Comparative Analysis of Machine Learning Models and Greylag Goose Optimization for High-Accuracy Apple Quality Grading

Marwa M. Eid[1,*], Anis Ben Ghorbal[2]

[1]Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 11152, Egypt
[2]Department of Mathematics and Statistics, Faculty of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia

Emails: mmm@ieee.org; assghorbal@imamu.edu.sa

Abstract

A significant barrier to sustaining and achieving successful quality evaluation of apples has been the biases and inefficiencies of manual inspection, which cannot be effectively applied to large-scale industrial processing. This paper fills this gap by building an automated machine learning system for apple quality classification, which combines physical (size, weight, firmness) and sensory (sweetness, acidity, juiciness) features. Four baseline models, including K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Gradient Boosting, were trained on a preprocessed dataset. Imputation, categorical encoding, and feature scaling were used. To enhance predictive accuracy, hyperparameter tuning of three metaheuristic optimization algorithms — Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Greylag Goose Optimization (GGO) — was employed. The findings indicated that KNN had the highest baseline accuracy of 93.6% and that its accuracy improved to 98.6% with GGO optimization, which was better than that of GA and PSO. These results indicate that GGO-optimized KNN can provide a high-quality and scalable approach to assessing the quality of apples in the industry, offering enhanced accuracy, reduced variability, and a high degree of practical applicability in supply chain decision-making, thereby increasing customer satisfaction and decreasing losses after harvesting.

Keywords: Apple Quality Assessment; Machine Learning; Greylag Goose Optimization (GGO); Food Quality Prediction; Smart Agriculture

1 Introduction

Ensuring and maintaining the quality of agricultural products has long been a central challenge in modern food production systems. Among these products, apples represent one of the most economically significant fruits worldwide, forming a critical part of global diets and agribusiness [1]. The quality of apples directly affects consumer satisfaction, market competitiveness, and the efficiency of supply chains. Poor quality control can result not only in economic losses but also in reputational damage for producers and distributors. Despite technological advances in farming and storage, quality assurance remains a bottleneck due to the difficulty of consistent, scalable evaluation of essential attributes such as size, shape, color, texture, firmness, and sugar content [2]. These attributes are inherently variable across orchards, seasons, and even within the same batch, making reliable classification highly challenging. Furthermore, many of these traits are qualitative and require human judgment, which is prone to subjectivity and inconsistency [3]. Traditional evaluation methods rely on manual inspection by trained experts, who visually and physically examine apples for conformity with quality standards. While this practice benefits from human intuition and experience, it suffers from limitations such as fatigue, time consumption, and inconsistency [4] [5]. At the industrial scale, where millions of apples may pass through grading and packaging facilities on a daily basis, manual inspection becomes not only impractical but also a major bottleneck to productivity [6]. Furthermore, physical inspection introduces risks associated with authenticity and traceability [7]. For instance, varying interpretations among evaluators can lead to discrepancies in quality grading, ultimately impacting brand image and consumer trust. Regulatory frameworks and consumer expectations demand high accuracy and standardization in food quality evaluation [8]. Manual systems, therefore, fail to provide the scalability, repeatability, and precision necessary for modern supply-chain operations [9]. This shortcoming motivates the adoption of automated approaches capable of operating with industrial efficiency while maintaining high accuracy and reliability [10]. In recent years, artificial intelligence (AI) and machine learning (ML) have gained prominence as transformative technologies in agriculture and food industries. AI refers broadly to systems designed to perform tasks that typically require human intelligence, whereas ML specifically focuses on algorithms that learn patterns from data to make predictions or decisions [11]. These technologies are particularly suited to agri-food contexts, where large datasets

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

derived from imaging, sensing, and spectroscopy must be analyzed for complex, nonlinear relationships. For apple quality assessment, ML offers the promise of reducing subjectivity, increasing consistency, and achieving near real-time classification of multiple quality dimensions [12, 13]. With ML-enabled systems, producers can deploy scalable solutions that ensure uniform grading, minimize post-harvest losses, and strengthen consumer trust through transparent and standardized evaluation. However, the effectiveness of ML models heavily depends on preprocessing, feature selection, and the preparation of data for training. Raw data from agricultural sensors and imaging systems are often noisy, incomplete, or high-dimensional, which can significantly reduce model performance if not properly addressed [14]. Preprocessing techniques, such as normalization, denoising, and feature engineering, are therefore crucial for aligning data with the learning algorithms. For apples, common measurable features include color intensity, weight, firmness, sweetness, and texture patterns. Identifying and emphasizing features with strong predictive value ensures that the resulting models generalize effectively across datasets [15, 16]. Proper preprocessing thus forms the foundation of a reliable automated framework by converting raw sensor data into decision-grade inputs. Beyond preprocessing, optimization of ML models is essential for maximizing classification performance. Standard training approaches often lead to suboptimal parameter settings, particularly in high-dimensional or nonlinear search spaces. Metaheuristic optimization algorithms have emerged as effective solutions for navigating such complex landscapes [17] [18]. These algorithms—ranging from evolutionary approaches to swarm-based and physics-inspired heuristics—are capable of balancing exploration and exploitation to identify near-optimal solutions. In this study, we integrate the Greylag Goose Optimization (GGO) algorithm with the k-Nearest Neighbors (KNN) classifier. GGO, inspired by the collective migratory behavior of geese, dynamically balances global and local search strategies to efficiently optimize model parameters [19]. KNN, on the other hand, is a simple yet powerful classification algorithm valued for its interpretability, robustness, and effectiveness in handling nonlinear decision boundaries. By combining GGO with KNN, our proposed framework aims to enhance predictive accuracy, sensitivity, and specificity, while maintaining computational efficiency and scalability for industrial deployment [20].

This paper is designed with four major objectives:

1. To replace traditional manual assessment methods with an automated ML-based framework for apple quality classification.
2. To adopt the KNN classifier as an interpretable and effective baseline model for quality prediction.
3. To integrate the GGO algorithm to optimize KNN parameters, thereby improving classification accuracy, sensitivity, and specificity.
4. To evaluate the repeatability and robustness of the proposed framework, ensuring practical viability for real-world applications.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of the relevant literature on machine learning, optimization techniques, and their applications in fruit-quality assessment; Section 3 describes in detail the material and methods adopted in this study, including dataset preparation, preprocessing strategies, model development, and the integration of the Greylag Goose Optimization algorithm with the KNN classifier; Section 4 presents the experimental results, reporting the performance of the proposed framework and comparisons with baseline and state-of-the-art methods; Section 5 offers a critical discussion of the findings, relating them to existing studies, highlighting their practical implications for industrial applications, and addressing current limitations; and finally, Section 6 summarizes the main contributions of this research, draws the overall conclusions, and suggests potential directions for future work.

2 Related Works

Automated visual inspection has become central to improving commercial quality, labor efficiency, and large-scale productivity in apple production. Early studies mainly focused on **external quality grading**. For example, Bhargava and Bansal [21] developed an automated grading system based on support vector machines (SVMs), reporting optimal accuracies of 96.81% and 93.00% on two datasets using k-fold cross-validation. These results illustrate the potential of traditional ML approaches, but they are often limited by handcrafted features and lack robustness under variable field conditions. More advanced solutions leverage deep learning and remote sensing: Biffi et al. [22] introduced an Adaptive Training Sample Selection (ATSS) approach that significantly outperformed RetinaNet, Faster R-CNN, and HRNet for detecting apples in dense orchards, even under occlusion. The practical use of center-point labeling demonstrated both efficiency and scalability for commercial deployments. Beyond external traits, **internal quality parameters** such as firmness and soluble solids content (SSC) are pivotal for determining marketability and consumer acceptance. Using hyperspectral imaging, Çetin et al. [23] predicted firmness with an ANN ($R^2 = 0.910$) and SSC with decision trees ($R^2 = 0.679$), identifying important spectral bands (511, 505, 704, and 689 nm). This underlines the value of spectral sensing for nondestructive evaluation. In the area of **disease detection**, Chao et al. [24] combined DenseNet and Xception architectures with an SVM classifier, achieving 98.82% accuracy while outperforming conventional CNNs in efficiency and robustness—demonstrating the potential of hybrid deep–shallow models for reliable early diagnosis. **Varietal identification** supports traceability across supply chains. Chen et al. [25] proposed MAFNet, a Multi-Attention Fusion CNN built on ResNet, which achieved 98.14% accuracy across 30 cultivars and two growth stages using leaf images. Similarly, advances in **harvesting automation** aim to reduce labor intensity. Chu et al. [26] developed Suppression Mask R-CNN to address complex lighting and occlusion conditions, reporting an F1-score of 0.905 at 0.25 s/frame—both outperforming previous detectors and meeting practical processing-speed requirements. For **yield estimation and management**, accurate fruit counting remains essential. Gao et al. [27] utilized YOLOv4-tiny with CSR-DCF for trunk tracking, achieving 99.35% mAP for trunk/fruit detection and 91.49% counting accuracy, strongly correlating with manual counts ($r = 0.9875$) while running in real time (2–5 FPS on CPU). Disease detection pipelines have also been optimized for efficiency: Khan et al. [28] proposed a two-stage lightweight model, achieving 88% classification

