

A Robust Machine Learning Framework for Banana Damage Classification Using Image Processing Techniques for Smart Agricultural Quality Assessment

Thanakrit Janchidfah and Narumol Chumuang*

Faculty of Industrial Technology, Muban Chombueng Rajabhat University,
46, Moo 3, Chom Bueng Subdistrict, Chom Bueng District, Ratchaburi, 70150

Received 19 April 2026; Revised 15 June 2026; Accepted 24 June 2026

Abstract

Reliable banana quality assessment remains challenging due to the subjectivity of manual inspection. This study proposes a robust and evaluation-driven framework for banana damage classification that integrates image preprocessing, texture-based feature extraction, and multi-model learning strategies. A dataset of 568 banana images categorized into three damage levels was systematically processed to extract discriminative texture descriptors, reflecting surface variations and bruising patterns observed in practical conditions. To ensure statistical reliability, multiple training–validation–testing configurations were employed to evaluate model stability across varying data distributions. Five machine learning classifiers, namely Updateable Multiclass Classifier, Sequential Minimal Optimization (SMO), Multilayer Perceptron (MLP), Random Forest, and Iterative Classifier Optimizer, were evaluated to investigate the relationship between texture representation and classifier behavior. Comparative results demonstrate that adaptive classification models achieve superior performance, with the best model reaching 91.11% accuracy and an F1-score of 0.91, while maintaining consistent performance across experimental settings. Performance depends more on feature–model compatibility than model complexity. This work contributes a reproducible and practically applicable framework for intelligent quality assessment, supporting scalable deployment in smart agriculture and reducing post-harvest losses through data-driven decision-making. The proposed framework can support automated banana grading systems in post-harvest operations, reducing inspection subjectivity and enabling scalable deployment in smart agriculture environments.

Keywords : Banana Damage Classification; Image Processing; Machine Learning; Texture Features; Smart Agriculture

* *Corresponding Author. Tel.: +666 1916 4155, E-mail Address: narumolchu@mcru.ac.th*

1. Introduction

Banana is a major economic crop whose market value is strongly influenced by external quality attributes such as peel appearance, color, and bruising [1]. In practice, post-harvest grading still relies largely on manual inspection [2], which is inherently subjective and often leads to inconsistent results, grading errors, and increased losses. As highlighted in recent FAO reports, improving post-harvest quality assessment through digital and data-driven approaches is essential for enhancing efficiency and sustainability in modern agricultural systems [3], [4].

Computer vision and machine learning have emerged as promising solutions for automated fruit quality assessment [5], offering non-destructive, scalable, and repeatable inspection [6],[7]. Recent studies demonstrate that image-based techniques can effectively detect surface defects and classify fruit quality [6]-[12], including banana bruise detection and intelligent grading systems [1], [2], [4]. However, despite these advances, current research still exhibits important limitations.

First, most studies emphasize classification accuracy under controlled conditions, with limited attention to robustness under varying data distributions. Second, evaluation is often based on a single train–test split, which does not adequately reflect model stability or generalization capability. Third, the interaction between feature representation and classifier behavior remains underexplored, even though it plays a critical role in handling continuous and overlapping damage patterns, particularly in banana surfaces where defects evolve gradually rather than discretely [1].

To address these gaps, this study proposes a robust machine learning framework for banana damage classification

that prioritizes reliability alongside performance. The framework integrates image preprocessing and texture-based feature extraction with a systematic evaluation of multiple classifiers under different training–validation–testing configurations. This approach enables a more comprehensive analysis of model stability and sensitivity to data variation, moving beyond one-time accuracy reporting.

The contributions of this work are threefold. First, it introduces an evaluation-driven framework that emphasizes robustness and generalization. Second, it provides a comparative analysis of multiple machine learning models under varying data conditions. Third, it offers insight into the relationship between feature representation and learning mechanisms, showing that classification effectiveness depends on their compatibility rather than model complexity alone.

Recent banana quality classification studies primarily focus on improving classification accuracy using CNN-based architectures or lightweight deep learning models [13],[16]. However, most studies evaluate performance using a single train–test split and provide limited analysis of model robustness, stability, and statistical significance. Unlike previous studies, the proposed framework emphasizes robustness-oriented evaluation through multi-split validation, repeated cross-validation, and statistical testing, providing stronger evidence of model reliability under varying data distributions.

