Received: 29 July 2026 | Revised: 7 September 2026 | Accepted: 29 September 2026 | Published online: 10 November 2026

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2026, Vol. 00(00) 1–13
DOI: 10.47852/bonviewJDSIS620211565

BON VIEW PUBLISHING

Harris Hawks Optimization–Optimized Explainable Quality Index for Tomato Grading

Van Van Chau[1], Phayung Meesad[2,*] and Minh Tuan Nguyen[1]

[1] *Faculty of Information Technology, Posts and Telecommunications Institute of Technology, Vietnam*

[2] *Department of Information Technology Management, King Mongkut's University of Technology North Bangkok, Thailand*

Abstract: Automated tomato ripeness and quality grading is essential for precision agriculture and industrial packing lines, replacing slow, subjective manual sorting. Most reported systems, however, rely on heavy deep networks or a single accuracy figure without ablation, statistical validation, or an explicit accuracy–interpretability–latency trade-off. This study proposes HHO-XQI, a Harris Hawks Optimization (HHO)-optimized Explainable Quality Index that fuses interpretable color-statistical features into a continuous Tomato Quality Index for four-class grading (Damaged, Old, Ripe, Unripe). The framework extracts 125 RGB/HSV/Lab color-statistical descriptors, fuses four base learners (logistic regression, random forest, extra trees, k-nearest neighbors) via HHO-optimized soft voting on 7,226 images, and is benchmarked against a fine-tuned ResNet18 CNN with Gradient-weighted Class Activation Mapping (Grad-CAM). Evaluation combined held-out accuracy/macro-F1, repeated cross-validation with Holm-corrected Wilcoxon tests, feature- and ensemble-composition ablation, HHO sensitivity/convergence analysis, SHapley Additive exPlanations (SHAP) attribution, and latency/training-time profiling. HHO-XQI reached 94.62% accuracy (92.90% macro-F1), significantly beating three baselines ($p < 10^{-4}$) but not extra trees alone ($p = 0.071$); the CNN reached 99.31% accuracy at comparable CPU latency (11.31 vs. 12.05 ms) yet required approximately 147 × longer training (575.7 vs. 3.92 s). These findings support a two-tier deployment: HHO-XQI as a near-instantly retrainable, SHAP-native first-stage screener, and the CNN + Grad-CAM branch as a slower-to-retrain but more accurate second-stage auditor. The study contributes a statistically validated, fully reproducible metaheuristic-fused explainable grading pipeline that fills a rigor gap common in agricultural swarm-intelligence literature.

Keywords: tomato classification, Harris Hawks Optimization, Explainable Quality Index, SHAP, Grad-CAM

1. Introduction

Tomato is one of the most widely produced and consumed horticultural crops worldwide, and its commercial value depends strongly on the ripeness stage, surface quality, and the presence of visible defects such as bruising, over-ripening, or physical damage. Manual sorting remains common in many supply chains, but it is labor-intensive, subjective, slow, and difficult to standardize across operators and shifts. Therefore, computer vision offers a useful pathway toward objective, repeatable grading for greenhouses, packing houses, and digital agriculture systems. Prior tomato studies have addressed plant disease classification, fruit ripeness recognition, tomato quality grading, yield prediction, and production forecasting using machine learning and deep learning models [1–5].

Despite this progress, four gaps remain in the tomato image classification literature. First, many high-performing deep learning models require heavy computation, large labeled datasets, and GPU-class hardware, which can be difficult to deploy on low-cost conveyor-line sorting devices [6–8]. Second, several studies have focused on either disease recognition or yield forecasting rather than the postharvest visual grading task studied here (Damaged, Old, Ripe, and Unripe classes considered jointly). Third, the role of swarm intelligence in tomato grading has not been fully explored beyond feature selection, segmentation, and hyperparameter tuning [9–12]. In particular, using a metaheuristic to fuse a heterogeneous ensemble into a single explainable, continuously ranked quality index, rather than only tuning a single model's hyperparameters, remains rare in the agricultural imaging literature. Fourth, and most importantly, from a methodological standpoint, published swarm-optimization studies in agricultural image classification typically report a single training/validation split with a single seed and rarely include an ablation study, a statistical significance test against baselines, a sensitivity analysis of the optimizer's own hyperparameters, or

***Corresponding author:** Phayung Meesad, Department of Information Technology Management, King Mongkut's University of Technology North Bangkok, Thailand. Email: phayung.m@itd.kmutnb.ac.th

© The Author(s) 2026. Published by BON VIEW PUBLISHING PTE. LTD. This is an open access article under the CC BY License (https://creativecommons.org/licenses/by/4.0/).

a feature-attribution analysis (e.g., SHapley Additive exPlanations (SHAP)) that explains *why* the optimized model makes its decisions [13]. This makes it difficult to distinguish a genuine, reproducible improvement from the noise that is attributable to a single lucky data split or seed.

This study directly addresses all four gaps. It proposes HHO-XQI, a Harris Hawks Optimization (HHO)-optimized Explainable Quality Index framework that (i) fuses four heterogeneous, computationally cheap classical learners trained on interpretable color-statistical descriptors, (ii) uses HHO to optimize the fusion weights against a validation objective rather than tuning a single classifier's hyperparameters, (iii) converts the fused classifier output into a continuous, explainable Tomato Quality Index (TQI) rather than a bare class label, and (iv) is validated with the full battery of evidence a reviewer would expect from a rigorous empirical machine learning study: repeated cross-validation, paired statistical significance testing with multiple-comparison correction, a two-axis ablation study (feature space and ensemble composition), a hyperparameter sensitivity analysis, an HHO convergence study, and a SHAP feature-attribution analysis. In addition, because tomato grading is fundamentally a visual task, this study trains a fine-tuned convolutional neural network (CNN) on the identical data split and reports Gradient-weighted Class Activation Mapping (Grad-CAM) visual explanations, providing a second, independent explainability channel and an accuracy-vs-latency-vs-interpretability comparison against the proposed lightweight framework.

The novelty of HHO-XQI relative to prior HHO-based tomato and ensemble learning studies is threefold and specific. Existing HHO applications in fruit/crop imaging use the optimizer to tune a *single* model's hyperparameters, such as the dropout rate of MangoNet [14] or the kernel/feature-selection parameters of a support vector machine (SVM) [11], and report one accuracy figure without ablation or significance testing. HHO-XQI instead uses HHO to search a 4-dimensional soft-voting *simplex* over a heterogeneous four-learner ensemble (Eqs. (1)–(2)); the optimization target is the fusion policy itself rather than any single learner's internal hyperparameters. Second, unlike prior tomato-grading ensembles that output only a discrete class label [15, 16], HHO-XQI converts the optimized fused probabilities into a continuous TQI (Eq. 4), a thresholdable risk score with a direct packing-line interpretation (Section 3) that none of the closest related studies in Table 1 provide. Third, whereas the swarm intelligence for agriculture literature typically reports a single train/test split and a single seed [13], to our knowledge, this study is the first HHO-fused tomato-grading pipeline validated with repeated cross-validation, Holm-corrected paired significance testing, a two-axis ablation, an HHO hyperparameter sensitivity sweep, and an SHAP attribution analysis in a single, internally consistent protocol (Section 5).

The main contributions are as follows:

1) We formulate a complete, reproducible grading pipeline for the Damaged, Old, Ripe, and Unripe tomato classes, built on 125 interpretable color-statistical descriptors extracted from the RGB, HSV, and Lab color spaces.
2) We adapt HHO to jointly optimize soft-voting fusion weights across four heterogeneous base learners and to construct the TQI, a continuous, explainable score that ranks samples by predicted grade risk rather than returning only a class label.
3) We validate the framework with a level of statistical rigor rarely reported in this literature, combining repeated stratified cross-validation with Holm-corrected Wilcoxon signed-rank tests, a two-axis ablation study, an HHO hyperparameter sensitivity analysis, a convergence study, and a SHAP feature-attribution analysis.
4) We train a fine-tuned ResNet18 CNN with Grad-CAM on the same data as an independent explainability channel and accuracy/latency benchmark, enabling an explicit discussion of an industrial two-tier deployment scenario that pairs fast screening with CNN-based auditing.

Table 1
Positioning of HHO-XQI against the closest related work (✓= yes; × = no; ∂ = partial)

Study	Swarm	Stat.	Abl.	XAI	Multi-br.
Shrivastav et al. [16]	✓	×	∂	×	×
Rajalaxmi et al. [14]	✓	×	×	×	×
Malini et al. [15]	✓	×	×	×	×
Quach et al. [17]	×	×	×	✓	×
Tomas et al. [18]	×	×	×	×	∂
HHO-XQI (this work)	✓	✓	✓	✓	✓

Notes: XAI: Quach et al. use Grad-CAM only; HHO-XQI uses SHAP + Grad-CAM (paired). Multi-br. (multi-branch, i.e., a classical/non-deep branch paired with a CNN branch on the same data split): Tomas et al. compare two CNN variants, which we count as partial (∂) since both branches are CNN-based; HHO-XQI pairs a classical color-statistical branch and a CNN branch on the same split (✓).