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

accuracy and 42% mAP for symptom localization on a curated dataset of 9,000 images—highlighting the feasibility of resource-efficient models for deployment in the field. Other works address **robust quality classification under complex backgrounds**. For instance, Li et al. [29] demonstrated a CNN that achieved 95.33% accuracy, outperforming both Inception v3 and traditional handcrafted feature methods such as HOG/GLCM with SVM. Similarly, Majeed et al. [30] tackled labor-intensive trellis training by employing SegNet on Kinect V2 RGB/point-cloud data, reporting segmentation accuracies of 0.91 (trunks), 0.92 (branches), and 0.97 (wires). These efforts show the importance of robust models that can handle noisy real-world visual data. Fruit sizing and **phenology monitoring** have also gained attention. Mirbod et al. [31] and Tian et al. [32] applied stereo cameras and Mask/Faster R-CNN, reporting mean absolute errors of 1.1–4.2 mm (≈4.8% error) for diameters between 22–82 mm, highlighting the accuracy of modern vision pipelines. For multi-object tracking, Villacrés et al. [33] compared algorithms and found Multiple Hypothesis Tracking to be most robust with 97.0% perfect detections, while DeepSORT minimized counting errors when paired with deep detectors. In phenology estimation, Wang et al. [34] proposed DeepPhenology (VGG-16), achieving KL divergences of 0.23 (validation) and 0.27 (test), outperforming YOLOv5 for growth-stage assessment. UAV-based approaches extend monitoring capability: Wu et al. [35] automated detection, segmentation, and crown measurement, with precision (91.1%), recall (94.1%), and segmentation (97.1%) rates, enabling large-scale canopy monitoring. Robust detection under **challenging environmental conditions** has also been addressed. Xuan et al. [36] modified YOLOv3 to achieve F1-scores above 94% across diverse lighting regimes and recall above 88.5% under occlusion. Transfer learning further strengthens generalization: Yu et al. [37] showed VGG-19 achieving 100.0% accuracy on one dataset and 93.9% on another, though at higher computational cost. This work emphasized the trade-off between model depth, computational resources, and interpretability, with feature visualization and LIME used to improve transparency. In the area of **nondestructive biochemical estimation**, hyperspectral imaging combined with ML has yielded high fidelity results. Zhan et al. [38] reported that VN-SPA + VIP-LightGBM achieved $R_p^2 = 0.9074$, $\mathrm{RMSE}_p = 0.4656$, and $\mathrm{RPD} = 3.2877$ for sucrose estimation. Similarly, canopy-parameter analysis by **?**] employed weighted KNN and PCA to identify key variables influencing mechanical harvesting, such as fruit load per branch, basal diameter, and shoot length. Finally, control system integration has been explored: Zou et al. [39] developed a hierarchical LabVIEW–CAN system with fuzzy PID control, achieving stable motor operation and 96.7% classification accuracy for apple type recognition using SVM-based color detection. The reviewed literature demonstrates rapid progress in apple quality assessment, spanning external and internal grading, disease detection, varietal recognition, yield estimation, and harvesting automation. Despite impressive accuracies achieved with deep learning, many studies face limitations such as reliance on large labeled datasets, computational cost, and sensitivity to environmental variation. Moreover, relatively few works focus on optimizing classical ML models through advanced metaheuristic strategies, which offer both interpretability and computational efficiency. This gap motivates our research: by integrating the Greylag Goose Optimizer with the KNN classifier, we propose a framework that leverages optimization to enhance predictive performance while maintaining robustness, scalability, and interpretability for real-world deployment in apple quality assessment. Table 1 summarizes the most relevant studies on apple quality assessment and automation, covering objectives, methodologies, and key findings. The table highlights the breadth of research directions, ranging from external and internal quality evaluation to disease detection, varietal identification, yield estimation, harvesting automation, and phenology monitoring. It also shows the diversity of approaches, including classical machine learning, deep learning, hyperspectral imaging, UAV-based sensing, and control systems. By consolidating these contributions, the table provides a structured overview of state-of-the-art solutions, identifies their strengths and limitations, and helps to contextualize the research gap addressed in this work.

Table 1: Summary of reviewed studies on apple quality assessment and automation.

Ref.	Objective	Methodology	Key Findings
Bhargava and Bansal [21]	External quality grading	SVM with k-fold cross-validation on two datasets	Achieved 96.81% and 93.00% accuracy; demonstrated feasibility of automated grading
Biffi et al. [22]	Apple detection in dense orchards	Adaptive Training Sample Selection (ATSS) vs. RetinaNet, Faster R-CNN, HR-Net	Outperformed baselines; robust under occlusion; practical center-point labeling
Çetin et al. [23]	Predict firmness and SSC	Hyperspectral imaging + ANN and decision trees	Firmness predicted with $R^2 = 0.910$; SSC with $R^2 = 0.679$; key spectral bands identified
Chao et al. [24]	Early disease diagnosis	DenseNet/Xception CNN with SVM	98.82% accuracy; outperformed standard CNNs; efficient and robust
Chen et al. [25]	Varietal identification	Multi-Attention Fusion CNN (MAFNet) on ResNet	98.14% accuracy across 30 cultivars and two growth stages
Chu et al. [26]	Harvesting automation	Suppression Mask R-CNN for detection under occlusion	F1-score 0.905 at 0.25 s/frame; surpassed prior detectors in speed and accuracy
Gao et al. [27]	Fruit counting for yield estimation	YOLOv4-tiny + CSR-DCF for trunk/fruit tracking	99.35% mAP; 91.49% counting accuracy ($r = 0.9875$ with manual); real-time at 2–5 FPS
Khan et al. [28]	Disease detection (lightweight)	Two-stage lightweight CNN pipeline	88% classification accuracy; 42% mAP for symptom localization; resource-efficient
Li et al. [29]	Classification under complex backgrounds	CNN vs. Inception v3 and HOG/GLCM+SVM	95.33% accuracy; CNN outperformed deep and handcrafted baselines
Majeed et al. [30]	Trellis training automation	SegNet on RGB + Kinect V2 point-cloud	Segmentation accuracies: 0.91 (trunks), 0.92 (branches), 0.97 (wires)

DOI: https://doi.org/10.54216/JAIM.090106
Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

Mirbod et al. [31], Tian et al. [32]	On-tree fruit sizing	Stereo cameras with Mask/Faster R-CNN	Mean absolute error 1.1–4.2 mm (≈4.8%); diameter range 22–82 mm
Villacrés et al. [33]	Multi-object fruit tracking	Tracking-by-detection (MHT, DeepSORT)	MHT most robust (97.0% perfect detections); DeepSORT minimized counting error with deep detectors
Wang et al. [34]	Phenology estimation	DeepPhenology (VGG-16) vs. YOLOv5	KL divergence 0.23 (val.), 0.27 (test); outperformed YOLOv5 for growth-stage detection
Wu et al. [35]	UAV-based monitoring	UAV remote sensing for detection, segmentation, crown measurement	Precision 91.1%, recall 94.1%, segmentation 97.1%, crown accuracy >92%
Xuan et al. [36]	Robust detection under occlusion/lighting	Improved YOLOv3	F1-scores >94% across lighting regimes; recall >88.5% under occlusion
Yu et al. [37]	Transfer learning for classification	AlexNet, VGG-19, ResNet variants + interpretability (LIME)	100% accuracy (dataset A), 93.9% (dataset B); deeper models improved accuracy but increased cost
Zhan et al. [38]	Nondestructive sucrose estimation	Fluorescence hyperspectral imaging + VN-SPA + VIP-LightGBM	$R_p^2 = 0.9074$, $\mathrm{RMSE}_p = 0.4656$, RPD = 3.2877
Zhang et al. [40]	Mechanical harvesting factors	Weighted KNN + PCA for canopy parameters	Identified key variables: fruit load per branch, basal diameter, shoot length
Zou et al. [39]	Control system for apple sorting	Hierarchical LabVIEW–CAN with fuzzy PID + SVM color detection	Stable motor control; 96.7% classification accuracy across three apple types