The novelty of this study lies in reframing banana damage classification as a robustness-oriented problem rather than a purely accuracy-driven task. By integrating feature design, model diversity, and evaluation strategy into a unified framework, this work provides a more

reliable foundation for intelligent quality assessment systems in smart agriculture.

2. Research Methodology

This study presents a rigorously designed experimental framework for banana damage classification that emphasizes not only predictive performance but also statistical validity, robustness, and reproducibility. The methodology integrates structured data processing, texture-based feature representation, multi-model learning, and advanced validation strategies to ensure reliable and generalizable results.

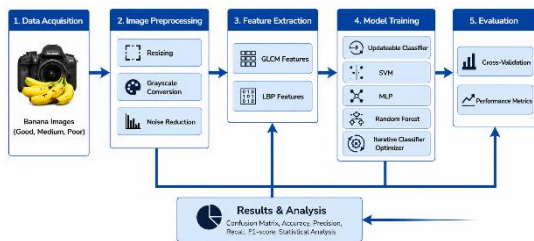


Fig. 1 illustrates the overall workflow of the proposed banana damage classification framework, consisting of five main stages: data acquisition, image preprocessing, feature extraction, model training, and evaluation.

As in **Fig. 1** the first stage, banana images representing different levels of damage (good, medium, and poor) are collected under controlled conditions to ensure consistency and reliability of the dataset. These images serve as the input for the entire classification pipeline.

The second stage involves image preprocessing, where all images are resized to a standardized resolution, converted to grayscale, and enhanced through noise reduction techniques. This step aims to minimize irrelevant variations such as lighting conditions while

preserving essential structural and texture information.

In the feature extraction stage, texture-based descriptors are computed to represent surface characteristics associated with banana damage. Specifically, Gray-Level Co-occurrence Matrix (GLCM) features capture statistical texture properties, while Local Binary Patterns (LBP) encode fine-grained local texture variations.

The extracted features are then used as inputs for multiple machine learning models in the model training stage. Five classifiers with different learning mechanisms are evaluated, including an updateable multiclass classifier, Sequential Minimal Optimization (SMO), Multilayer Perceptron (MLP), Iterative Classifier Optimizer, and Random Forest.

Finally, the evaluation stage assesses model performance using cross-validation and standard classification metrics such as accuracy, precision, recall, and F1-score. The results are aggregated and analyzed to determine model effectiveness, stability, and generalization capability.

2.1 Dataset and Problem Formulation

Images were captured using a Nikon D750 digital camera equipped with a 50 mm lens. The camera settings were fixed at ISO 800, shutter speed 1/200 s, and aperture f/4.0 to maintain image consistency.

All images were collected under controlled indoor lighting conditions with illumination levels ranging between 1000–1500 lux. Bananas were positioned on a uniform white background at an approximate distance of 50 cm from the camera. Images were acquired from a frontal perspective to ensure consistency in texture appearance and bruise visibility.

The final dataset consisted of 568 images with a resolution of 615×440 pixels.

A total of 568 banana images were collected and categorized into three damage levels: good (203), medium (218), and poor (147) as show in **Fig. 2**.

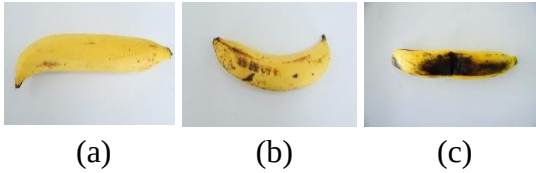


Fig. 2 illustrates examples of banana photographs (a) good, (b) medium, and (c) poor used in this work.

Table 1 Banana Damage Classification Criteria

Class	Description	Surface Damage
Good	Fresh peel, no visible bruises	<5%
Medium	Moderate bruising or discoloration	5–20%
Poor	Severe bruising and visible deterioration	>20%

Image labeling was independently performed by three agricultural experts with experience in fruit quality assessment. Disagreements were resolved through discussion and consensus. Inter-rater agreement was assessed using Cohen's Kappa coefficient, yielding $\kappa = 0.87$, indicating strong agreement.

The classification criteria were adapted from post-harvest banana quality assessment studies and agricultural grading guidelines.