The remainder of this paper is organized as follows. Section 2 reviews related work on tomato image analysis, metaheuristic optimization in agricultural imaging, and explainable AI. Section 3 presents the HHO-XQI framework, its mathematical formulation, and the optimization algorithm. Section 4 describes the dataset, features, baselines, CNN branch, and experimental protocol. Section 5 reports the experimental results, including the ten validation analyses listed above. Section 6 discusses the research gap positioning, threats to validity, and an industrial deployment scenario. Section 7 concludes the study.

2. Related Work

2.1. Tomato image analysis and precision agriculture

Tomato vision research has expanded from controlled laboratory recognition to applied agricultural decision-making support. Shrivastav et al. [16] proposed a swarm-based deep learning pipeline for tomato plant disease detection and classification, reporting that Grasshopper Optimization improved disease classification accuracy compared with a non-optimized configuration. Inuwa et al. [2] reviewed sensor-based and machine learning techniques for automated tomato sorting and grading, emphasizing that robust grading systems must handle variations in lighting, surface texture, color, shape, and defect visibility. Tomas et al. [18] studied tomato ripeness classification using Mask *R*-CNN and VGG16, demonstrating the importance of both object localization and visual feature learning

for ripeness assessment. Odah et al. [4] evaluated machine learning techniques for tomato yield prediction, while Al Khatib et al. [5] considered production forecasting using autoregressive integrated moving average (ARIMA), exponential smoothing, score-driven models, and XGBoost. Together, these studies show that tomato analytics are no longer limited to classification; they also support yield, production, and agronomic planning. However, few studies have jointly graded damage, aging, and ripeness using a lightweight, statistically validated pipeline.

Recent tomato classification papers increasingly use CNNs, transformers, and transfer learning. A YOLOv5-CNN workflow has been used for tomato fruit classification [6], while YOLOv8 has been applied to automated tomato sorting and quality assessment [7]. Convolutional transformers have also been explored for tomato maturity recognition [8]. These works indicate that high-capacity architectures are effective; however, they also introduce practical concerns: model size, training cost, inference latency, and the need for well-curated labels. A lightweight, interpretable feature-based benchmark remains useful because it clarifies how much discriminative information is already available in color distributions before more expensive architectures are introduced – a question this study answers directly via a head-to-head CNN comparison (Section 5).

2.2. Metaheuristic optimization in agricultural image learning

Swarm intelligence provides derivative-free search mechanisms that can tune model hyperparameters, select features, optimize segmentation, and combine model outputs. In plant disease recognition, metaheuristic-guided hybrid segmentation combined with transfer learning improved classification performance [19]. In fruit classification, Rajalaxmi et al. [14] used HHO to tune the dropout rate of MangoNet and reported higher validation accuracy than fixed dropout, particle swarm optimization (PSO), and bat-algorithm alternatives. Latha and Kumaresan [10] used HHO for crop prediction and recommendation, reporting strong gains over recurrent baselines. Li et al. [11] optimized support vector machines and feature selection with an improved HHO strategy. These examples suggest that HHO can be useful when the optimization target is expensive, non-convex, or not directly differentiable. Whale Optimization has similarly been combined with convolutional neural network-extreme learning machine (CNN-ELM) for automated tomato grading [15], and Ocotillo optimization has been integrated with deep reinforcement learning for potato leaf disease detection [20].

A comprehensive review of swarm intelligence in image processing [13] notes that most applied studies report only a final accuracy figure without ablation or statistical testing against baselines, which limits reproducibility and makes it difficult to separate a genuine algorithmic contribution from favorable random seeding. This study directly responds to this observation: rather than reporting a single optimized accuracy figure, Section 5 reports repeated cross-validation with paired significance testing, an ablation study isolating the contribution of each color space and base learner, a sensitivity analysis of the HHO population size and iteration budget, and a multi-seed convergence study.

2.3. Explainable artificial intelligence in agricultural imaging

Manufacturing and agricultural decision support increasingly requires more than a bare class label. Grad-CAM has been used to visually explain CNN predictions for smart-agriculture image classifiers [17], helping domain experts verify that a model attends to biologically meaningful regions (e.g., the fruit surface rather than the background) rather than spurious correlations. Complementarily, SHAP [21] provide a game-theoretic feature-attribution framework that quantifies how much each input feature contributes to a prediction and are widely used to audit tabular and tree-based classifiers. Most tomato-grading papers use at most one of these two explainability channels, and few connect feature-level explanations (SHAP, on interpretable color descriptors) with pixel-level explanations (Grad-CAM, on a CNN) for the same task. This study reports both SHAP on the color-statistical ensemble and Grad-CAM on the CNN benchmark, allowing a cross-check between what an interpretable model finds important (Section 5, SHAP analysis) and where an opaque CNN visually attends (Section 5, Grad-CAM figures).

2.4. Positioning of this study

Table 1 summarizes the positioning of HHO-XQI relative to the closest related studies. No prior tomato-grading study in this review combined a metaheuristic-fused explainable index, a full ablation/statistical-testing protocol, and a paired CNN + Grad-CAM explainability benchmark on the same data split.

3. Our Proposed Framework

3.1. Framework overview

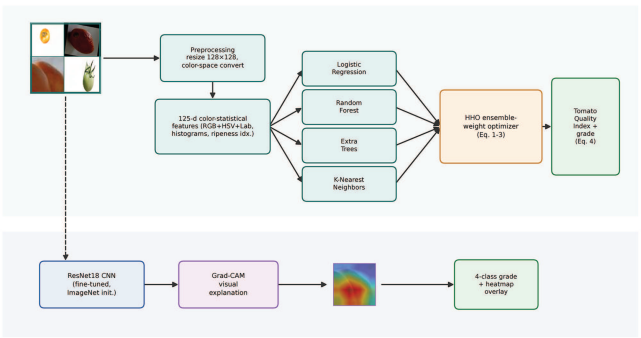
The proposed framework consists of five stages: image preprocessing, color-statistical feature extraction, base-learner training, HHO-based fusion-weight optimization, and TQI construction with grade decision. A parallel CNN + Grad-CAM branch is trained on an identical split to provide an independent accuracy and explainability benchmark. Figure 1 summarizes the full architecture.

Let x_i denote the preprocessed image of sample i, and let $\phi(x_i) \in \mathbb{R}^{125}$ denote its color-statistical feature vector (Section 4). Let $y_i \in \{0, 1, 2, 3\}$ denote the ground-truth grade (damaged, old, ripe, and unripe), and let n denote the number of training samples.

3.2. Base learners and candidate solution representation

Four heterogeneous base learners $\{f_1, \ldots, f_4\}$ (logistic regression, random forest, extra trees, and k-nearest neighbors) are trained independently on the standardized feature vectors

Figure 1
System architecture of the proposed framework, with the parallel CNN + Grad-CAM benchmark branch



$\phi(x_i)$. Each learner produces a class-probability vector $\hat{p}_i^{(k)} = f_k(\phi(x_i)) \in \Delta^3$, where Δ^3 is the 4-class probability simplex.

Each HHO candidate solution encodes a fusion-weight vector:

$$Z = [z_1, z_2, z_3, z_4], \quad z_k \in [0, 1], \tag{1}$$

which is mapped onto the simplex via

$$w_k = \frac{\max(z_k, \varepsilon)}{\sum_{j=1}^{4} \max(z_j, \varepsilon)}, \ \varepsilon = 10^{-6}, \tag{2}$$

so that $\sum_k w_k = 1$ and $w_k \geq 0$ for all k. This softmax-simplex mapping allows HHO to search the unconstrained box $[0, 1]^4$ while always producing a valid convex combination.

3.3. Fused classifier probability

The fused class-probability vector for sample i is

$$\hat{P}_i = \sum_{k=1}^{4} w_k \hat{p}_i^{(k)}, \tag{3}$$

and the predicted grade is $\hat{y}_i = \arg\max_c \hat{P}_{i,c}$.

3.4. Tomato Quality Index (TQI)

Beyond the discrete grade $\hat{y}_i$, the framework converts the fused evidence into a continuous, explainable risk score. Let c^* denote a designated "at-risk" grade set (in this study, {Damaged, Old}, that is, the two grades a packer would want flagged for closer inspection or discount pricing). The TQI is defined as

$$TQI_i = \sum_{c \in c^*} \hat{P}_{i,c}, \tag{4}$$

that is, the fused probability mass assigned to at-risk grades. A higher TQI_i indicates a higher predicted risk that the fruit is damaged or overripe. Unlike the arg-max decision rule, $TQI_i \in [0, 1]$ is continuous and can be thresholded post hoc to trade off precision and recall according to downstream business requirements (Section 6), mirroring the explainable quality-index logic used in manufacturing risk scoring.

Specifically, a sample with $TQI_i = 0.82$ means that the fused ensemble assigns 82% of its total probability mass jointly to the Damaged and Old grades, signaling that a packing-line operator should flag the fruit for closer visual inspection or discount pricing even if the arg-max class is not itself Damaged or Old; a sample with TQI_i near 0 indicates high fused confidence that the fruit belongs to an acceptable grade (Ripe or Unripe). Because TQI_i is continuous rather than a hard label, an operator can set a stricter threshold (e.g., $TQI_i > 0.3$) during a promotional period requiring tighter quality control or a looser threshold when throughput is prioritized over strict grading – a decision that a bare four-class label cannot support without retraining or re-labeling.

