best-weight restore) and model checkpointing on validation accuracy prevent overfitting. During Stage 2, we enable SWA [11], accumulating an equal-weight mean of consecutive snapshots from the second half of fine-tuning. Figure 2 shows the learning-rate trace collected per epoch.

F. Inference with Test-Time Augmentation (TTA)

At inference, we apply TTA by averaging predictions over four transforms: identity, horizontal flip, and two central crops (retain 95% and 90% of the area) resized back to 224×224. The final probability vector for an image is the arithmetic mean of the four forward passes [8].

G. Post-hoc Probability Calibration

To improve probabilistic interpretability, we calibrate validation predictions with TS [9]. Because logits are not exposed in the deployed path, we temper softmax probabilities using a power transform,

$$\hat{\mathbf{p}}^{(T)} = \frac{\hat{\mathbf{p}}^{\,1/T}}{\sum_{k=1}^{K} \hat{p}_k^{\,1/T}},$$

TABLE I
TEST PERFORMANCE SUMMARY (TOP-1/TOP-3 AND LOSS).

Setting	Top-1 Acc.	Top-3 Acc.
No TTA	0.9700	0.9995
TTA	0.9735	0.9995
Test loss (cross-entropy): 0.4416		

and select $T^\star \in [0.5, 3.0]$ on a grid by minimizing the negative log-likelihood on the validation set. The learned $T^\star$ is then applied to test-time probabilities (after TTA).

H. Evaluation Protocol

We report top-1 accuracy and top-3 accuracy (Keras `SparseTopKCategoricalAccuracy`) on the test set. For detailed analysis, we compute per-class precision, recall, and F1-score; a normalized confusion matrix; ROC; AUC [13]; and PR curves for all classes [14]. We further assess calibration with reliability diagrams and the ECE [16]. Finally, we provide qualitative explanations via Grad-CAM visualizations [15] on both correctly and incorrectly classified, high-confidence samples.

I. Reproducibility

All results were obtained with batch size 32, image resolution 224×224, and two training stages of six epochs each. Augmentations and preprocessing were defined within the model graph to ensure determinism under a fixed random seed. Code emits CSV reports and saves the final Keras model artifact to facilitate downstream use.

III. RESULTS

This section reports the performance of the proposed fruit–ripeness classifier on the held-out test set. Unless otherwise stated, predictions were produced with TTA and probabilities were later calibrated by temperature scaling tuned on the validation split (the full implementation is provided in the code listing). We summarize results using standard metrics and plots: the confusion matrix, ROC curves and AUC, PR curves and AP, per-class precision/recall/F1, and reliability diagrams to quantify calibration via the ECE. Table I summarizes overall test performance (Top-1/Top-3 accuracy) and loss on the held-out test set. In our setting, TTA yields a modest Top-1 gain and no material change in Top-3, indicating a favorable but marginal accuracy–latency trade-off.

As a reproducibility check, all scalar results in Tables I and II match the values emitted by our script in `metrics.txt`.

A. Training Dynamics and Convergence

Fig. 3 summarizes the learning behavior across epochs. Training and validation curves evolve smoothly without signs of instability, and early stopping prevents overfitting. These plots, together with the learning-rate trace in Fig. 2, indicate a stable optimization process under the two-stage fine-tuning schedule.

TABLE II
GLOBAL RANKING AND CALIBRATION SUMMARY ON THE TEST SET.

Metric	Micro	Macro
AUC (ROC)	**1.000**	**0.999**
AP (PR)	**0.997**	n/a
Calibration (Expected Calibration Error, ECE)		
Before temperature scaling	0.044	
After temperature scaling	**0.004**	

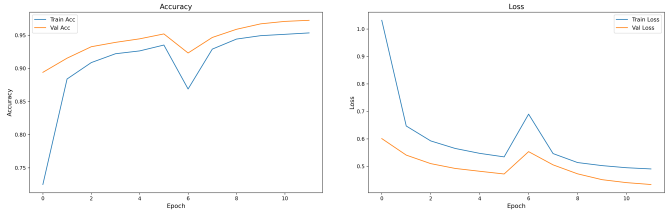


Fig. 3. Training dynamics. Left: training vs. validation accuracy per epoch. Right: training vs. validation loss per epoch. Curves show smooth convergence without divergence, consistent with the early-stopping and Reduce-on-Plateau strategy.

In addition, Fig. 4 tracks top-3 accuracy over epochs, mirroring the stable trends observed for top-1 accuracy and loss.