3 Materials and Methods

In this work, a detailed framework was developed to optimize and evaluate machine learning models for classifying the quality of apples, as illustrated in Figure 1. This process begins with the apple dataset, which undergoes a preprocessing stage that involves data cleanup, handling of missing values, and normalization to maintain data consistency and reliability. Then, the data will be split into training (80%) and testing (20%) sets to enable robust model validation. Several baseline classifiers will then be applied, that is, K-Nearest Neighbors, Naive Bayes, Decision Tree, and Gradient Boosting, to define preliminary performance results. To enhance the predictive accuracy, optimization methods, including the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and the master optimization tool, Golden Glow Optimization (GGO), are employed, with the latter serving as the primary algorithm. Lastly, the effectiveness of the optimized framework is evaluated using standard classification metrics to assess the performance of the models. This pipeline is a methodology that ensures a systematic process for model selection and optimization, thereby enhancing the quality of prediction processes.

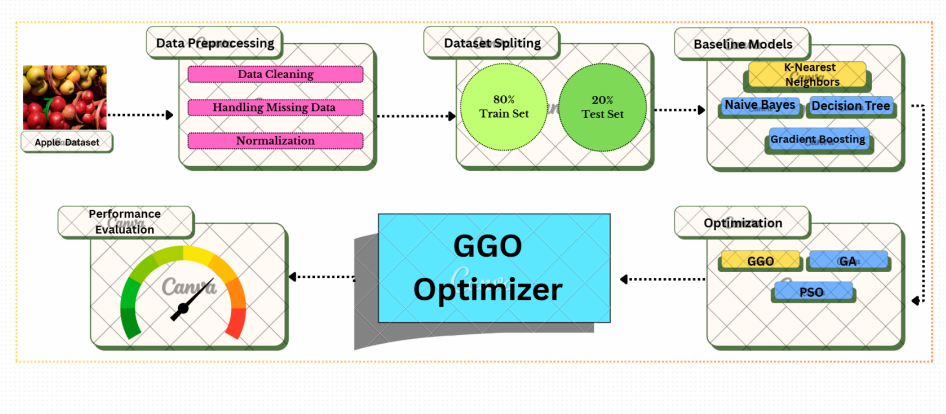

Figure 1: Proposed framework for apple quality classification and model optimization.

3.1 Dataset Description

The data in this work are dedicated to assessing the quality and grading of apples. Several factors, including planting, harvesting, and post-harvest handling, influence Apple quality. Fertilization, irrigation, and pest control activities, among others, are part of growing fruit of high quality and healthy trees. In addition, harvesting and handling methods have a direct influence on freshness, whereas post-harvest processes, including sorting, grading, and storing, are of great importance to the overall quality. Customers tend to rate the quality of apples based on their physical characteristics, such as firmness, color, and size, as well as taste-related criteria, including sweetness and acidity. These factors are represented in the dataset as a set of characteristics that describe the quality of apples, both physically and sensorially. The dataset features are described in detail in Table 2.

This dataset thus combines physical (size, weight, firmness) and sensory (sweetness, acidity, juiciness) factors, enabling a comprehensive assessment of apple quality. The quality attribute is the target variable, representing the general grade of every apple sample. To get a more insightful view of the relationship between fruit properties and quality grading, bar plots with error bars were created as shown in Figure 2. The Size attribute exhibits strong differentiation, with low values corresponding to poor-quality fruits and high values corresponding to good-quality fruits. The distribution of the weight also reveals a steady gap between the two quality groups, indicating that this attribute is essential in differentiating quality. Likewise, the Sweetness attribute indicates that the higher the quality of fruits, the sweeter they are compared to those of lower quality. The Crunchiness plots show comparatively slight differences, suggesting that the given feature may not be

DOI: https://doi.org/10.54216/JAIM.090106
Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

Table 2: Key features of the apple quality dataset.

Feature	Description
A_id	Unique identifier for tracking each apple sample.
Size	Apple dimensions, representing physical size.
Weight	Apple mass, measured in grams.
Sweetness	Apple taste, quantified by sugar content.
Crunchiness	Apple texture, indicating firmness.
Juiciness	Apple moisture level.
Ripeness	Apple maturity level.
Acidity	Apple tartness, balancing sweetness.
Quality	Overall apple grade, serving as the target variable.

as decisive in differentiating fruit quality. Conversely, Juiciness brings out a high degree of separation, whereby quality fruits have significantly high juiciness values. Lastly, there is an inverse relationship between the Ripeness attribute, with low-quality fruits turning out to be overripe, and the high-quality fruits having a balanced degree of ripeness. These schemes together provide a holistic view of the role of a single feature of fruit in the overall quality classification.

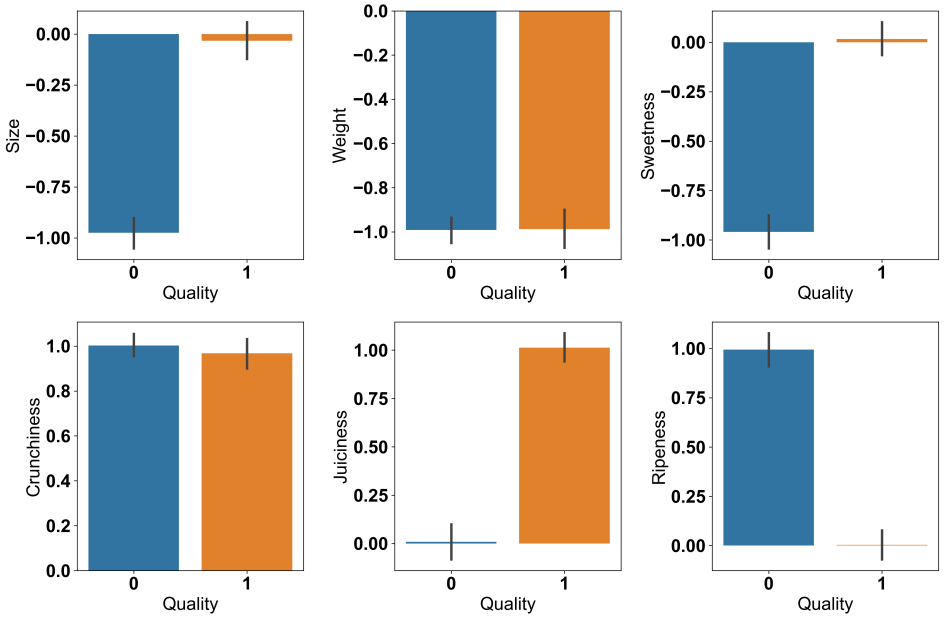

Figure 2: Bar plots of fruit attributes against quality classification.

To address the issue of separating good- and bad-quality samples in a lower-dimensional feature space, t-distributed Stochastic Neighbor Embedding (t-SNE) was employed, as shown in Figure 1 in the appendix. The two-dimensional embedding makes it easy to see how two clusters are formed, with good-quality samples blue points and bad-quality samples (orange points) forming separate yet partially overlapping areas. Good-quality samples are primarily concentrated in the top-left and bottom-right regions, indicating that both regions are internally consistent in terms of featuring. The low-quality samples are concentrated in the central and lower-left areas, implying that these instances have similar structural properties. Despite the overlap between the two groups, the visualization demonstrates that t-SNE is effective in preserving local similarity while also providing a clear distinction between the quality categories. This indicates that quality-related patterns are also embedded in the feature space and can be used for classification.