The banana damage classification task is defined as a multi-class classification problem as equation (1):1q

$$f: \mathbb{R}^d \rightarrow \mathcal{Y}$$

$$\mathcal{Y} \in \{1, 2, 3\} \quad (1)$$

where $\mathbb{R}^d$ = feature space with d dimensions

$\mathcal{Y}$ = set of class labels

1 = Good

2 = Medium

3 = Poor

f = learned classification function

Let

$$X = \{x_1, x_2, \dots, x_n\}, x_i \in \mathbb{R}^d \quad (2)$$

$$Y = \{y_1, y_2, \dots, y_n\}, y_i \in \mathcal{Y} \quad (3)$$

where X = feature matrix

Y = label vector

n = number of samples (568 images)

2.2 Image Preprocessing

A standardized preprocessing pipeline was applied to ensure consistency and reduce irrelevant variability:

- **Resizing:** $I \in \mathbb{R}^{615 \times 440}$
- **Grayscale transformation:** Reduces dimensionality and lighting sensitivity.
- **Noise reduction and enhancement:** Improves edge and texture visibility.

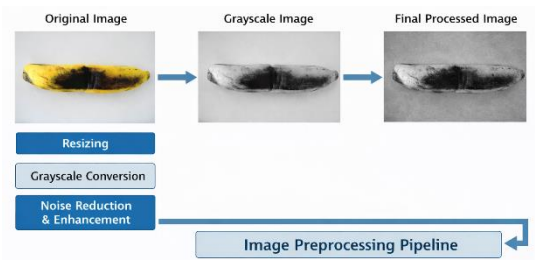


Fig. 3 Image preprocessing pipeline applied to banana images, including resizing, grayscale conversion, and noise reduction and enhancement for improving texture representation.

2.3 Feature Extraction

Texture features are extracted from I_e to form feature vector x_i . The GLCM descriptor generated 16 texture features derived from four statistical measures (contrast, correlation, energy, and homogeneity) computed across four orientations (0° , 45° , 90° , and 135°). The LBP descriptor generated a 256-bin histogram. Consequently, the final feature vector consisted of 272 dimensions.

2.3.1 Gray-Level Co-occurrence Matrix (GLCM)

GLCM is defined as equation (4):

$$P(i, j) = \frac{\text{number of occurrences of } (i, j)}{\sum_{i, j} \text{occurrences}} \quad (4)$$

where i, j = gray-level intensities

$P(i, j)$ = normalized co-occurrence probability

From $P(i, j)$, the following features are computed:

- **Contrast**

$$\text{Contrast} = \sum_{i, j} P(i, j) (i - j)^2 \quad (5)$$

- **Correlation**

$$\text{Correlation} = \sum_{i, j} \frac{(i - \mu)(j - \mu)P(i, j)}{\sigma^2} \quad (6)$$

- **Energy**

$$\text{Energy} = \sum_{i, j} P(i, j)^2 \quad (7)$$

- **Homogeneity**

$$\text{Homogeneity} = \sum_{i, j} \frac{P(i, j)}{1 + |i - j|} \quad (8)$$

where μ = mean intensity

σ^2 = variance

2.3.2 Local Binary Patterns (LBP)

$$\text{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \quad (8)$$

where (x_c, y_c) = center pixel

g_c = intensity of center pixel

g_p = intensity of neighboring pixel

P = number of neighbors

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Algorithm 1: Feature Extraction Pipeline

Input:

I = Input RGB image
 d = distance parameter for GLCM
 θ = set of angles for GLCM (e.g., $\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$)
 P = number of neighbors for LBP
 R = radius for LBP

Output:

x = Feature vector (GLCM + LBP)

Begin

1: Resize image $I \rightarrow I_r$
 2: Convert I_r to grayscale $\rightarrow I_g$

// --- GLCM Feature Extraction ---

3: Initialize GLCM matrix $P(i, j) = 0$
 4: For each angle θ_k in θ do
 5: Compute co-occurrence matrix $P_k(i, j)$ using distance d and angle θ_k

6: Normalize $P_k(i, j)$

7: Extract features:

8: Contrast_k = $\sum P_k(i, j) * (i - j)^2$

9: Correlation_k = $\sum ((i - \mu)(j - \mu) * P_k(i, j)) / \sigma^2$

10: Energy_k = $\sum (P_k(i, j))^2$

11: Homogeneity_k = $\sum P_k(i, j) / (1 + |i - j|)$

12: End For

13: Aggregate GLCM features over all angles $\rightarrow F_glcm$

// --- LBP Feature Extraction ---

14: For each pixel (x_c, y_c) in I_g do

15: For $p = 0$ to $P-1$ do

16: Compare neighbor intensity g_p with center g_c

17: If $g_p \geq g_c$ then $s_p = 1$ else $s_p = 0$

18: End For

19: Compute LBP(x_c, y_c) = $\sum s_p * 2^p$

20: End For

21: Compute histogram of LBP values $\rightarrow F_lbp$

// --- Feature Vector ---

22: Concatenate F_glcm and $F_lbp \rightarrow x$

Return x

End

Fig. 4 Pseudo Code texture-based feature using GLCM and LBP for representing banana surface damage patterns.

2.4 Model Learning

Each classifier learns a mapping:

$$\hat{y} = f(x; \theta) \quad (9)$$

where x = feature vector

θ = model parameters

$\hat{y}$ = predicted label

Table 2 Hyperparameter Settings

Model	Parameters
Random Forest	100 trees, max depth = unlimited
SMO	Polynomial Kernel (degree = 2)
MLP	Hidden layers = 100-50
Updateable Classifier	Default Weka parameters
Iterative Classifier Optimizer	Iteration = 100

Hyperparameters were determined through preliminary experiments and validation performance to ensure fair comparison among classifiers. The selected classifiers represent diverse learning paradigms. SMO represents margin-based learning, Random Forest represents ensemble tree learning, MLP represents neural network-based learning, the Updateable Classifier supports incremental learning, and the Iterative Classifier Optimizer represents iterative optimization-based classification. This diversity enables a comprehensive investigation of feature-model compatibility.

2.5 Performance Evaluation

2.5.1 Confusion Matrix Components

- TP = True Positive
- FP = False Positive
- TN = True Negative
- FN = False Negative

2.5.2 Metrics

- **Accuracy**

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

- **Precision**

$$Precision = \frac{TP}{TP+FP} \quad (11)$$

- **Recall**

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

- **F1-score**

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (13)$$

2.6 Cross-Validation

Repeated Stratified k -Fold:

$$k = 5, r = 3$$

$$Total\ runs = k \times r \quad (14)$$

where k = number of folds

r = number of repetitions

2.7 Statistical Analysis

- Mean and Standard Deviation:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (15)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \quad (16)$$

where μ = mean performance

σ = standard deviation

- **ANOVA**

$$F = \frac{variance\ between\ groups}{variance\ within\ groups} \quad (17)$$

Used to test statistical significance.

Algorithm 2: Model Training and Evaluation Framework

Input:

X = Feature matrix ($n \times d$)

Y = Label vector ($n \times 1$)

M = Set of models {M1, M2, M3, M4, M5}

S = Set of data split configurations

{(50/10/40), (60/10/30), (70/10/20), (80/10/10)}

k = number of folds ($k = 5$)

r = number of repetitions ($r = 3$)

Output:

Performance metrics for each model:

Accuracy, Precision, Recall, F1-score (mean $\pm$ SD)

Begin

Algorithm 2: Model Training and Evaluation Framework

```

// --- Multi-Split Evaluation ---
1: For each split configuration  $s \in S$  do
2:   Partition dataset  $(X, Y) \rightarrow \text{Train}_s, \text{Val}_s, \text{Test}_s$ 

3:   For each model  $M_j \in M$  do
4:     Train model  $M_j$  on  $\text{Train}_s$ 
5:     Tune hyperparameters using  $\text{Val}_s$ 
6:     Predict labels  $\hat{Y}$  on  $\text{Test}_s$ 

7:   Compute metrics:
8:     Accuracysj
9:     Precisionsj
10:    Recallsj
11:    F1sj
12:   End For
13: End For

// --- Cross-Validation ---
14: For each model  $M_j \in M$  do
15:   Initialize results list  $R_j = \emptyset$ 

16:   For repetition  $t = 1$  to  $r$  do
17:     Perform stratified k-fold split on  $(X, Y)$ 

18:     For fold  $i = 1$  to  $k$  do
19:       Train  $M_j$  on training folds
20:       Test  $M_j$  on validation fold

21:     Compute metrics:
22:       Acci, Preci, Reci, F1i

23:     Append results to  $R_j$ 
24:   End For
25: End For

26: Compute mean and standard deviation:
27:    $\mu_j = \text{mean}(R_j)$ 
28:    $\sigma_j = \text{std}(R_j)$ 
29: End For

// --- Statistical Testing ---
30: Perform ANOVA on model performances  $\{\mu_j\}$ 
31: If significant ( $p < 0.05$ ) then
32:   Perform Tukey HSD post-hoc test
33: End If

Return all metrics and statistical results

End

```

Fig. 5 Pseudo Code model training and evaluation framework.