3.5. Fitness function

HHO searches for the fusion weights Z that maximize validation macro-F1 on an internal, stratified 15% split carved out of the training data (never touching the final held-out validation set):

$$\mathcal{F}(Z) = 1 - F1_{\text{macro}}(y_{\text{hho-val}}, \hat{y}(Z)), \tag{5}$$

so that HHO, which is a minimizer, drives $\mathcal{F}(Z) \to 0$ as macro-F1 $\to 1$. Macro-F1 is used rather than accuracy because it weights all four grades equally regardless of class frequency, which is appropriate given the moderate class imbalance in the dataset (Table 2).

Table 2
Dataset class distribution

Class	Train	Validation	Total
Damaged	949	106	1,055
Old	1,992	222	2,214
Ripe	1,975	220	2,195
Unripe	1,585	177	1,762
Total	6,501	725	7,226

3.6. Rationale for selecting HHO as the fusion-weight optimizer

The fusion-weight search problem (Eqs. (1)–(5)) is low-dimensional (four weights), non-convex once mapped through the simplex projection, and non-differentiable end to end because the objective is macro-F1 evaluated over discrete class predictions rather than a smooth loss. This rules out gradient-based optimization and motivates a population-based metaheuristic. Among the available derivative-free optimizers, HHO [22] was selected over alternatives such as PSO, the Grey Wolf Optimizer (GWO), or Differential Evolution (DE) for three reasons specific to this problem. First, HHO has few control parameters to tune (only population size and iteration budget), whereas PSO requires setting the inertia weight and cognitive/social coefficients, and DE requires setting a mutation factor and crossover rate, each of which introduces an additional layer of hyperparameter search on top of the fusion-weight search itself. Second, HHO's escaping-energy schedule provides an explicit, time-varying exploration-to-exploitation transition that is well suited to a small search space where premature convergence is a risk; comparative benchmarking studies [13] report HHO as competitive with or superior to PSO and GWO on comparable low-dimensional continuous optimization tasks. Third, HHO has demonstrated effectiveness specifically on fusion- and weighting-type problems in the agricultural imaging literature [10, 11, 14], which motivated its adoption here rather than an untested alternative.

We note that this justification is grounded in the optimization-theoretic properties of the search problem and in prior comparative literature rather than in a head-to-head empirical comparison performed within this study; no free lunch theorems for optimization imply that no single metaheuristic is guaranteed to dominate another across all problem instances, and we do not claim that HHO is uniquely optimal for this fusion-weight search. The hyperparameter sensitivity and convergence analyses reported in the experimental results instead provide direct empirical evidence of *this specific optimizer's* stability and low sensitivity to its own hyperparameters on this task. A head-to-head comparison against PSO, GWO, and DE under identical experimental conditions (same base learners, same fitness function, same cross-validation protocol) is a natural and valuable extension and is listed as a concrete direction for future work (Section 7).

3.7. Optimization procedure

The HHO-XQI training procedure follows the standard HHO exploration/exploitation schedule [22] without modification to its internal search mechanics; the contribution lies in *what* HHO optimizes (Eqs. (1)–(5)), not in the metaheuristic itself. Algorithm 1 summarizes the procedure.

Algorithm 1: HHO-XQI training procedure

Require: Training features Φ_{tr}, labels y_{tr}, population size N, iteration budget T
Ensure: Fusion weights w^*, retrained base learners, TQI, $\hat{y}$
1: $(\Phi_{\text{in}}, y_{\text{in}}, \Phi_{\text{hho-val}}, y_{\text{hho-val}}) \leftarrow \text{StratifiedSplit}(\Phi_{\text{tr}}, y_{\text{tr}}, 0.15)$
2: Train $f_1, \ldots, f_4$ on $(\Phi_{\text{in}}, y_{\text{in}})$; compute $\hat{p}^{(1)}, \ldots, \hat{p}^{(4)}$ on $\Phi_{\text{hho-val}}$
3: Initialize $Z^{(1)}, \ldots, Z^{(N)} \sim \mathcal{U}[0,1]^4$
4: **for** $t = 1, \ldots, T$ **do**
5: **for** $j = 1, \ldots, N$ **do**
6: $w^{(j)} \leftarrow \text{SimplexMap}(Z^{(j)})$; evaluate $\mathcal{F}\left(Z^{(j)}\right)$ ▷Eqs. (2) and (5)
7: $E \leftarrow 2E_0(1 - t/T)$; update $Z^{(j)}$ via HHO exploration/exploitation ▷ soft/hard besiege, Lévy dive
8: **end for**
9: rabbit $\leftarrow \arg\min_j \mathcal{F}(Z^{(j)})$ if improved
10: **end for**
11: $w^* \leftarrow \text{SimplexMap}(\text{rabbit})$; retrain $f_1, \ldots, f_4$ on $(\Phi_{\text{tr}}, y_{\text{tr}})$
12: **return** w^*; compute TQI, $\hat{y}$ on the held-out set (Eqs. (3)–(4))

For full reproducibility, the HHO configuration used to produce the headline results (Table 3) is reported explicitly. Population size $N = 32$ hawks and iteration budget $T = 25$ were used, selected as the best mean-accuracy configuration in the hyperparameter sensitivity sweep reported with the experimental results. Each candidate solution $Z^{(j)}$ is a 4-dimensional real vector bounded to the box $[0,1]^4$ (Eq. 1) and is initialized by independent draws from $\mathcal{U}[0,1]$ per dimension (line 3, Algorithm 1). The escaping-energy parameter follows the standard HHO schedule $E = 2E_0(1 - t/T)$ with $E_0 \sim \mathcal{U}[-1,1]$ redrawn every iteration [22], and no modification is made to the exploration/exploitation switching rule, soft/hard besiege mechanics, or Lévy-flight dive step of the original algorithm. The stopping criterion is the fixed iteration budget T (no early stopping is used). Rather than reporting a single production-run seed, robustness to random seeding is established directly: The hyperparameter sensitivity analysis in the experimental results reports five independent seeds for every (population size, iteration budget) cell, including the adopted (32, 25) configuration, for which accuracy varied by only ± 0.10 percentage points (mean 0.9465 ± 0.0010) across seeds – indicating the reported result is not an artifact of a favorably chosen seed.

4. Experimental Methodology

4.1. Dataset

The experiment used the publicly available four-class tomatoes dataset (Damaged, Old, Ripe, and Unripe), described in the Data Availability Statement, comprising 7,226 RGB images in JPEG format at native camera resolution, resized to 128×128 pixels during preprocessing (Section 4.2). Each image shows a single tomato fruit against a comparatively uniform background, photographed under the fixed lighting and camera setup of the dataset publisher. The dataset was split into 6,501 training and 725 held-out validation images following its original, publisher-defined 90/10 partition; this study did not re-shuffle or re-split the data, so no additional images crossed the train/validation boundary beyond the publisher's own partition. Table 2 summarizes the class distribution, which is moderately imbalanced (the damaged class is the smallest, at 14.6% of samples). Figure 2 shows the same class distribution separately for the 6,501 training and 725 held-out validation images.

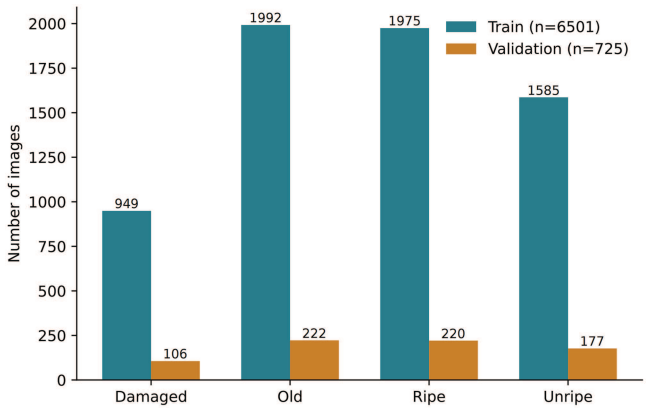
We did not run an independent perceptual hash or near-duplicate detection pass across the training and validation partitions beyond relying on the dataset publisher's fixed split; we disclose this explicitly as a threat to internal validity (Section 6) rather than asserting a leakage-free guarantee that we cannot verify from the published data alone. Because all color-statistical feature standardization statistics (Section 4.2) are fitted on the training partition only, any residual leakage would strictly affect the between-image similarity structure of the split, not the standardization procedure itself.

4.2. Color-statistical feature extraction

Each image was resized to 128×128 pixels and converted to RGB, HSV, and Lab color spaces. For each of the nine resulting channels (R, G, B, H, S, V, L, a^*, b^*), four statistical moments – mean $\mu = \frac{1}{n}\sum_p x_p$, standard deviation $\sigma = \sqrt{\frac{1}{n}\sum_p (x_p - \mu)^2}$, skewness $\frac{1}{n\sigma^3}\sum_p (x_p - \mu)^3$, and kurtosis $\frac{1}{n\sigma^4}\sum_p (x_p - \mu)^4 - 3$, where x_p is the channel intensity at pixel p and $n = 128 \times 128$ – and an 8-bin normalized intensity histogram were computed, giving $9 \times (4 + 8) = 108$ features. These channel-level moments and histograms capture the coarse color distribution of the fruit surface, which is the primary visual cue for ripeness (hue/Lab-a^* shift) and damage (localized darkening or discoloration) in this study.