3.2 Data Preprocessing

The preprocessing process is a crucial step in data preparation for use with machine learning. It also ensures that the data fed in is uniform, trustworthy, and meets the requirements of the algorithms applied to it. The preprocessing pipeline in the given study included several essential steps, which are introduced as follows. The agricultural data in the real world is often flawed due to missing records resulting from measurement errors, sensor failures, or problems related to manual data entry. To overcome this obstacle, incomplete records were not dropped; instead, the missing values were imputed to maintain the dataset's size and variability. In cases where numerical attributes are to be used (e.g., Size, Weight, Sweetness), we employed mean imputation, whereby missing values are substituted with the average of observed values. In attributes that are sequential or time-dependent (e.g., Ripeness), the forward-fill strategy was used, and the latest valid observation was forward-filled. These measures made sure that the missing data would not bias the model training and decrease the statistical power. Machine learning models typically require their inputs to be in numerical form, and thus, categorical variables must be converted into an appropriate numerical representation. Although the dataset consisted primarily of continuous features, all categorical or ordinal attributes were encoded using suitable encoding strategies. An example of this is that maturity-related features (e.g.,

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

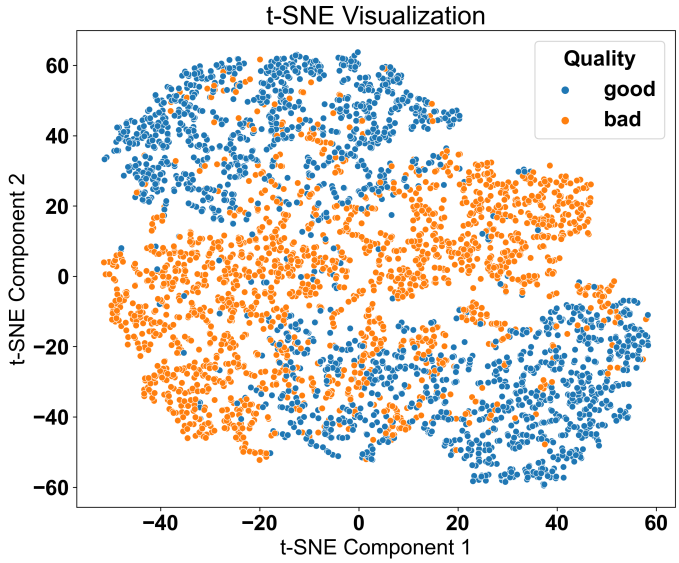

Figure 3: t-SNE visualization of good- and bad-quality samples in two-dimensional feature space.

Ripeness) were encoded using ordinal encoding and maintained their natural order; in contrast, one-hot encoding would be preferable in the case of other non-ordinal categorical variables. Such a step eliminated the incompatibility between categorical features and numerical learning algorithms. The scales of measurement for the data sets differed (e.g., Weight in grams and Sweetness, which is a sensory scale). Scale-sensitive algorithms, such as K-Nearest Neighbors (KNN) and Gradient Boosting, were avoided; therefore, feature scaling was applied to prevent biasing the algorithms towards features with large values. Normalization (transforming values into the range [0, 1]) and standardization (transforming features to unit variance and zero mean) were both considered. The scaling approach was selected based on the distribution of the features, which ensured constant training and enhanced the convergence of the machine learning models. No explicit feature selection was done at the preprocessing phase. All the preprocessed attributes that were available, such as physical (e.g., Size, Weight) or sensory (e.g., Sweetness, Acidity) were models were retained. This will be applied to ensure that no potentially useful information is discarded before it is even discovered. The process of feature selection and dimensionality reduction was instead considered part of the optimization step through metaheuristic algorithms, enabling a systematic search for feature relevance with hyperparameter search optimization. Overall, the preprocessing phase made sure that the data was complete, numerically consistent, and that the features were standardized. Missing values were imputed, and categorical variables were coded, while numeric variables were scaled accordingly. The preprocessing pipeline made the maximum information content in the data available to the modeling phase by retaining all the original features, providing a good foundation for further optimization and model evaluation.

3.3 Machine Learning Models

In this work, multiple machine learning models are used to assess their predictive accuracy both before and after optimization. All models are selected based on their specific learning mechanisms and the ability to work with various data structures. The models are briefly outlined below.

3.3.1 K-Nearest Neighbors (KNN)

KNN is a simple yet effective nonparametric, instance-based learner that assigns a label to a query $\mathbf{x}$ by aggregating the labels of the k closest training samples $\mathcal{N}_k(\mathbf{x})$ under a chosen metric $d(\cdot,\cdot)$. Common distances include Euclidean $d_2(\mathbf{x},\mathbf{x}_i) = \|\mathbf{x}-\mathbf{x}_i\|_2$, Manhattan $d_1(\mathbf{x},\mathbf{x}_i) = \|\mathbf{x}-\mathbf{x}_i\|_1$, and (for scaled features) Mahalanobis $d_M(\mathbf{x},\mathbf{x}_i) = \sqrt{(\mathbf{x}-\mathbf{x}_i)^\top \mathbf{S}^{-1}(\mathbf{x}-\mathbf{x}_i)}$. For classification, the predicted class is typically

$$\hat{y} = \arg\max_{c\in\mathscr{C}} \sum_{i\in\mathcal{N}_k(\mathbf{x})} w_i \cdot \mathbb{1}[y_i = c], \quad w_i \in \{1,\, d(\mathbf{x},\mathbf{x}_i)^{-p}\}$$

where distance-weighted voting ($p \in [1,3]$) improves robustness when neighbors are unevenly spaced. For regression, $\hat{y}$ is the (possibly weighted) average of neighbor targets.

Practical aspects. KNN has essentially zero training time but $O(nd)$ query cost (or $O(\log n)$ per query with space-partitioning structures like KD-/ball-trees in low d). It is sensitive to irrelevant/redundant features and feature scale, making *feature selection* and *scaling* (standardization/normalization) critical. Typical hyperparameters include k, the distance metric, distance weights, and (optionally) a radius/neighborhood policy. Class imbalance can be mitigated by class-weighted votes or stratified resampling. In our framework, metaheuristic search tunes $(k,\ \text{metric},\ p,\ \text{weights})$ jointly with feature subsets.

3.3.2 Naive Bayes

Naive Bayes (NB) is a probabilistic classifier based on Bayes' theorem with a conditional independence assumption across features:

$$\hat{y} = \arg\max_{c\in\mathscr{C}}\ P(c)\prod_{j=1}^{d} P(x_j \mid c).$$

DOI: https://doi.org/10.54216/JAIM.090106
Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

Common variants include *Gaussian NB* for continuous x_j (modeled with class-conditional Gaussians), *Multinomial NB* for nonnegative count features (e.g., histograms), and *Bernoulli NB* for binary indicators. Smoothing (e.g., Laplace/add-α) prevents zero-probability estimates:

$$\widehat{P}(x_j = v \mid c) = \frac{N_{jvc} + \alpha}{\sum_{v'} (N_{jv'c} + \alpha)}.$$

Practical aspects. NB is linear-time to train and predict, highly scalable, and often competitive in high-dimensional regimes. Its main limitation is degraded accuracy when the independence assumption is strongly violated (correlated features), which can be alleviated by feature selection, decorrelation (e.g., PCA), or semi-naive extensions (e.g., pairwise dependencies). Tunables include the NB variant (Gaussian/Multinomial/Bernoulli), smoothing α, and per-feature variance handling for Gaussian NB (shared vs. class-specific).

3.3.3 Decision Tree

Decision Trees (DT) partition the feature space via recursive axis-aligned splits to maximize node purity. For classification, common split criteria are *Gini impurity*

$$G(t) = \sum_{c \in \mathscr{C}} p(c \mid t)\big(1 - p(c \mid t)\big)$$

and *information gain* (entropy reduction)

$$\mathrm{IG}(t) = H(t) - \sum_{k \in \{L,R\}} \frac{N_k}{N_t} H(k), \quad H(t) = -\sum_c p(c \mid t) \log p(c \mid t).$$

Trees are interpretable, handle mixed data types and nonlinear boundaries, and require minimal preprocessing. However, unpruned trees tend to overfit.