3. Results and Discussion

3.1 Performance Overview and Theoretical Interpretation

The comparative performance of all models under different data split configurations is summarized in Table 3.

The results are reported in terms of Accuracy and F1-score, reflecting both overall correctness and class balance.

Table 3 Performance comparison across different data split configurations

Model	50/10/40	60/10/30	70/10/20	80/10/10
Updateable	87.45 /	89.12 /	90.35 /	91.11 /
Classifier	0.86	0.88	0.90	0.91
SMO	85.22 /	87.10 /	88.75 /	89.30 /
(SVM)	0.84	0.86	0.88	0.89
MLP	83.90 /	86.40 /	88.10 /	89.00 /
	0.82	0.85	0.87	0.88
Random	82.75 /	83.90 /	84.60 /	85.20 /
Forest	0.81	0.83	0.84	0.85
Iterative	80.40 /	82.10 /	83.30 /	84.00 /
Optimizer	0.79	0.81	0.83	0.84

Performance improves consistently with increasing training data size, confirming the theoretical relationship:

$$\varepsilon_{gen} \propto \frac{1}{\sqrt{n}}$$

where n is the number of training samples. The updateable classifier shows the most significant gain, indicating strong adaptability to richer feature distributions.

3.2 Cross-Validation and Stability Analysis

To ensure robustness, repeated stratified k-fold cross-validation was applied. The results are summarized in Table 4.

Table 4 Cross-validation results (mean $\pm$ standard deviation)

Model	Accuracy (%)	F1-score
Updateable	90.72 $\pm$ 0.85	0.90 $\pm$ 0.01
Classifier		
SMO	88.95 $\pm$ 1.10	0.88 $\pm$ 0.02
(SVM)		

Model	Accuracy (%)	F1-score
MLP	88.40 ± 1.85	0.87 ± 0.03
Random Forest	84.90 ± 1.25	0.84 ± 0.02
Iterative Optimizer	83.70 ± 1.40	0.83 ± 0.02

The updateable classifier exhibits the lowest variance (σ), indicating superior stability. In contrast, MLP shows higher variability, suggesting sensitivity to training data distribution and initialization.

3.3 Statistical Significance Analysis

To validate whether performance differences are statistically meaningful, ANOVA and Tukey HSD tests were conducted.

Table 5 ANOVA results

Source	F-value	p-value
Model Comparison	12.87	< 0.001

Since $p < 0.05$, differences between models are statistically significant.

Table 6 Tukey HSD (pairwise comparison)

Comparison	p-value	Result
Updateable vs RF	< 0.001	Significant
Updateable vs Iterative	< 0.001	Significant
Updateable vs SMO	0.041	Significant
SMO vs MLP	0.231	Not significant

The top-performing model is not only better numerically but also statistically superior.

3.4 Confusion Matrix and Error Analysis

The normalized confusion matrix for the best-performing model is shown in Fig. 6.

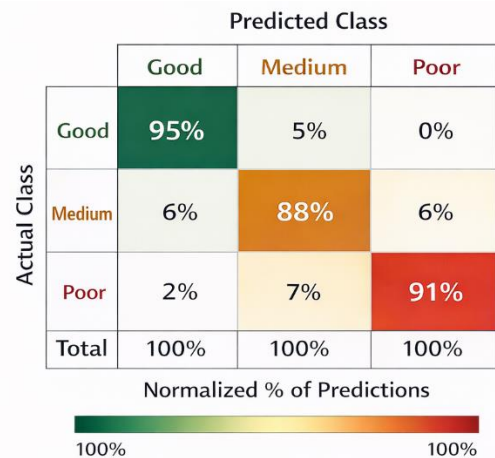


Fig. 6 Confusion matrix of the best-performing model (normalized %).