B. Confusion Matrix and Error Modes

Fig. 5 (row-normalized confusion matrix) shows near-perfect recognition across the nine classes. The *freshbanana* and *rottenbanana* categories achieve near-perfect recall (0.998 and 0.995, respectively), while *freshapples*, *freshoranges*, *rottenapples*, and *rottenoranges* remain in the 0.97–0.99 range. The most challenging categories are the unripe classes: *unripe apple* attains a recall of 0.889 (332/374 correct), with confusions primarily toward *unripe orange* (5.9%) and *freshapples* (3.2%); *unripe orange* reaches 0.942 (254/270 correct), with 3.7% of samples predicted as *unripe apple*. Misclassifications remain largely within the same fruit family or adjacent maturity stages, reflecting the fine-grained visual similarity among these categories.

C. ROC Characteristics

Fig. 6 reports ROC curves. The micro-averaged AUC—aggregating decisions over all classes—is **1.000**, while the macro-averaged AUC—averaging per-class ROCs—is **0.999**. These values indicate excellent separability for every class and minimal class imbalance effects on ranking performance.

D. Precision–Recall Behavior

The PR analysis in Fig. 7 shows a micro-AP of **0.997**. Precision remains essentially at 1.0 over almost the entire recall range, with only a slight drop as recall approaches 1.0. This behavior confirms that the classifier maintains very low false-positive rates even when operating at high recall.

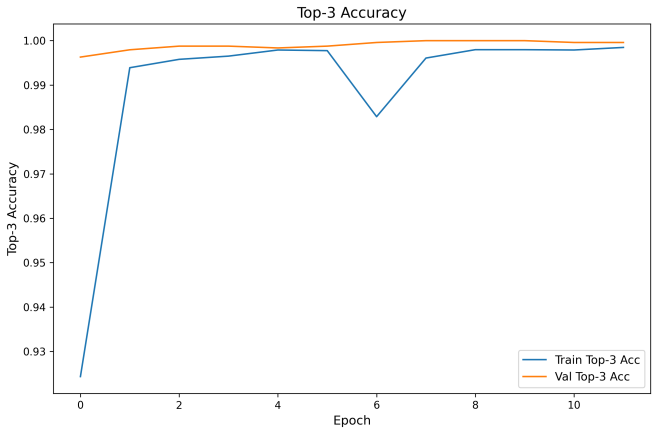


Fig. 4. Top-3 accuracy over epochs (train/validation), complementing the Top-1 trends.

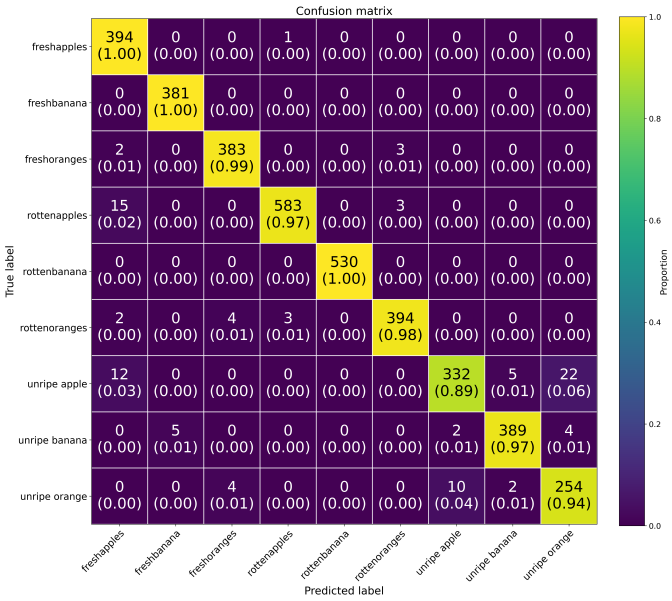


Fig. 5. Confusion matrix for the nine-class fruit-ripeness classifier.

E. Per-Class Metrics

Per-class precision/recall/F1 scores are summarized in Fig. 8. Precision is ≥ 0.96 for all categories, reaching ≈ 1.00 for *freshbanana* and *rottenbanana*. F1-scores are correspondingly high ($\approx$ 0.97–1.00) for the majority of classes. The lowest F1 occurs for *unripe apple*, driven by the aforementioned recall of 0.89; nonetheless, its precision remains high, indicating that most predictions for this class are correct when made. Overall, these results corroborate the confusion-matrix trends and highlight that remaining errors concentrate in fine-grained, visually similar maturity states.

F. Probabilistic Calibration

Reliability diagrams in Figs. 9 and 10 quantify confidence alignment. Prior to temperature scaling, the model is mildly

TABLE III
TOP CONFUSIONS ON THE TEST SET (ROW-NORMALIZED).