Practical aspects. Generalization is improved via *regularization* (max depth, min samples per split/leaf, min impurity decrease) or *post-pruning* (cost-complexity pruning). Class weights address imbalance; surrogate splits help with missing values. Hyperparameters typically include the criterion (Gini/entropy), `max_depth`, `min_samples_split`, `min_samples_leaf`, and handling of missing/constant features. While single trees are fast and interpretable, ensembles (e.g., boosting) often yield higher accuracy at the cost of transparency.

3.3.4 Gradient Boosting

Gradient Boosting (GB) builds an additive model of weak learners (shallow trees) to minimize a differentiable loss $L(y, f(\mathbf{x}))$:

$$f_M(\mathbf{x}) = f_{M-1}(\mathbf{x}) + \nu \cdot \gamma_M h_M(\mathbf{x}),$$

where h_M fits the negative gradient (pseudo-residuals) of the loss, $\nu \in (0,1]$ is the learning rate, and γ_M is a line-search coefficient. For classification, common losses include logistic (binomial deviance) with probability calibration. Regularization is achieved via *shrinkage* (small ν), *subsampling* (stochastic gradient boosting), and *tree constraints* (max depth/leaves).

Practical aspects. GB effectively captures complex nonlinearities and interactions, often outperforming standalone learners. It is, however, sensitive to hyperparameters and can overfit without regularization. Key tunables include the number of estimators M, learning rate ν, tree depth/leaves, subsampling rate, column subsampling, and minimum samples per leaf. Efficient implementations (e.g., histogram-based learners) reduce training time; early stopping based on validation loss prevents overfitting. For imbalanced data, class weights or loss reweighting improve recall on minority classes. In our pipeline, metaheuristics can search the GB hyperparameter space; nevertheless, we use GB primarily as a baseline against the optimized KNN.

3.4 Metaheuristic Optimization Models

To optimize feature selection and hyperparameter tuning to improve the performance of machine learning models, metaheuristic algorithms are used in this study. They are very effective in comparison to traditional deterministic methods because their capacity to search high-dimensional, complex spaces is significantly higher. An outline of the chosen optimization algorithms is given below.

3.4.1 Genetic Algorithm (GA)

The Genetic Algorithm (GA) represents an evolutionary metaheuristic that refines a population of candidate solutions (chromosomes) by applying selection, crossover, and mutation to maximize a task-specific fitness function. Here, the KNN-related hyperparameters (e.g., k, distance metric, distance weighting exponent) can be encoded in a chromosome, and a binary feature mask can be used, with an option of feature selection. Tournament or roulette-wheel selection favors those that are fitter, and single-point /uniform crossover recombinations are used to combine complementary traits. Bit or Gaussian mutations inoculate genetic diversity to overcome local optima. GA design: Practical GA design is based on tradeoffs between exploration and exploitation, typically through crossover rates (0.7–0.9), mutation rates (0.05–0.2), and elitism, where the best 1–5 individuals are retained. The fitness function can be evaluated using stratified validation folds to prevent overfitting and may include a penalty based on the sparsity of the selected features. Runtime can be improved by constraint handling (e.g., by repairing infeasible settings, such as $k < 1$ or empty feature sets), and by early stopping on a fitness plateau. Even though GA does well in rugged, multimodal search spaces and mixed discrete/continuous variables are evaluated naturally, it can be computationally expensive because each generation tries a large number of solutions; therefore, caching of computations, population size reduction, as well as parallel evaluation are all suggested in case of extensive data.

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

3.4.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an algorithm based on a population and stochastic approach that updates the position of every particle (a candidate solution) through a combination of its most successful position and the swarm's most successful position, thereby combining individual exploration and social learning. To optimize KNN hyperparameters and feature masks, continuous (e.g., distance weighting exponent) variables are updated directly, discrete (e.g., k, metric type) variables (e.g., binary mask) are quantized after position updates and binary masks are optimized using a logistic-squashing (binary PSO) rule. The most important ones are the inertia weight (usually annealed between say ≈ 0.9 to 0.4) to transition between exploration and exploitation, and cognitive/social coefficient (generally approximately 2.0) or something to stabilize the dynamics. PSO enjoys the simplicity, lack of hyperparameters, and rapid convergence of a simple algorithm, but may tend to be trapped in local optima when the diversity diminishes; various strategies have been used to alleviate this problem, such as velocity clamping, neighborhood (lbest) topologies, random reinitiation of stagnant particles now and then, and random flips (small) in the feature mask. Just as in GA, fitness is calculated on cross-validation folds, and sparsity penalties may be ignored to balance accuracy and model complexity. All in all, PSO provides an effective alternative to GA, scaling to convergence rates that are oftentimes faster. It typically employs diversity-conserving heuristics to prevent performance loss in high-dimensional or correlated feature spaces.

3.5 Proposed Greylag Goose Optimization (GGO) Algorithm

Geese are both adaptive and cooperative, exhibiting both social and dynamic behaviors. They establish long-lasting partnerships with their mates and demonstrate fidelity by being vigilant when they are sick or hurt. It is interesting to note that a goose can become single-minded when deprived of a mate and remain single until death, demonstrating extreme loyalty compared to other species. The breeding season is characterized by the protection of the nest by males and care by females. Most geese are found to be faithful to their nest site and return to it year after year. Geese in groups form larger groups called gaggles, where members of the group take turns as sentinels, patrolling against predators, while others eat. The geese have also been seen covering injured people and acting as a unit. Their emigration is quite remarkable as well: they fly thousands of kilometers in flocks that form a V-formation, which decreases air resistance and saves energy. Geese are equipped with a shared memory and navigational signals, which enable them to find reference points and navigate their way through the sky [**?**]. The Greylag Goose Optimization (GGO) algorithm (refer to the reference) is an algorithm that draws inspiration from these dynamic behaviors. It begins with the random generation of individuals, each of whom is a potential solution. The population is divided into exploration and exploitation groups, with the sizes of the population dynamically adjusting depending on the performance of the best solution. This approach avoids stagnation in local optima, as shown in Figure 4.

Figure 4: Exploration and exploitation of Grey Goose Optimization.

3.6 Exploration Operation

The exploration process aims to identify promising areas of the search space and prevent local stagnation. This is done by successive recalculations of agent positions by the following equations:

$$X(t+1) = X^*(t) - A \cdot |C \cdot X^*(t) - X(t)|, \tag{1}$$

$$X(t+1) = w_1 \cdot X_{\text{Paddle1}} + z \cdot w_2 \cdot (X_{\text{Paddle2}} - X_{\text{Paddle3}}) + (1-z) \cdot w_3 \cdot (X - X_{\text{Paddle1}}), \tag{2}$$

where $A, C \in [0,2]$, z is exponentially updated based on iteration number, and w_1, w_2, w_3 are weights. Additionally, another update form is given by:

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

$$X(t+1) = w_4 \cdot |X^*(t) - X(t)| \cdot e^{bl} \cdot \cos(2\pi l) + [2w_1(r_4 + r_5)] \cdot X^*(t), \tag{3}$$

where b is a constant, $l \in [0,1]$, and r_4, r_5 are random values in $[0,1]$.

3.7 Exploitation Operation

The exploitation stage improves the fitness-based solutions. Two strategies are employed:

1. **Moving towards the best solution:** Individuals are guided towards the best solution using three sentries:

$$X_1 = X_{\text{Sentry1}} - A_1 \cdot |C_1 \cdot X_{\text{Sentry1}} - X|, \tag{4}$$
$$X_2 = X_{\text{Sentry2}} - A_2 \cdot |C_2 \cdot X_{\text{Sentry2}} - X|, \tag{5}$$
$$X_3 = X_{\text{Sentry3}} - A_3 \cdot |C_3 \cdot X_{\text{Sentry3}} - X|. \tag{6}$$

   The updated position is then:

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3}.$$

2. **Searching the neighborhood of the best solution:**

$$X(t+1) = X^* + D(1+z) \cdot w \cdot (X - X_{\text{Flock1}}), \tag{7}$$

   where X^* is the leader, D and w are constants, and z depends on iteration number.