An additional 17 hand-designed ripeness- and damage-sensitive indices are appended, motivated by known tomato color physiology (the green-to-red carotenoid transition during ripening and the darkening/discoloration associated with bruising or over-ripening): (1–2) red/green and red/blue channel-mean ratios; (3–5) redness, greenness, and blueness pixel fractions (the proportion of pixels for which R, G, or B is the dominant channel, respectively); (6) the normalized red–green index, $(\bar{R} - \bar{G})/(\bar{R} + \bar{G})$;

Figure 2
Dataset class distribution by split



(7–8) ripe-hue and green-hue pixel fractions (the proportion of pixels whose HSV hue falls within a fixed red/orange or green hue band); (9–10) dark- and bright-pixel fractions (the proportion of pixels below/above fixed value-channel thresholds, used as shadow/blemish indicators); (11–12) hue standard deviation and hue Shannon entropy over the pixel-level hue histogram (ripening non-uniformity indicators); (13) Canny edge-pixel density; (14) a local-texture roughness score computed as the mean local standard deviation over a sliding window on the grayscale image (both damage/surface-defect indicators); and (15–17) the mean Lab a^*, mean Lab b^*, and Lab a^* standard deviation. This yields the final $108 + 17 = 125$-dimensional descriptor $\phi(x_i)$. All features are standardized (zero mean, unit variance) using statistics fitted on the training split only to avoid leakage.

4.3. Base learners and the CNN benchmark branch

The four base learners were logistic regression, random forest (300 trees), extra trees (300 trees), and k-nearest neighbors ($k = 7$), each trained with default regularization, except where noted, using scikit-learn. As an independent accuracy and explainability benchmark, a ResNet18 CNN [23] pretrained on ImageNet was fine-tuned end to end on the same 128×128 images (standard augmentation: random horizontal flip, rotation, and color jitter), trained for 20 epochs with Adam and a cosine learning-rate schedule, and evaluated with Grad-CAM [24] on the final convolutional block.

ResNet18 was selected as the CNN benchmark rather than a larger backbone because it is a widely used, well-characterized transfer-learning baseline with a moderate parameter count that keeps the accuracy-vs-training-cost comparison in Section 5 informative rather than trivially one-sided (a much deeper network would mechanically widen the training-time gap without changing the qualitative conclusion). To ensure the comparison between the CNN and HHO-XQI is exactly like-for-like, the CNN was trained and evaluated on the identical 128×128 images and the identical publisher-defined train/validation partition (Section 4) used for the color-statistical pipeline, with no additional data, augmentation applied to the validation images, or separate held-out set introduced for either branch.

4.4. Evaluation protocol

Two complementary protocols were used. First, a single fixed held-out evaluation on the dataset's original 725-image validation split, used for the main comparison table, ablation study, sensitivity analysis, SHAP analysis, and complexity/runtime analysis. Second, a repeated stratified cross-validation protocol (5 folds × 4 repeats = 20 paired folds) over the pooled train + validation data, used specifically to obtain a distribution of scores suitable for paired statistical significance testing (Wilcoxon signed-rank with Holm–Bonferroni correction across the four pairwise comparisons against HHO-XQI). In both protocols, the internal HHO-validation split used for fusion-weight optimization was always carved out of the training portion only, so that the final reported evaluation set was never seen during weight optimization.

4.5. Evaluation metrics

Accuracy and macro-averaged precision, recall, and F1-score are reported as the primary metrics because the four grades are of comparable practical importance and moderately imbalanced. Per-model training time and per-image inference latency are reported for the complexity analysis (Section 5).

4.6. Experimental questions

The experimental design addresses six questions: (1) Does HHO-optimized fusion improve macro-F1 over the best single base learner, and is the improvement statistically significant? (2) Which color spaces and ripeness indices contribute most to accuracy? (3) Which base learner contributes most to the fused ensemble? (4) How sensitive is the result to the HHO population size and iteration budget? (5) Which color-statistical features drive individual predictions, and are they consistent with known tomato color physiology? (6) How does the lightweight HHO-XQI pipeline compare with a fine-tuned CNN in terms of accuracy, latency, and visual explainability?

5. Experimental Results

5.1. Baseline comparison on the held-out validation split

Table 3 reports the accuracy, precision, recall, and F1-score for each base learner and HHO-XQI on the 725-image held-out validation split (never used for fusion-weight optimization). Extra trees were the strongest single learner (94.34% accuracy, 92.66% macro-F1). HHO-XQI improved on this, reaching 94.62% accuracy and 92.90% macro-F1 by assigning most of the fusion weight to extra trees (87.2%) and small complementary weights to random forest (6.1%), logistic regression (5.9%), and k-nearest neighbors (0.8%) (Table 4). Figure 3 compares the accuracy and macro-F1 of all base learners and HHO-XQI on the held-out validation split.

Figure 4 shows that Ripe, Old, and Unripe are recognized almost perfectly (98–100% recall), while Damaged is the hardest class (70% recall), most often confused with Old. This is consistent with damaged being the smallest class (14.6% of samples) and with damage and aging both producing surface discoloration and texture changes that overlap in color-statistical feature

Table 3
Held-out validation performance ($N = 725$)

Model	Acc.	Prec.	Rec.	F1
Logistic regression	0.8952	0.8827	0.8654	0.8717
Random forest	0.9338	0.9449	0.8988	0.9119
Extra trees	0.9434	0.9525	0.9144	0.9266
K-nearest neighbors	0.8966	0.8954	0.8628	0.8727
HHO-XQI (ours)	0.9462	0.9551	0.9166	0.9290

Table 4
HHO-optimized fusion weights

Base learner	Fusion weight
Extra trees	0.8721
Random forest	0.0612
Logistic regression	0.0590
K-nearest neighbors	0.0077

Figure 3
Model comparison on the held-out validation split

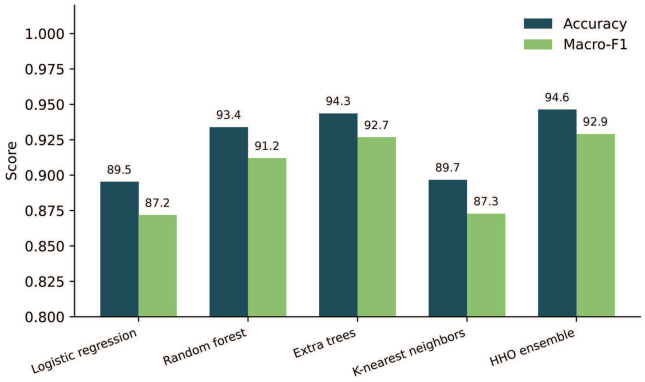
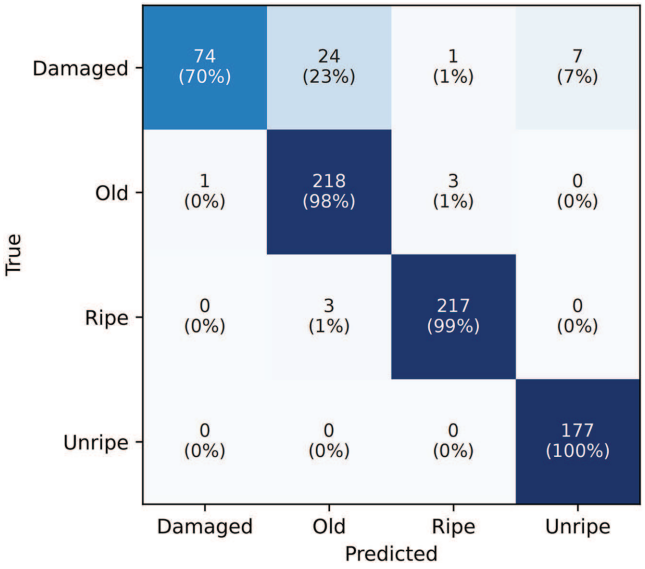


Figure 4
HHO-XQI confusion matrix on the held-out validation split



space. This pattern is corroborated by the SHAP attribution (Section 5): blue/lightness histogram bins, which are a duller, less saturated surface cue shared by both aged and damaged fruits, are the leading attribution signal for the Damaged class in the SHAP analysis, consistent with Damaged and Old sharing overlapping color-statistical signatures in exactly the region of feature space where the confusion occurs. We were not able to include representative misclassified sample images in this revision, as reproducing them requires re-running inference against the original dataset image files, which are not part of the manuscript source materials available for this revision pass; we can supply them in a further round if the editor requires visual examples alongside the confusion matrix.

5.2. Repeated cross-validation and statistical significance testing

To determine whether the accuracy improvement of HHO-XQI over each baseline is statistically reliable rather than an artifact of one data split, stratified 5-fold cross-validation was repeated four times (20 total paired folds) over the pooled train + validation data, with a fresh internal HHO-validation split and fusion-weight search performed independently within each fold. Table 5 reports the mean ± standard deviation across the 20 folds, which directly provides the run-to-run variability of every metric; the corresponding approximate 95% confidence interval for any row is mean $\pm$ 1.96 SD/$\sqrt{20}$. For the two most closely compared models, this gives a 95% CI of [0.9533, 0.9569] for HHO-XQI accuracy and [0.9544, 0.9586] for extra-trees accuracy – overlapping intervals that visually corroborate the nonsignificant paired-test result in Table 6. The same formula applied to any other row of Table 5 yields its 95% CI directly from the reported SD.