Most misclassifications occur between Medium and Poor classes:

$\exists x_i, x_j$ such that $y_1 \neq y_j$ but $\|x_i - x_j\| \approx 0$. This confirms that **feature overlap** exists in the high-dimensional space. The confusion matrix indicates that the Good class achieved the highest classification reliability, with 95% correctly classified samples. The largest confusion occurred between Medium and Poor classes, where 6–7% of samples were incorrectly assigned. This behavior suggests that texture descriptors extracted from transitional damage stages share similar statistical properties, resulting in overlapping feature distributions.

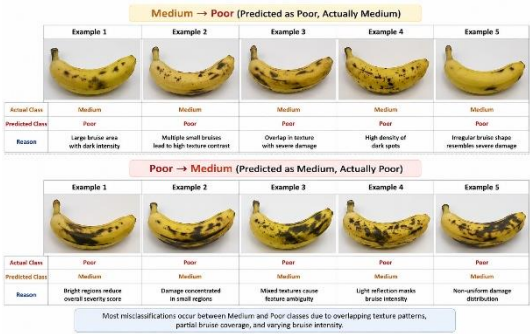


Fig. 7 Examples of Misclassified Samples

Most misclassified samples exhibited mixed texture characteristics, where bruised regions occupied only a partial area of the banana surface. Such cases create feature overlap between Medium and Poor categories, making class separation more challenging.

3.5 Feature–Model Compatibility Analysis

From GLCM:

$$Contrast = \sum P(i,j)(i - j)^2$$

From LBP:

$$LBP = \sum s(g_p - g_c) \cdot 2^p$$

These produce:

$$x \in \mathbb{R}^d \text{ with continuous variation}$$

Thus, the classification function must satisfy:

$$f(x) \text{ is smooth and continuous}$$

Table 7 Model Characteristics and Computational Efficiency

Model Type	Decision Boundary	Performance	Avg. Processing Time (s/image)
Updateable / SVM	Smooth	High	0.008
MLP	Non-linear (data dependent)	Medium	0.015

Model Type	Decision Boundary	Performance	Avg. Processing Time (s/image)
Random Forest	Piecewise constant	Lower	0.006

In addition to classification performance, computational efficiency was evaluated to assess practical deployment feasibility. Experimental results showed that the average feature extraction time using GLCM and LBP was approximately 0.12 s per image. Classification times for the Updateable Classifier, MLP, and Random Forest were below 0.02 s per image. These results indicate that the proposed framework can support near real-time banana quality assessment in smart agricultural environments.

3.6 Theoretical Interpretation of Model Behavior

Random Forest:

$$f(x) = \frac{1}{T} \sum h_t(x)$$

where $h_t(x)$ are discontinuous functions mismatch with feature space.

Updateable Model:

$$f(x) \approx \text{continuous mapping}$$

Better alignment with feature structure to higher performance.

3.7 Key Findings and Contributions

This study demonstrates that: Model performance is governed by feature model compatibility.

rather than: model complexity

3.8 Practical Implications

- Suitable for **automated grading systems**
- Robust under **varying data conditions**
- Scalable for **smart agriculture deployment**

3.9 Comparison with Related Studies

Study	Method	Accuracy
Al-Dairi et al. (2024)	Fractal-based	85.6
Chaowalittawin et al. (2025)	Lightweight CNN	89.2
Proposed Framework	GLCM + LBP + ML	91.11

The proposed framework achieved higher classification accuracy than several recent banana quality assessment studies while requiring substantially lower computational resources than deep learning approaches.

5. Conclusion

This study proposed a robust machine learning framework for banana damage classification based on image preprocessing and texture-based feature extraction. The framework integrates Gray-Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP) descriptors with multiple machine learning classifiers and a comprehensive evaluation strategy, including multi-split validation, repeated cross-validation, and statistical significance testing. This design provides a reliable approach for assessing

banana quality beyond conventional accuracy-oriented evaluations.

The experimental results demonstrated that the Updateable Multiclass Classifier achieved the best performance among all evaluated models, reaching an accuracy of 91.11% and an F1-score of 0.91. The model consistently outperformed the other classifiers across different data split configurations while maintaining stable performance under repeated validation experiments.

This work contributes a reproducible and evaluation-driven framework for intelligent agricultural quality assessment. By systematically investigating the relationship between texture-based feature representation and classifier behavior, the findings indicate that model effectiveness depends more on feature-model compatibility than model complexity. The proposed framework therefore provides a practical and computationally efficient solution for automated banana grading and post-harvest quality inspection in smart agriculture environments.