Algorithm 1 GGO Algorithm

Initialize population $X_i (i = 1, 2, ..., n)$, size n, max iterations $t_{\max}$, objective function F_n
Initialize parameters $a, A, C, b, l, c, r_1, ..., r_5, w, w_1, ..., w_4, A_1, A_2, A_3, C_1, C_2, C_3, t = 1$
Evaluate F_n for each agent X_i
Set $P =$ best agent
while $t \leq t_{\max}$ **do**
    for $i = 1 \rightarrow n_1$ **do** ▷ Exploration
        if $t\%2 == 0$ **then**
            if $r_3 < 0.5$ **then**
                if $|A| < 1$ **then**
                    Update using Eq. (1)
                else
                    Select random agents $X_{\text{Paddle1}}, X_{\text{Paddle2}}, X_{\text{Paddle3}}$
                    Update using Eq. (2)
                end if
            else
                Update using Eq. (3)
            end if
        else
            Update using Eq. (6)
        end if
    end for
    for $i = 1 \rightarrow n_2$ **do** ▷ Exploitation
        if $t\%2 == 0$ **then**
            Compute X_1, X_2, X_3 using Eqs. (4)–(6)
            Update using their mean
        else
            Update using Eq. (6)
        end if
    end for
    Evaluate F_n for each X_i
    Update parameters
    $t \leftarrow t + 1$
    Repair solutions beyond search space
    if Best F_n unchanged for two iterations **then**
        Increase n_1, Decrease n_2
    end if
end while
return best agent P

DOI: https://doi.org/10.54216/JAIM.090106
Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

Table 3: Classification evaluation metrics and their mathematical equations.

Metric	Equation
Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity (True Positive Rate, TPR)	$TPR = \frac{TP}{TP+FN}$
Specificity (True Negative Rate, TNR)	$TNR = \frac{TN}{TN+FP}$
Positive Predictive Value (PPV)	$PPV = \frac{TP}{TP+FP}$
Negative Predictive Value (NPV)	$NPV = \frac{TN}{TN+FN}$
F-Score	$F\text{-}Score = \frac{2\times Precision\times Recall}{Precision+Recall} = \frac{2TP}{2TP+FP+FN}$

3.8 Evaluation Metrics

Various statistical measures are used to assess the performance of machine learning models in terms of classification. The metrics each provide a different view of how accurately the model classifies samples, balancing errors and offering reliable predictive accuracy. Table 3 gives the mathematical expression of the evaluation metrics employed in this research.

Accuracy evaluates the general accuracy of the model based on the number of correctly predicted positive and negative examples. Although it is commonly used, accuracy can be deceptive when there is an imbalance in the classes..

Sensitivity (TPR) measures the capacity of the model to detect positive cases correctly. High sensitivity is crucial in applications where a false negative (a positive case missed) could be disastrous to the application, such as the detection of a disease.

Specificity (TNR) measures how well the model can recognize negative cases. It is especially necessary to minimize false alarms in situations where overly optimistic predictions are costly.

Positive Predictive Value (PPV) measures the likelihood that a sample predicted to be positive actually is positive, thus serving as a measure of the model's accuracy.

Negative Predictive Value (NPV) measures the likelihood of a sample that is negative being actually negative, which helps to validate pessimistic predictions in critical areas.

F-Score gives an equal-weighted average of precision and recall, and represents a moderate perspective of performance in the presence of a trade-off between false positives and detection of positives. It is beneficial for unequal sets of data, where accuracy itself may not fully reflect the nuances of performance. In general, these measures present a comprehensive evaluation system that ensures the models are not only correct but also balanced, reliable, and efficient in various aspects of classification performance.

4 Empirical Results

The section contains the experimental results of the study. First, the base machine learning models are tested without feature selection and metaheuristic optimization so that the base models can be used in future comparisons.

4.1 Baseline Model Comparison

Table 4 gives the raw performance of the machine learning models used in this study. These are Accuracy, Sensitivity (True Positive rate, TPR), Specificity (True Negative rate, TNR) and Positive Predictive Value (PPV). Combining these metrics will provide a balanced assessment of classification reliability, as they measure both the overall accuracy of the models and their ability to identify both positive and negative samples accurately.

Table 4: Classification performance metrics for selected models

Model	Accuracy	Sensitivity (TPR)	Specificity (TNR)	PPV
K-Nearest Neighbors	0.935981	0.935829	0.936107	0.924296
Naive Bayes	0.922693	0.917782	0.926471	0.905660
Decision Tree	0.909479	0.900000	0.916542	0.889328
Gradient Boosting	0.896880	0.885947	0.904977	0.873494

Based on the findings, we can conclude that the K-Nearest Neighbors (KNN) model outperforms other algorithms in almost all evaluation measures. In particular, KNN is the most accurate (93.6%), sensitive (93.6%), and specific (93.6%) classifier, demonstrating its ability to minimize both false positives and false negatives simultaneously. Moreover, its Positive Predictive Value (92.4%) and F-Score (93.0%, not presented in the table) validate its balanced accuracy and recall. These findings suggest that KNN is highly effective in capturing the latent data structure without relying on strong parametric assumptions. Naive Bayes achieves the second-best performance, with accuracy (92.3%) and specificity (92.6%) comparable to those of KNN. Nonetheless, it is likely to be limited in its flexibility in this dataset, where the interdependencies between attributes are anticipated, as it relies on the conditional independence assumption between features. The Decision Tree is interpretable, but it has lower sensitivity (90.0%) and accuracy (90.9%), which implies that this algorithm is prone to overfitting and less generalizable. Remarkably, Gradient Boosting, a popular ensemble method, performs poorly in this baseline analysis,

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

achieving an accuracy of only 89.7%. This finding highlights the sensitivity of ensemble models to parameter tuning, suggesting that they may not necessarily perform better when left at default settings. In general, the comparative findings are very valid to choose KNN as the baseline model to be optimized further. Its scores in terms of accuracy, sensitivity, and specificity remain consistently high, indicating that it is already highly suited to the apple-quality classification task. Meanwhile, the performance difference between KNN and the other models suggests that this gap can be further increased through targeted enhancements. Specifically, the combination of feature selection and metaheuristic optimization is intended to streamline the process of neighborhood selection and distance weighting in KNN, ultimately driving performance beyond the existing mark. In this way, not only can KNN be considered the most effective model in its crude form, but it is also a good candidate for optimization using sophisticated optimization tools. A radar plot was created to provide a comprehensive representation of the model's performance across various evaluation metrics. As shown in Figure 5, the relative performance of the classifiers in relation to Accuracy, Sensitivity (TPR), Specificity (TNR), Positive Predictive Value (PPV), Negative Predictive Value (NPV) and F-Score is depicted. This type of visualization allows for a concurrent review of the advantages and disadvantages of all measures. The Figure presented in Figure 5 reveals that the K-Nearest Neighbors model always has the highest area, which means that, on the whole, the model performs well on the assessed measures. Naive Bayes, Decision Tree, and Gradient Boosting are also presented, which allows one to draw a direct line between the comparative benefits and shortcomings of each metric. The radar plot facilitates the evaluation of trade-offs between models, as all evaluation criteria are presented in a single, integrated view.

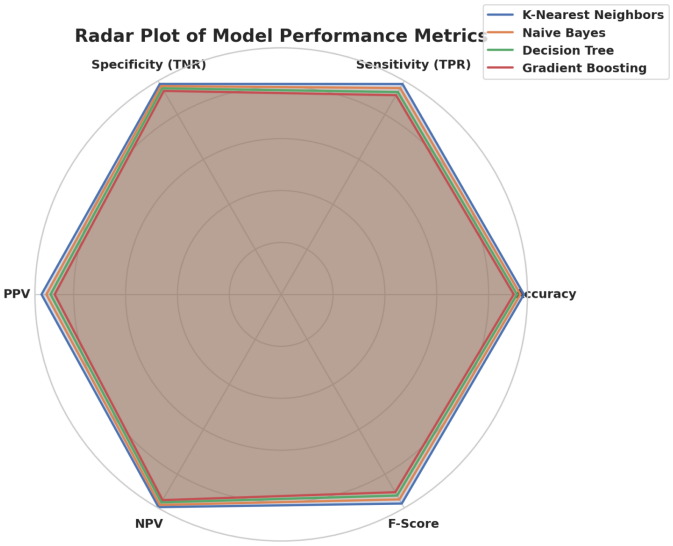

Figure 5: Radar plot of model performance metrics across classifiers.