True → Predicted	Percent	Count
unripe apple → *unripe orange*	5.9%	22
unripe apple → *freshapples*	3.2%	12
unripe orange → *unripe apple*	3.7%	10

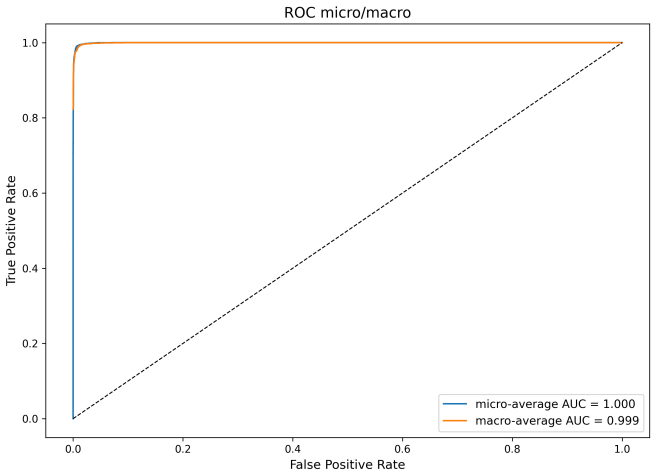


Fig. 6. Receiver Operating Characteristic (ROC) curves with micro and macro averaging.

over-confident in mid-confidence bins. After applying temperature scaling fitted on the validation split, the reliability curve closely follows the identity line and the ECE decreases to **0.004**, yielding probability estimates that are appropriate for thresholding and downstream decision-making. We compute ECE with 10 equal-width bins in confidence. The temperature $T^\star$ is selected by grid search on the validation set to minimize the negative log-likelihood, and then applied to test-time probabilities. These calibrated probabilities enable reliable thresholding for automated sorting and human-in-the-loop triage, mitigating over-confidence in ambiguous, fine-grained cases.

IV. CONCLUSIONS

1) **Effective calibrated transfer learning.** The proposed two-stage fine-tuning pipeline—centered on EfficientNet-B0 with MobileNetV2 as a lightweight fallback—in combination with class-weighted cross-entropy, label smoothing, AdamW, gradient clipping, and SWA, yielded strong performance on the nine-class Fruit Ripeness benchmark. With TTA, the model reached a Top-1 accuracy of **0.9735** (Top-3: **0.9995**) and near-perfect ranking quality (micro ROC–AUC **1.000**, macro ROC–AUC **0.999**), indicating that the learned representation discriminates the fine-grained maturity states effectively.
2) **Reliability substantially improved by TS.** Post-hoc temperature scaling reduced the ECE from **0.044** to **0.004**, aligning predicted confidences with empirical accuracy without sacrificing top-k performance. Together with mild TTA, this reliability stack provides well-calibrated probabilities suitable for threshold-based automation and human-in-the-loop triage in practical sorting workflows.
3) **Errors are concentrated in adjacent maturity states.** Confusion analysis revealed that residual mistakes mainly occur between visually neighboring categories within the same fruit family. In particular, the *unripe* classes remain the most challenging (e.g., *unripe apple* recall of 0.889, with predominant confusions toward *unripe orange* at 5.9% and *freshapples* at 3.2%). This pattern is consistent with subtle color/texture transitions and underscores the task's fine-grained nature rather than systemic model failure.
4) **Interpretability supports trustworthy deployment.** Grad-CAM visualizations indicate that decisions are driven by semantically meaningful fruit regions, rather than by spurious backgrounds or containers. This improves transparency for stakeholders, aids diagnosis of rare failure cases, and strengthens the case for deploying

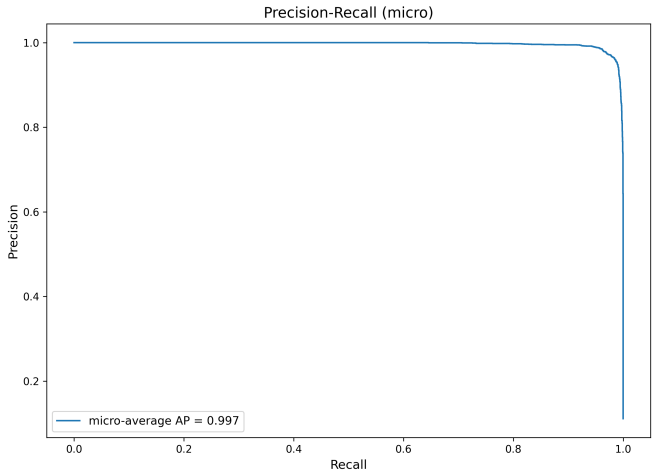


Fig. 7. Precision–Recall (PR) curves with micro-averaged summary.