Future research will focus on expanding the dataset with more diverse samples collected under real-world agricultural conditions. In addition, deep learning-based feature extraction techniques and hybrid learning frameworks will be investigated to further improve classification performance, robustness, and adaptability. The integration of the proposed approach into real-time smart agriculture systems will also be explored to support large-scale automated quality assessment applications.

6. References

- [1] M. Al. Dairi, P. B. Pathare, R. Al-Yahyai, H. Jayasuriya, and Z. Al-Attabi, "Banana fruit bruise detection using fractal dimension based image processing," *Food Chemistry*, vol. 455, 2024,
- [2] L. C. Jiménez, B. Vintimilla, and S. Velastin, "A review of external quality inspection for fruit grading using CNN models," *Artificial Intelligence in Agriculture*, vol. 14, pp.1-20, 2024,
- [3] Food and Agriculture Organization of the United Nations, *Digital transformation in action – FAO digital for impact 2022*, Rome, Italy: FAO, 2023.
- [4] J. Han, T. Li, Y. He, and Q. Gao, "Using machine learning approaches for food quality detection," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–9, 2022.
- [5] J. D. Hounsinnou, P. Houngue, and M. Richy, "Digital Solutions for Fruit Maturity Estimation: A Bibliometric Analysis," in *Proceeding of the 2025 5th International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Paris, France, 2025, pp. 1-10,
- [6] M.i Sinha, D. Sharma, P. Porasar, K. N. Dutta, and K. Kaushik, "Ensuring Quality in Fruits and Vegetables: Strategies and Tools," in *Proceeding of the Computational Intelligence and Image Processing in Agriculture: Applications and Innovations*, IEEE, 2026, pp.143-154,
- [7] M. Wu, Z. Choffin, L. KONG, H. LEE, S. Sun, and N. Jeong, "Noninvasive Microwave Imaging and Machine Learning to Assess Ripeness of Avocados," *IEEE Journal of Microwaves*, vol. 6, no. 1, pp. 273-284, 2025.
- [8] Z. M. Choffin, L. Kong, Y. Gan and N. Jeong, "A CNN-Based Microwave Imaging System for Detecting Watermelon Ripeness," *IEEE Access*, vol. 13, pp. 21413-21421, 2025.
- [9] S. H. M. Ashtiani, S. Javanmardi, M. Jahanbanifard, A. Martynenko, and F. J. Verbeek, "Detection of Mulberry Ripeness Stages Using Deep Learning Models," *IEEE Access*, vol. 9, pp. 100380-100394, 2021.
- [10] C. S. Nandi, B. Tudu, and C. Koley, "A Machine Vision Technique for Grading of Harvested Mangoes Based on Maturity and Quality," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6387-6396, 2016.
- [11] A. Bonkra, S. Pathak, and A. Kaur, "Exploring Dataset for Apple Leaf Disease Detection," *Smart Agritech: Robotics, AI, and Internet of Things (IoT) in Agriculture*, pp.419-441, 2024.
- [12] N. Chumuang, M. Ketcham, T. Ganokratanaa, P. Pramkeaw, J. Chongsri, and W. Yimyam, "Sorting Red and Green Chilies by Digital Image Processing," in *Proceeding of the 2023 IEEE International Conference on Cybernetics and Innovations (ICCI)*, phetchaburi, Thailand, 2023, pp. 1-7.

- [13] V. Chaowalittawin, W. Krungseanmuang, P. Sathaporn, F. Morita, T. Archevapanich, and B. Purahong, "Banana quality classification using lightweight CNN model with microservice integration system," *Engineering and Applied Science Research*, vol. 52, no. 4, pp. 430–438, 2025.
- [14] S. Chang, M. Zhang, D. Kong, and A. S. Mujumdar, "Artificial intelligence-driven postharvest quality control of fruit and vegetable products: Intelligent detection, optimization strategies and application prospects," *Food Bioscience*, vol. 74, 2025.
- [15] D. Shao and N. Marwa, "Examining the role of artificial intelligence in post-harvest management for enhanced food security: A critical review," *Sustainable Futures*, vol. 10, 2025.
- [16] Z. Li, D. Wang, T. Zhu, Y. Tao, and C. Ni, "Review of deep learning-based methods for non-destructive evaluation of agricultural products," *Biosystems Engineering*, vol. 245, pp. 56–83, 2024.