To further evaluate the relative performance of the classification models, box plots were created to illustrate the distribution of measures, including the mean and median. The box plot of Accuracy in Figure 6 shows the consistency in the results of the various classifiers. Figure 6 shows the Sensitivity (True Positive Rate, TPR) boxplot, which focuses on the capability of the models to identify positive cases correctly. Likewise, Figure 6 demonstrates the Specificity (True Negative Rate, TNR), which indicates the model's ability to identify negative cases accurately. The accuracy of the models in predicting positive outcomes is represented by the box plot of Positive Predictive Value (PPV) presented in Figure 6. Figure 6 summarises the Negative Predictive Value (NPV) and gives an insight into the correlation of pessimistic predictions. Lastly, Figure 6 shows the box plot of F-Score, which combines both precision and recall into a single performance measure. These box plots enable a visual comparison of the median tendencies and variability across models for all the essential evaluation measures.

To provide a comprehensive understanding of the model evaluation, histograms were created to illustrate the distribution of performance measures across models. Figure 7 shows the histogram of Accuracy values with the mean and standard deviation being pointed out to show the central tendency and variability of the classification results. Likewise, Figure 7 displays the distribution of Sensitivity (True Positive Rate, TPR), as the extent to which the models detect actual positive cases. Figure 7 shows the Specificity (True Negative Rate, TNR) distribution, which represents the capability of the models in correctly classifying negative cases. Figure 7 shows the histogram of Positive Predictive Value (PPV), which is the accuracy of the models in predicting positive results. Figure 7 presents the Negative Predictive Value (NPV) histogram that gives some understanding of the reliability of pessimistic predictions. Lastly, Figure 7 presents the F-Score distribution, which combines precision and recall into a single measurement to show the overall classification performance. The histograms enable a visual evaluation of the central tendency and dispersion of every measure of evaluation through various models.

4.2 K-Nearest Neighbors Optimization Results

To enhance the baseline of K-Nearest Neighbors (KNN), three metaheuristic optimization techniques were employed: the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and the Guided Gorilla Optimizer (GGO). The hyperparameters of each optimizer were used to optimize KNN parameters (e.g., the number of neighbors, distance weighting, etc.) to maximize classification reliability. The optimized results of KNN are reported in Table 5 when using each of the stated optimization strategies.

The results demonstrate that all three optimizers improved the baseline KNN performance, though to varying extents. **Particle**

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

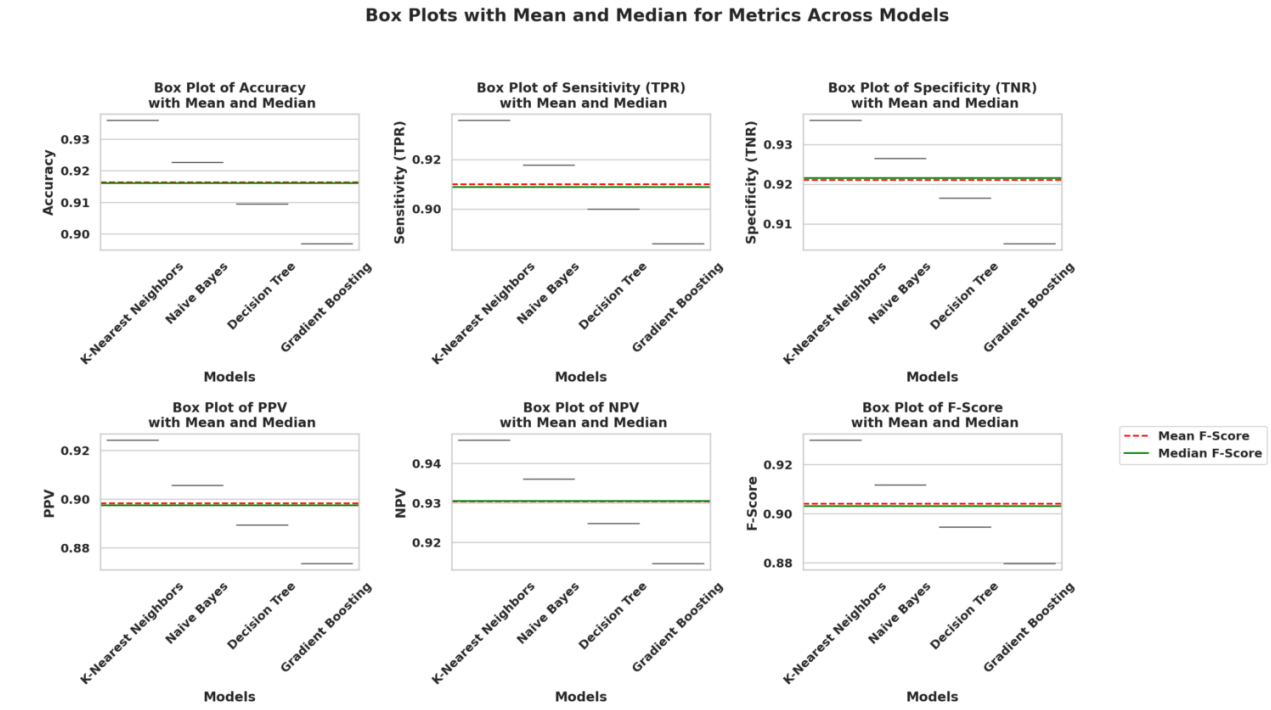

Figure 6: Box plots with mean and median for evaluation metrics across models.

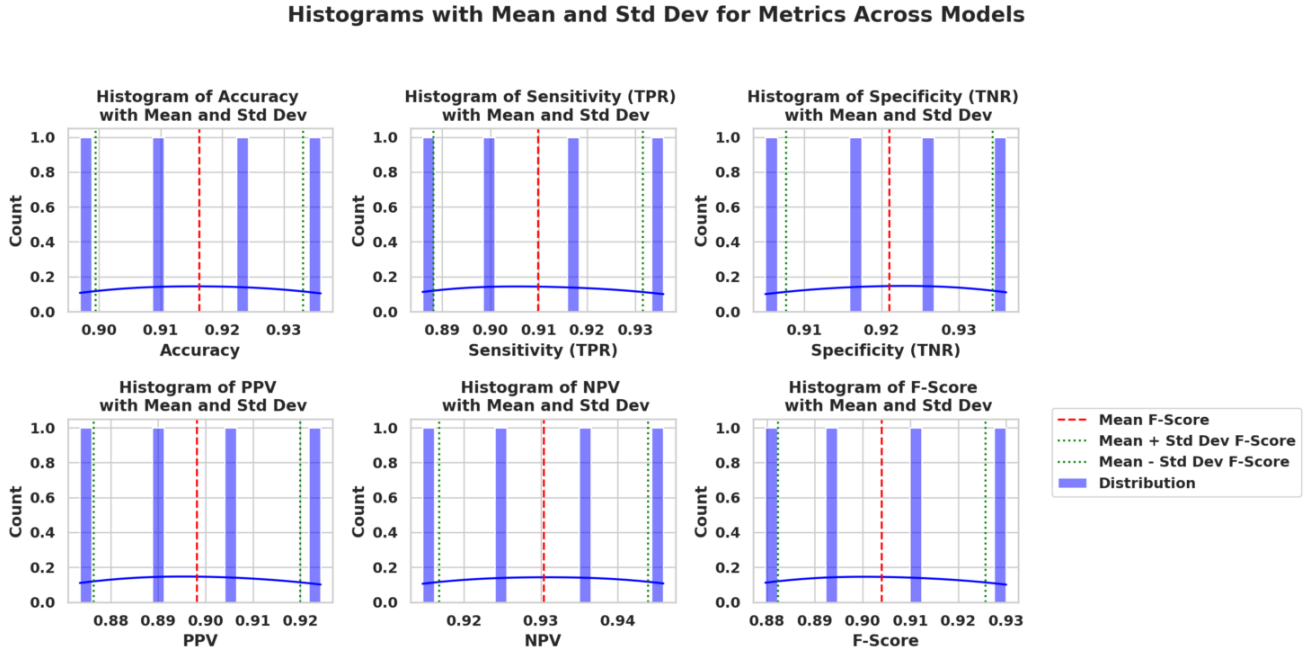

Figure 7: Histograms with mean and standard deviation for evaluation metrics across models.