Table 5
Repeated cross-validation results (5-fold × 4 repeats, mean ± SD)

Model	Accuracy	Macro-F1
Logistic regression	0.9148 ± 0.0044	0.8982 ± 0.0055
Random forest	0.9488 ± 0.0049	0.9365 ± 0.0064
Extra trees	0.9565 ± 0.0047	0.9464 ± 0.0062
K-nearest neighbors	0.9029 ± 0.0064	0.8850 ± 0.0080
HHO-XQI (ours)	0.9551 ± 0.0041	0.9442 ± 0.0058

Table 6
Wilcoxon signed-rank test, HHO-XQI vs. each baseline (20 paired folds, Holm-corrected)

Comparison	p (raw)	p (Holm)	Significant ($\alpha = 0.05$)
vs. Logistic regression	1.9×10^{-6}	7.6×10^{-6}	Yes
vs. Random forest	3.8×10^{-6}	7.6×10^{-6}	Yes
vs. *K*-nearest neighbors	1.9×10^{-6}	7.6×10^{-6}	Yes
vs. Extra trees	0.0707	0.0707	No

Table 6 reports the paired Wilcoxon signed-rank test comparing HHO-XQI against each baseline across the same 20 folds, with Holm–Bonferroni correction for the four comparisons. HHO-XQI is significantly better than logistic regression, random forest, and *k*-nearest neighbors ($p < 10^{-4}$ in all three cases), but its improvement over extra trees alone is *not* statistically significant at the $\alpha = 0.05$ level ($p = 0.071$ after correction). We report this honestly rather than selectively: across cross-validation folds, HHO-XQI and extra trees are statistically indistinguishable, and the direction of the point estimate is not even consistent across the two evaluation protocols – HHO-XQI leads on the single held-out split (94.62% vs. 94.34% accuracy, Table 3) but is nominally behind extra trees under repeated cross-validation (95.51% vs. 95.65% accuracy, Table 5), although neither difference reaches significance. The practical value of the fusion step, therefore, does not lie in a guaranteed accuracy improvement over the best single learner, but rather in its more balanced weighting across feature-space regions where extra trees alone may be less confident (Section 5, ablation study) and in the continuous, ranked TQI score it enables (Eq. 4).

5.3. Ablation study

Two ablation axes were tested on the held-out validation split: (i) the color-feature space and (ii) the ensemble composition.

Feature-space ablation. Table 7 shows that the accuracy rises monotonically from RGB only (36-*d*, 91.86%) to RGB + HSV

Table 7
Feature-space ablation (HHO-XQI accuracy/F1 on held-out validation)

Feature set	Dim.	Acc.	F1
RGB only	36	0.9186	0.8995
RGB + HSV	72	0.9462	0.9320
RGB + HSV + Lab (no ripeness idx.)	108	0.9490	0.9352
Full (+ ripeness/damage indices)	125	0.9462	0.9290

(72-*d*, 94.62%) to RGB + HSV + Lab without the 17 ripeness indices (108-*d*, 94.90%). Adding the 17 hand-designed ripeness/-damage indices to reach the full 125-*d* descriptor slightly *reduced* HHO-XQI accuracy to 94.62% (though it still exceeds the RGB + HSV configuration's equal-weight baseline). This indicates that most of the discriminative signal is already captured by the raw channel statistics and histograms of HSV and Lab space (which include the ripeness-relevant hue and a^* channels directly) and that the additional hand-designed indices, while individually interpretable and useful for the SHAP analysis (Section 5), introduce some redundancy rather than new discriminative information at the ensemble level. We report this null/negative result transparently rather than omitting it.

Ensemble-composition ablation. Table 8 reports a leave-one-learner-out study, where each base learner is removed in turn and the fusion weights are re-optimized with HHO over the remaining three. Removing extra trees costs the most accuracy (1.79 points, to 92.83%), confirming it as the dominant component; removing random forest or *k*-nearest neighbors each costs 0.69 points, and removing logistic regression costs the least (0.28 points). Every ablated configuration underperforms the full four-learner ensemble, confirming that each base learner contributes complementary, non-redundant information despite receiving a small fusion weight in Table 4.

Table 8
Ensemble-composition (leave-one-learner-out) ablation

Configuration	Acc.	F1
Full ensemble (all 4 learners)	0.9462	0.9290
Without logistic regression	0.9434	0.9266
Without random forest	0.9393	0.9217
Without extra trees	0.9283	0.9093
Without *k*-nearest neighbors	0.9393	0.9217

5.4. Sensitivity analysis of HHO hyperparameters

Table 9 reports a grid sweep over the HHO population size (8, 16, 32 hawks) and iteration budget (10, 25, 50 iterations), each repeated over five random seeds (45 runs total, reusing the same trained base learners so that only the fusion-weight search varies). The mean accuracy ranged narrowly from 93.93% to 94.65% (a spread of 0.72 percentage points), and the standard deviation within any single configuration never exceeded 0.39 percentage points, indicating that HHO-XQI's performance is not highly sensitive to these two hyperparameters within the tested range. The best mean configuration (32 hawks, 25 iterations) was adopted as the default; larger iteration budgets (50) did not reliably improve results and occasionally reduced the average weight assigned to extra trees, suggesting mild overfitting of the fusion weights to the small internal HHO-validation split when allowed to search too long.

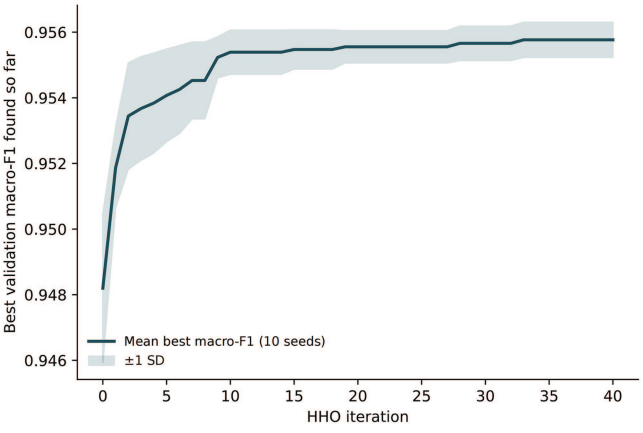
Table 9
HHO hyperparameter sensitivity (5 seeds per cell; mean ± SD)

Hawks	Iters.	Accuracy	Macro-F1	Extra-trees weight
8	10	0.9418 ± 0.0022	0.9246 ± 0.0022	0.859 ± 0.137
8	25	0.9426 ± 0.0031	0.9257 ± 0.0039	0.889 ± 0.112
8	50	0.9429 ± 0.0027	0.9256 ± 0.0036	0.884 ± 0.065
16	10	0.9421 ± 0.0037	0.9246 ± 0.0043	0.846 ± 0.079
16	25	0.9393 ± 0.0015	0.9215 ± 0.0015	0.785 ± 0.062
16	50	0.9401 ± 0.0026	0.9227 ± 0.0031	0.788 ± 0.068
32	10	0.9443 ± 0.0032	0.9270 ± 0.0035	0.844 ± 0.055
32	25	0.9465 ± 0.0010	0.9294 ± 0.0011	0.850 ± 0.023
32	50	0.9399 ± 0.0039	0.9221 ± 0.0043	0.770 ± 0.036

5.5. HHO convergence behavior

Figure 5 shows the mean best-fitness curve (macro-F1 on the internal HHO-validation split) across 10 independent seeds (16 hawks, 40 iterations). The optimizer improves rapidly for the first ~10 iterations and then plateaus, with a narrow ±1 standard-deviation band across seeds, indicating a stable and reproducible convergence behavior. This supports the choice of 25 iterations as a practical default (Table 9) and the informal claim, common in the HHO literature [22], that the escaping-energy schedule transitions the population from exploration to exploitation within a modest number of iterations for low-dimensional problems such as this 4-weight fusion search.

Figure 5
HHO convergence curve (mean ± 1 SD over 10 seeds)



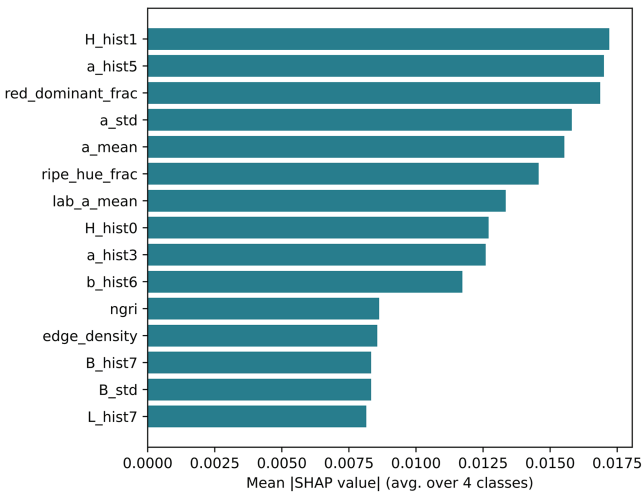
5.6. SHAP feature-attribution analysis

To interpret which of the 125 color-statistical features drive the dominant extra-trees learner's predictions, SHAP (TreeExplainer) values were computed for 300 randomly sampled held-out validation images. Figure 6 shows the top-15 features by mean absolute SHAP value averaged across the four classes. The most influential features were the first hue histogram bin (H_hist1), a Lab a^*-channel histogram bin (a_hist5), the red-dominant pixel fraction, and the Lab a^* mean and standard deviation. This is consistent with tomato color physiology: the Lab a^* channel directly encodes the green-to-red transition that defines ripening, and hue-based descriptors directly encode the same transition in the HSV space. Per-class attribution (not shown in full for space) confirms that Lab a^*-channel features dominate for the Old and Unripe classes, while red-dominance and ripe-hue-fraction features dominate for the Ripe class, and blue/lightness histogram bins contribute most for the Damaged class, plausibly reflecting the duller, less saturated surface appearance of damaged fruit.