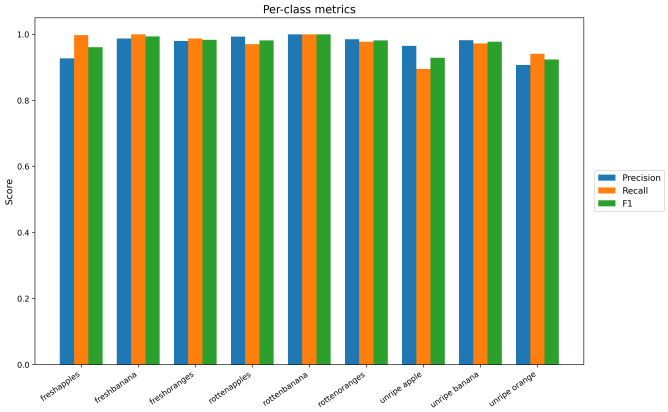


Fig. 8. Per-class precision, recall, and F1-score across the nine categories.

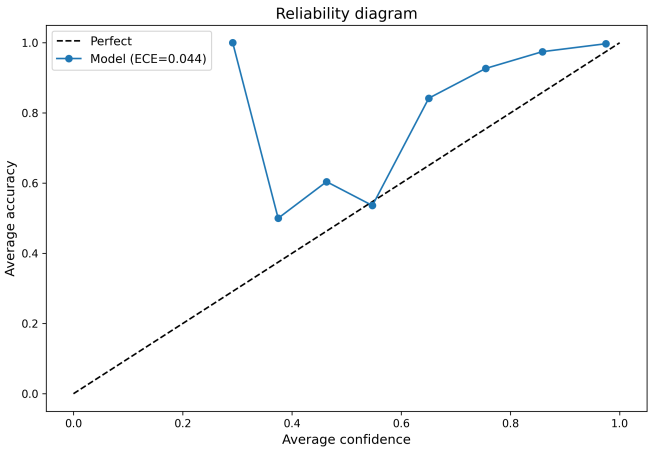


Fig. 9. Reliability diagram before temperature scaling.

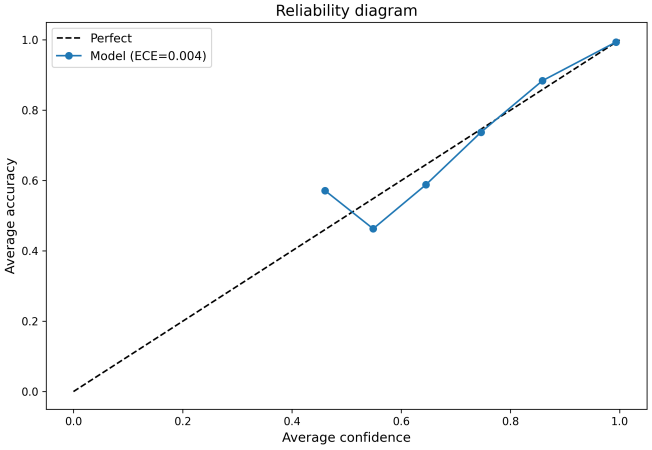


Fig. 10. Reliability diagram after temperature scaling.

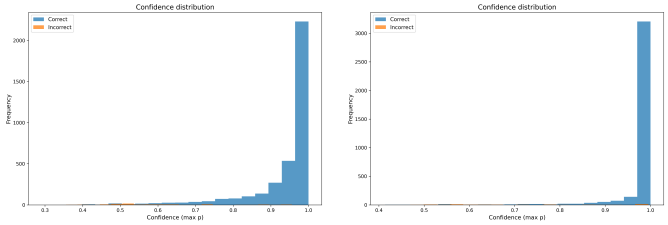
the model in settings where explainability and quality assurance are required.

5) **Practicality and reproducibility for low-resource settings.** The end-to-end recipe relies solely on standard TensorFlow/Keras components and commodity augmentations, uses deterministic seeds, and exports metrics and artifacts for auditability. The modest TTA accuracy gain, coupled with markedly better calibration, offers a favorable accuracy–latency–reliability trade-off. These properties make the approach an actionable baseline for food-quality applications on smartphones or single-GPU systems, while leaving room for future improvements via newer backbones (e.g., EfficientNetV2, ConvNeXt) or adaptive test-time training.

REFERENCES

[1] Leftin, "Fruit ripeness: Unripe, ripe, and rotten," https://www.kaggle.com/datasets/leftin/fruit-ripeness-unripe-ripe-and-rotten, 2020, accessed: 2025-08-27.

[2] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.



(a) Before calibration (b) After calibration

Fig. 11. Confidence histograms of the maximum predicted probability on the test set, before and after temperature scaling.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[5] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] M. S. Ayhan and P. Berens, "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks," *ICLR Workshop Track*, 2018.

[9] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[10] M. Minderer, J. Djolonga *et al.*, "Revisiting the calibration of modern neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[11] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

[12] Y. Sun, X. Wang, Z. Miller, A. A. Efros, and K. He, "Test-time training with self-supervision for generalization under distribution shifts," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[13] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[14] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006.

[15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[16] M. Pakdaman Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.