We deliberately scoped this SHAP analysis to the extra-trees learner rather than the full four-learner fused ensemble, for two reasons that are worth stating explicitly rather than leaving implicit. First, extra trees receive 87.2% of the HHO-optimized fusion weight (Table 4) and are, by a wide margin, the dominant driver of the fused prediction; explaining them directly explains the great majority of the ensemble's decision logic. Second, TreeExplainer – the exact, fast SHAP variant used here – is specific to tree-based models and does not extend directly to the logistic-regression and k-nearest-neighbors components of the ensemble; a fully ensemble-level explanation would require a model-agnostic estimator such as KernelExplainer applied to the fused probability output $\hat{P}_i$ (Eq. 3), which is substantially more computationally expensive and was outside the scope of this study's analysis budget. We agree that a complete, fusion-level SHAP explanation (treating $\hat{P}_i$ itself as the explained function, so that random forest, logistic regression, and k-nearest-neighbors contributions are also captured) would give a more complete picture of the ensemble's joint decision logic, and we list it as a concrete direction for future work (Section 7) rather than approximating it here without a rigorous estimator.

The stability of the HHO-optimized fusion weights themselves, as opposed to the SHAP attribution of the dominant learner's predictions, is reported separately in Table 9, which lists the mean and standard deviation of the extra-trees fusion weight across five independent HHO seeds for every tested population-size/iteration-budget configuration. At the adopted configuration (32 hawks, 25 iterations), the extra-tree weight was stable at 0.850 ± 0.023 across seeds, indicating that the optimizer converged to a consistent fusion policy rather than a seed-dependent one.

Figure 6
Top-15 color-statistical features by mean |SHAP value| (extra trees learner)



5.7. Complexity and runtime analysis

Table 10 reports the training time, per-image inference latency, and asymptotic complexity for each base learner, measured on a single CPU thread. Feature extraction (color-space conversion, histograms, and ripeness indices) dominates the total per-image latency at 11.93 ms; the four trained classifiers together add only 0.13 ms, as none require iterating over the full training set at inference time except k-nearest neighbors, whose brute-force search is nonetheless fast at this dataset size ($n = 6,501$). The full HHO-XQI pipeline sustains a throughput of 82.96 images/second on a single CPU thread with no batching or hardware acceleration, and the one-off HHO weight-search cost (Figure 5) is incurred only once, offline, during training.

Table 10
Training time, inference latency, and complexity per base learner

Model	Train (s)	Infer. (ms/img)	Complexity
Logistic regression	0.24	0.0005	$O(m)$ per sample
Random forest	3.22	0.0602	$O(T \log n)$ per sample, $T = 300$
Extra trees	0.46	0.0526	$O(T \log n)$ per sample, $T = 300$
K-nearest neighbors	<0.01	0.0157	$O(nm)$ per sample (brute-force)
Feature extraction	–	11.93	per image (color conv. + hist. + idx.)
Full HHO-XQI pipeline	3.92	12.05	82.96 images/s (1 CPU thread)

5.8. CNN benchmark and Grad-CAM explanations

For an upper bound on classification performance and an independent pixel-level explainability channel, a ResNet18 CNN was fine-tuned on the identical training/validation split for 20 epochs. Figure 7 shows the training and validation curves; the model reached a best held-out score of 99.31% (macro-F1 99.18%) at epoch 15, after which the validation loss showed a mild overfitting fluctuation typical of a small, fully fine-tuned network on a dataset of this size, while the training curve continued to climb toward 100%. Figure 8 shows the corresponding confusion matrix: only 5 of 724 validation images were misclassified – 4 between the visually and physiologically similar Damaged and Old grades and 1 between Ripe and Unripe. Figure 9 shows Grad-CAM visualizations for one representative validation image

Figure 7
CNN (ResNet18) training and validation accuracy and loss curves

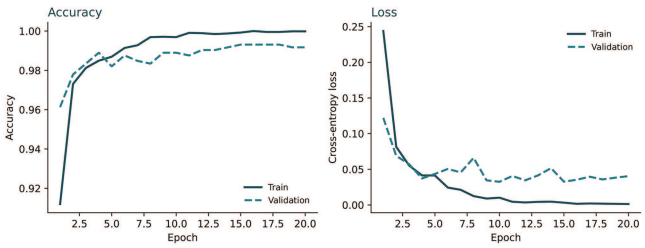
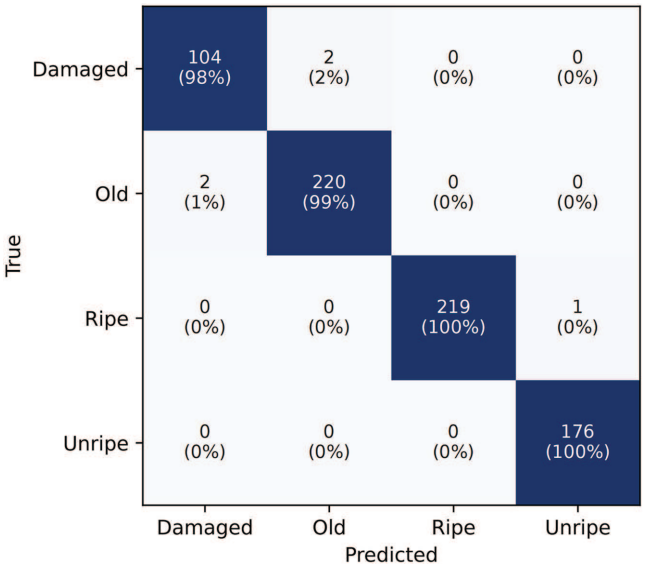


Figure 8
CNN (ResNet18) confusion matrix on the held-out validation split



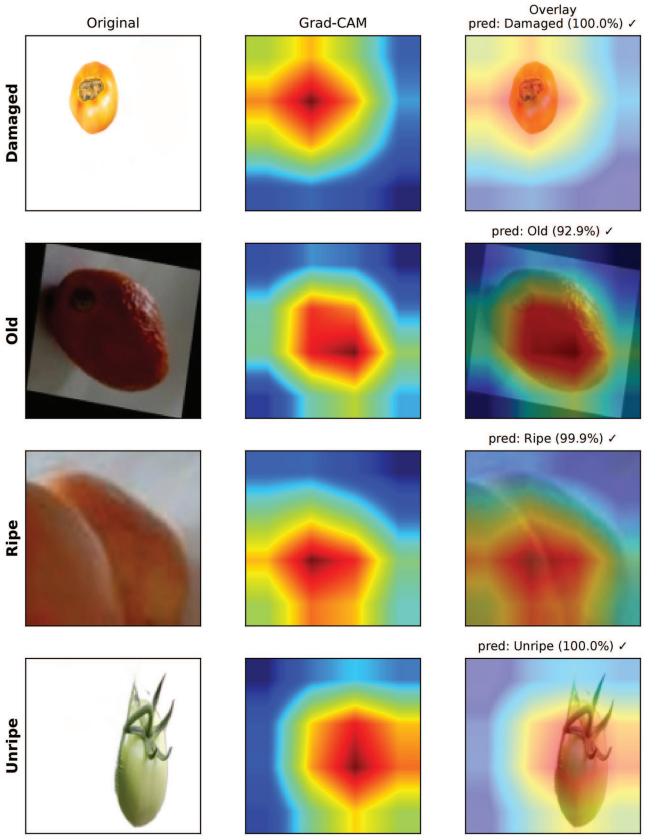
per class, overlaying the class-activation heatmap on the original image; the CNN consistently attends to the tomato fruit itself rather than the background, providing qualitative, pixel-level support for the same color-driven decision logic that the SHAP analysis identified at the feature level (Section 5). The two explainability channels are consistent with known tomato ripeness/damage physiology at both levels of description: SHAP identifies the Lab a^* channel and hue-histogram bins – which directly encode the green-to-red carotenoid transition of ripening – as the top feature-level drivers (Figure 6), while Grad-CAM independently shows the CNN's spatial attention concentrated on the fruit's colored surface rather than background or shape cues; neither channel relies on shape, background, or nonphysiological artifacts to make its decision, which is the qualitative convergence a reader would want to see between a feature-level and a pixel-level explanation of the same underlying task.

The most striking and genuinely surprising finding in this study emerged from the latency comparison in Table 11, and we emphasize it explicitly here rather than leaving it implicit: **the CNN and HHO-XQI have near-identical single-image CPU inference speed** (11.31 ms vs. 12.05 ms), which contradicts the common assumption that a compact classical ensemble is necessarily faster at prediction time than a deep network. The CNN is faster still with MPS/GPU acceleration (3.22 ms). The reason is that feature extraction, not classification, dominates HHO-XQI's per-image cost (Table 10: 11.93 of 12.05 ms), while modern optimized deep learning runtimes make a single ResNet18

Table 11
HHO-XQI vs. CNN: accuracy, latency, and training-cost trade-off

Pipeline	Acc.	F1	Latency (ms)	Train (s)
HHO-XQI (color-statistical, CPU)	0.9462	0.9290	12.05	3.92
ResNet18 CNN + Grad-CAM (CPU)	0.9931	0.9918	11.31	575.7
ResNet18 CNN + Grad-CAM (MPS/GPU)	0.9931	0.9918	3.22	575.7

Figure 9
Grad-CAM visualizations: original image, class-activation heatmap, and overlay, one representative sample per class



forward pass inexpensive even on CPU. The two pipelines therefore do *not* differ decisively in prediction-time latency. Where they differ decisively is *training* cost: HHO-XQI trained end to end (four base learners plus the HHO fusion search) in 3.92 s, versus 575.7 s for 20 CNN epochs on the same hardware – a factor of approximately 147 ×. We state this precisely to avoid ambiguity: HHO-XQI's central practical advantage over the CNN is its training-time efficiency, not its prediction-time efficiency, and any deployment argument in this paper (Section 6) is built on that specific, measured difference rather than on an assumed inference-speed gap.

Table 11 compares the CNN branch with HHO-XQI directly on accuracy, per-image inference latency, and training time. The

CNN achieves markedly higher accuracy at CPU inference latency comparable to (not higher than) the classical pipeline and lower latency still with GPU acceleration; its cost instead appears in training time, which motivates the two-tier deployment scenario discussed in Section 6.

6. Discussion

6.1. Revisiting the research gap and novelty

The four gaps identified in Section 1 are addressed as follows. The four gaps identified in Section refsec: introductionare addressed as follows: First, the computational-cost gap is addressed by Tables 10 and 11, though the evidence is more nuanced than initially hypothesized: HHO-XQI is within 4.69 accuracy points of the CNN, and while its inference latency is *not* lower than the CNN's (both are approximately 11–12 ms/image on CPU), its training cost is approximately 147 × lower (3.92 s vs. 575.7 s), which is the more decisive computational advantage for a system that may need frequent retraining as grading criteria evolve. Second, the postharvest visual-grading gap is addressed by jointly modeling Damaged, Old, Ripe, and Unripe grades on a single, statistically validated pipeline rather than isolating disease or yield prediction. Third, the swarm-intelligence gap is addressed by using HHO to optimize a fusion-weight vector and construct a continuous, explainable TQI score (Eq. 4), rather than only tuning a single classifier's hyperparameters as in most prior HHO-for-agriculture studies [25, 26]. Fourth, and most importantly, the statistical-rigor gap is addressed by reporting – transparently, including a null result (the ripeness-index features slightly reducing ensemble accuracy, Table 7) and a nonsignificant comparison (HHO-XQI vs. extra trees alone, Table 6) – the full battery of ablation, significance testing, sensitivity analysis, and feature attribution that this literature has historically lacked [27].

6.2. Threats to validity

Internal validity. The internal HHO-validation split used to optimize fusion weights was always carved from the training portion only, and the final held-out validation set (or, for the repeated cross-validation (CV) protocol, the held-out fold) was never used for weight optimization; nonetheless, all folds were drawn from the same underlying dataset distribution, and repeated folds were not fully statistically independent of one another (Section 5), which is a well-known limitation of repeated k-fold cross-validation that can understate the true variance of the paired difference and should be considered when interpreting the Wilcoxon p-values in Table 6. Relatedly, this study relies on the dataset publisher's fixed train/validation partition (Section 4) and did not run an independent near-duplicate check across the two partitions; while no such duplication is documented by the publisher, we did not verify its absence ourselves, and we disclose this explicitly rather than assume a leakage-free split.

Construct validity. The 17 hand-designed ripeness/damage indices (Section 4) reflect the authors' domain assumptions about which color and texture cues indicate ripeness or damage; the ablation study (Table 7) shows that these indices are not strictly necessary once full HSV and Lab channel statistics are included, suggesting that the underlying color spaces already capture most of the relevant signal and that the specific hand-designed indices should not be over-interpreted as uniquely necessary.

Statistical conclusion validity. The nonsignificant result for HHO-XQI versus extra trees (Table 6) is reported honestly; a larger number of independent (non-overlapping) dataset replications, ideally from separate data collection sessions or facilities, would be needed to further reduce the risk that this null result itself reflects limited statistical power rather than a true absence of difference.

External validity. The dataset originates from a single curated image source; the images were captured under comparatively controlled conditions (single fruit, relatively uniform background) rather than a live packing-line camera with motion blur, variable illumination, occlusion by other fruit, or conveyor vibration. Performance under such field conditions is expected to degrade relative to the figures reported here, and this study does not claim generalization beyond the evaluated dataset without further field validation. Tomato variety, cultivar, and regional lighting differences may also shift the color distributions on which both the color-statistical features and the CNN depend.

Conclusion validity of the CNN comparison. The CNN was trained once using a single architecture (ResNet18) and a single hyperparameter configuration. A more exhaustive comparison across CNN architectures, training budgets, and augmentation strategies could change the accuracy gap reported in Table 11. Notably, the inference-latency comparison did *not* confirm our initial hypothesis that the CNN would be substantially slower per image; the measured CPU latencies were comparable, and the CNN was faster with the GPU acceleration. Therefore, any deployment argument in this study is grounded in the measured training time and interpretability differences rather than an assumed latency gap, and readers deploying on different hardware (e.g., a microcontroller without optimized Basic Linear Algebra Subprograms (BLAS)/tensor libraries) should remeasure both pipelines before drawing latency conclusions.

6.3. Industrial deployment scenario

The results support a concrete two-tier industrial deployment, motivated primarily by training cost and interpretability rather than inference latency, which Section 5 showed to be comparable between the two pipelines on CPU. On a packing line with a single line-scan or area camera, HHO-XQI can run as a first-tier, real-time screening classifier: at 82.96 images/second single-threaded (Table 10), it comfortably exceeds the throughput of a typical manual or mechanical sorting lane (commonly on the order of a few fruits per second). Its main deployment advantage is that it can be retrained from scratch in approximately 4 seconds when a new tomato variety, camera, or grading policy is introduced, versus nearly 10 minutes for the CNN (Table 11), and that its predictions are natively explainable via SHAP on named color descriptors (Figure 6) without requiring a separate visual-explanation step. Fruits whose TQI (Eq. 4) exceeds an operator-tunable threshold – particularly borderline Damaged/Old cases, which Figure 4 shows are the primary source of residual error – can be routed to a second tier: either human visual inspection or the higher-accuracy CNN + Grad-CAM branch, which Table 11 shows can run at comparable or even lower per-image latency than HHO-XQI when GPU acceleration is available, making it a viable audit layer wherever such hardware exists. Where only a basic CPU is available and frequent retraining is expected (e.g., a new farm or packing facility onboarding its own labeled images), HHO-XQI remains the more practical default; where a GPU is already provisioned and retraining is infrequent, the CNN branch may be deployed as the primary classifier directly, with HHO-XQI retained as a fast, interpretable fallback or audit cross-check. The Grad-CAM overlay

(Figure 9) additionally gives a human auditor a visual, interpretable justification for the CNN's grade decision, which is important for operator trust and dispute resolution in a commercial grading context. This tiered design operationalizes the accuracy–interpretability–training-cost trade-off quantified in Section 5, rather than assuming a single model must be simultaneously the fastest, most accurate, and most explainable.

7. Conclusion

This study presents HHO-XQI, an HHO-optimized Explainable Quality Index framework for four-class tomato grading. On a held-out validation split of 725 images, the framework reached 94.62% accuracy and 92.90% macro-F1 by fusing four color-statistical base learners with HHO-optimized weights, improving on the strongest single learner (extra trees, 94.34%/92.66%) although this specific improvement was not statistically significant under repeated cross-validation ($p = 0.071$). HHO-XQI was, however, significantly better than the three remaining baselines ($p < 10^{-4}$). A comprehensive validation battery – feature-space and ensemble-composition ablation, HHO hyperparameter sensitivity analysis, a 10-seed convergence study, and SHAP feature attribution – confirmed that the framework's behavior is stable, interpretable, and consistent with known tomato color physiology, while also transparently reporting two honest null/negative findings (the marginal value of the hand-designed ripeness indices, and the nonsignificant gain over extra trees alone). A fine-tuned ResNet18 CNN with Grad-CAM explanations was trained on the identical split as an independent accuracy and explainability benchmark, reaching 99.31% accuracy; contrary to our initial expectation, its per-image CPU inference latency was comparable to HHO-XQI's (and lower still with GPU acceleration), while its training cost was approximately 147 times higher. Together, these results support HHO-XQI as an interpretable, statistically validated, near-instantly retrainable first-stage industrial grading tool, with the CNN + Grad-CAM branch serving as a more accurate second-stage auditing tool in a two-tier deployment chosen for its accuracy and training-cost profile rather than an assumed latency penalty. Future work should (i) benchmark HHO against PSO, the GWO, and DE as alternative fusion-weight optimizers under identical base learners, fitness function, and cross-validation protocol, directly testing the optimization-theoretic justification given in Section 3.6; (ii) extend the SHAP attribution from the dominant extra-trees learner to a model-agnostic, fusion-level explanation of the full four-learner ensemble; (iii) evaluate the framework on field-collected packing-line imagery to address the external-validity threat identified in Section 6; and (iv) explore multi-objective HHO variants that jointly optimize accuracy and a precision/recall balance for the Damaged class.

Acknowledgment

The authors sincerely appreciate the research support of the Posts and Telecommunications Institute of Technology in Ho Chi Minh City, Vietnam, and King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand, for this research.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The tomato image dataset used in this study is the publicly available Tomatoes dataset published by Enalis (2023) on Kaggle: https://www.kaggle.com/datasets/enalis/tomatoes-dataset. Per the journal's editorial guidance, this dataset is not listed as a numbered reference in the reference list; the full dataset citation and access link are provided here instead. The feature-extraction, HHO optimization, and CNN training code are available from the corresponding author upon reasonable request.

Author Contribution Statement

Van Van Chau: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **Minh Tuan Nguyen:** Conceptualization, Methodology, Validation, Data curation, Writing – review & editing, Project administration. **Phayung Meesad:** Resources, Writing – review & editing, Supervision, Project administration.

References

[1] Hoang, T.-M., Bui, V.-H., Nguyen, V.-S., Doan, D.-T., Dang, H.-A., & Pham, A.-T. (2026). A comprehensive evaluation of lightweight deep learning models for tomato disease classification on edge computing environments. *Scientific Reports*, *16*(1), 12320. https://doi.org/10.1038/s41598-026-42439-6

[2] Inuwa, A. B., Ahmad, A. S., Auwal, S. T., Muhammad, M., Rabiu, A. L., & Babangida, I. (2025). Advances in sensor-based and machine learning techniques for automated tomato sorting and grading: A review. *Journal of Sustainable Engineering and Technology*, *2*(1), 1–12.

[3] Panilag, S. P. A., Rosal, J. E. C., Tamayo, A. J. R., Galindo, J. A. M., & Tecson, C. A. (2024). Ripeness classification of *Lycopersicon esculentum* employing faster region-based convolutional neural network with Inception version 2. In *2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 222–227. https://doi.org/10.1109/iicaiet62352.2024.10730102

[4] Odah, K. A., Houetohossou, S. C. A., Houndji, V. R., & Glèlé Kakaï, R. L. (2025). Machine learning techniques for tomato yield prediction: A comprehensive analysis. *Smart Agricultural Technology*, *12*, 101067. https://doi.org/10.1016/j.atech.2025.101067

[5] Al, khatib., G, A. M., Alshaib, B. M., Mishra, P., Tiwari, S., Alshalaby, M. A., & Kumari, B. (2026). Forecasting tomato production in major Asian producers: A comparative study of ARIMA, exponential smoothing, score-driven models, and XGBoost. *Scientific Reports*, *16*(1), 15722. https://doi.org/10.1038/s41598-026-46110-y

[6] Phan, Q.-H., Nguyen, V.-T., Lien, C.-H., Duong, T.-P., Hou, M. T.-K., & Le, N.-B. (2023). Classification of tomato fruit using YOLOv5 and convolutional neural network models. *Plants*, *12*(4), 790. https://doi.org/10.3390/plants12040790

[7] Moya, V., Guerra, M., Pazmiño, K., Abedrabbo, F., Chicaiza, F. A., & Pozo-Espín, D. (2025). Tomato classification with YOLOv8: Enhancing automated sorting and quality

assessment. *Smart Agricultural Technology*, *12*, 101221. https://doi.org/10.1016/j.atech.2025.101221

[8] Khan, A., Hassan, T., Shafay, M., Fahmy, I., Werghi, N., Mudigansalage, S., & Hussain, I. (2023). Tomato maturity recognition with convolutional transformers. *Scientific Reports*, *13*(1), 22885. https://doi.org/10.1038/s41598-023-50129-w

[9] Vijayan, S., & Chowdhary, C. L. (2025). Hybrid feature optimized CNN for rice crop disease prediction. *Scientific Reports*, *15*(1), 7904. https://doi.org/10.1038/s41598-025-92646-w

[10] Latha, P., & Kumaresan, P. (2025). Deep learning model optimization for crop prediction and recommendation using Harris Hawks optimization. *Environmental Research Communications*, *7*(4), 045008. https://doi.org/10.1088/2515-7620/adc548

[11] Li, G., Shi, W., Gu, Z., Cheng, S., Li, Y., & Kou, Z. (2026). Parameter optimization of support vector machines and feature selection based on an improved Harris Hawks optimization. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *E109-A*(3), 676–687. https://doi.org/10.1587/transfun.2025EAP1043

[12] Nikam, S. B., Lande, S. B., Nagalkar, V. J., Wakchaure, G. C., & Suresh Kumar, P. (2025). Predictive classification model for quality grading and maturity detection of dragon fruit using fused deep CNN feature and ensemble learning. *Journal of Food Processing and Preservation*, *2025*(1), 6938071. https://doi.org/10.1155/jfpp/6938071

[13] Xu, M., Cao, L., Lu, D., Hu, Z., & Yue, Y. (2023). Application of swarm intelligence optimization algorithms in image processing: A comprehensive review of analysis, synthesis, and optimization. *Biomimetics*, *8*(2), 235. https://doi.org/10.3390/biomimetics8020235

[14] Rajalaxmi, R. R., Saradha, M., Fathima, S. K., Sathish Kumar, V. E., Sandeep Kumar, M., & Prabhu, J. (2023). An improved MangoNet architecture using Harris Hawks optimization for fruit classification with uncertainty estimation. *Journal of Uncertain Systems*, *16*(1), 2242006. https://doi.org/10.1142/S1752890922420065

[15] Malini, A. H., Archana, T., Rohini, M., Lakshana, D., & Jayashree, P. (2024). Automated tomato classification using CNN-ELM with enhanced whale optimization for grading. In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 1200–1205. https://doi.org/10.1109/ICACRS62842.2024.10841492

[16] Shrivastav, S., Jindal, V., & Eswarawaka, R. (2024). A fusion method for detection and classification of diseases in tomato plants using swarm-based deep learning. *International Journal of Experimental Research and Review*, *45*(spl), 135–152. https://doi.org/10.52756/ijerr.2024.v45spl.011

[17] Quach, L. D., Quoc, K. N., Quynh, A. N., Thai-Nghe, N., & Nguyen, T. G. (2023). Explainable deep learning models with gradient-weighted class activation mapping for smart agriculture. *IEEE Access*, *11*, 83752–83762. https://doi.org/10.1109/access.2023.3296792

[18] Tomas, M. C., Beltran, A. J., Aranez, Y. E., & Britanico, E. (2024). Tomato (*Solanum lycopersicum* L.) fruit ripeness classification based on VGG16 convolutional neural network. In *2024 The 8th International Conference on Machine Learning and Soft Computing*, 165–171. https://doi.org/10.1145/3647750.3647776

[19] Naveenkumar, M., & Nandagopal, S. (2025). Adaptive hybrid segmentation combined with meta heuristic optimization in transfer learning for plant leaf disease classification. *Scientific Reports*, *15*(1), 9838. https://doi.org/10.1038/s41598-025-93225-9

[20] Alharbi, A. H., Mattar, E. A., Elkenawy, S., & Ghoneim, M. E. (2025). Potato leaf disease detection using deep reinforcement learning and the Ocotillo optimization algorithm. *Potato Research*, *68*(4), 4493–4524. https://doi.org/10.1007/s11540-025-09941-2

[21] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4765–4774.

[22] Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, *97*, 849–872. https://doi.org/10.1016/j.future.2019.02.028

[23] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

[24] Karim, M. J., Goni, M. O. F., Nahiduzzaman, M., Ahsan, M., Haider, J., & Kowalski, M. (2024). Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight CNN architecture and Grad-CAM. *Scientific Reports*, *14*(1), 16022. https://doi.org/10.1038/s41598-024-66989-9

[25] Yonbawi, S., Alahmari, S., Satyanarayana Murthy, T., Maddala, P., Laxmi Lydia, E., Kadry, S., & Kim, J. (2023). Harris Hawks optimizer with graph convolutional network based weed detection in precision agriculture. *Computer Systems Science and Engineering*, *46*(2), 1533–1547. https://doi.org/10.32604/csse.2023.036296

[26] Rajeena, P. P., Ismail, F., N, W., & Ali, M. A. S. (2023). A metaheuristic Harris Hawks optimization algorithm for weed detection using drone images. *Applied Sciences*, *13*(12), 7083. https://doi.org/10.3390/app13127083

[27] Majdalawieh, M., Martins, C., Radi, M., Alaraj, M., & Khan, S. (2025). Precision agriculture in the age of AI: A systematic review of machine learning methods for crop disease detection. *Smart Agricultural Technology*, *12*, 101491. https://doi.org/10.1016/j.atech.2025.101491

How to Cite: Van Chau, V., Meesad, P., & Nguyen, M. T. (2026). Harris Hawks Optimization–Optimized Explainable Quality Index for Tomato Grading. *Journal of Data Science and Intelligent Systems*. https://doi.org/10.47852/bonviewJDSIS620211565