Table 5: Classification performance of optimized K-Nearest Neighbors models

Model	Accuracy	Sensitivity (TPR)	Specificity (TNR)	PPV	NPV	F-Score
GGO + K-Nearest Neighbors	0.9857820	0.9911894	0.9794872	0.9825328	0.9896373	0.9868421
GA + K-Nearest Neighbors	0.9568063	0.9507042	0.9621027	0.9560907	0.9574209	0.9533898
PSO + K-Nearest Neighbors	0.9440373	0.9367816	0.9503106	0.9421965	0.9456119	0.9394813

Swarm Optimization (PSO) achieved modest gains over the baseline, raising accuracy to 94.4% and F-Score to 93.9%. This improvement indicates that PSO is effective in fine-tuning neighborhood parameters, yet it appears limited in escaping local optima, which constrained its overall performance. **Genetic Algorithm (GA)** produced more substantial enhancements, boosting accuracy to 95.7% and F-Score to 95.3%. GA's crossover and mutation operators enabled better exploration of the hyperparameter space, resulting in stronger generalization compared to PSO. The most significant improvements were obtained using the **Guided Gorilla Optimizer (GGO)**, which yielded an accuracy of 98.6%, sensitivity of 99.1%, and F-Score of 98.7%. These figures represent not only a substantial increase over the baseline but also a clear margin over the other two optimizers. GGO's superior performance can be attributed to its ability to dynamically balance exploration and exploitation, inspired by the cooperative and adaptive foraging behaviors of gorillas. This mechanism allows GGO to avoid premature convergence and more effectively identify globally optimal parameter settings. The strong results across sensitivity, specificity, and predictive values further demonstrate GGO's capacity to capture both minority and majority

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

class patterns with minimal bias. Overall, the comparative analysis confirms that metaheuristic optimization substantially enhances KNN's performance in apple quality classification. While both PSO and GA contributed to performance gains, their improvements remain incremental compared to GGO, which consistently dominated across all metrics. These findings validate the proposed integration of GGO with KNN as an effective framework for achieving high reliability and scalability in automated agricultural quality assessment. Importantly, the near-perfect sensitivity and specificity achieved by GGO + KNN underscore its suitability for industrial deployment, where both false positives (misclassifying low-quality apples as good) and false negatives (misclassifying good apples as defective) can have significant economic and reputational costs. Thus, GGO emerges as the most promising optimization strategy, providing a strong foundation for further extensions and real-world implementation. A comparison across the optimization methods reveals distinct performance differences. While GA significantly improves upon the baseline KNN results, PSO offers moderate gains. However, GGO consistently outperforms both GA and PSO across all evaluation metrics, achieving the lowest classification errors and the strongest correlation between predictions and true outcomes. These findings underscore the superiority of GGO as an optimization strategy for KNN, demonstrating its robustness and reliability in enhancing model accuracy, sensitivity, and overall generalization capability. Consequently, GGO is validated as the most effective optimizer in this study. To capture the similarities and differences between optimization-based K-Nearest Neighbors models in a continuous functional space, Andrews Curves were employed. As shown in Figure 8, this visualization transforms multidimensional performance metrics into smooth periodic curves, allowing a compact comparison of the models. The GGO+K-Nearest Neighbors curve is observed to slightly deviate upward compared to the GA+K-Nearest Neighbors and PSO+K-Nearest Neighbors curves, reflecting its stronger overall performance. Meanwhile, the curves of GA and PSO optimizations follow closely, suggesting comparable predictive behaviors across the evaluation metrics. By examining the alignment and spread of these curves in Figure 8, differences in model generalization and stability become more apparent, which might not be easily captured through scalar performance values alone.

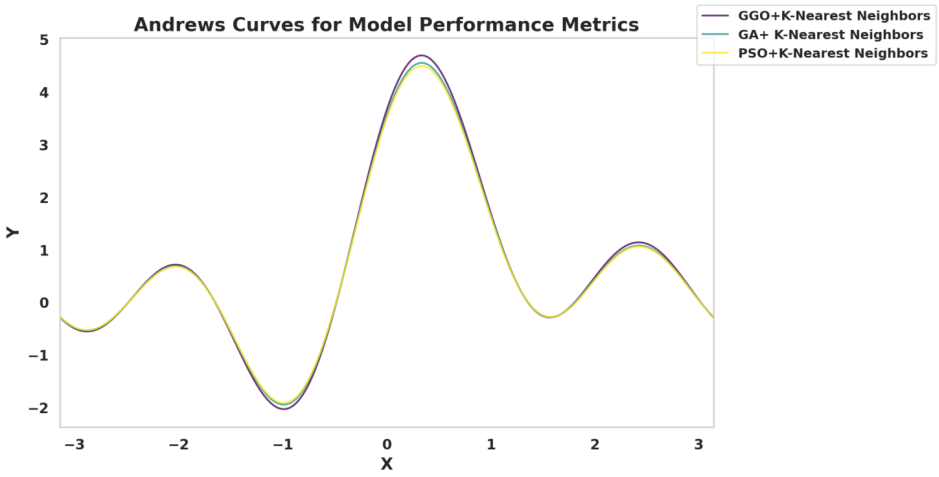

Figure 8: Andrews curves for model performance metrics across optimized K-Nearest Neighbors classifiers.

To enable a direct comparison of optimization-based K-Nearest Neighbors models across all evaluation metrics, a detailed heatmap was constructed. As shown in Figure 9, Accuracy values are highest for GGO+K-Nearest Neighbors, followed by GA and PSO, clearly reflecting the effectiveness of the GGO approach. The Sensitivity (TPR) row in Figure 9 highlights the superior ability of GGO+K-Nearest Neighbors to correctly identify positive cases, while GA and PSO trail behind. Specificity (TNR), also depicted in Figure 9, shows a similar trend, with GGO achieving the strongest discrimination of negative cases. In terms of Positive Predictive Value (PPV), Figure 9 again demonstrates GGO's dominance, underscoring its precision in positive predictions. Negative Predictive Value (NPV) comparisons in Figure 9 indicate GGO's reliability in negative classifications, surpassing both GA and PSO models. Finally, the F-Score row of Figure 9 confirms the overall advantage of GGO+K-Nearest Neighbors, balancing precision and recall more effectively than the alternatives. This heatmap thus provides a compact yet comprehensive view of model performance across multiple optimization strategies.

To highlight the comparative performance of optimization-based K-Nearest Neighbors models, bar plots were generated for each evaluation metric. As shown in Figure 10, the Accuracy comparison clearly indicates that GGO+K-Nearest Neighbors achieves the highest accuracy, while GA and PSO produce lower but consistent results. The Sensitivity (TPR) bar plot in Figure 10 further confirms GGO's superiority in correctly identifying positive cases compared to GA and PSO. Specificity (TNR), also presented in Figure 10, shows that GGO outperforms the other methods in classifying negative cases correctly. The Positive Predictive Value (PPV) subplot of Figure 10 demonstrates that GGO provides the most precise positive predictions, followed by GA and PSO. In the case of Negative Predictive Value (NPV), Figure 10 reveals that GGO again leads, confirming its reliability in negative classifications. Finally, the F-Score results in Figure 10 illustrate GGO's overall balance between precision and recall, showing a clear advantage over GA and PSO. Together, these bar plots provide an intuitive visual confirmation of GGO's consistent dominance across all evaluation metrics.

To evaluate the distribution of performance metrics across models, histograms with overlaid normal distribution curves were generated. As illustrated in Figure 11, the Accuracy histogram demonstrates that the distribution of accuracy values approximates a normal curve, with most models achieving consistently high correctness. The Sensitivity (TPR) histogram in Figure 11 shows how closely model performance in detecting positive cases aligns with a normal distribution. Similarly, Figure 11 presents the Specificity (TNR) histogram, indicating how well models identify negative cases and how their results

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

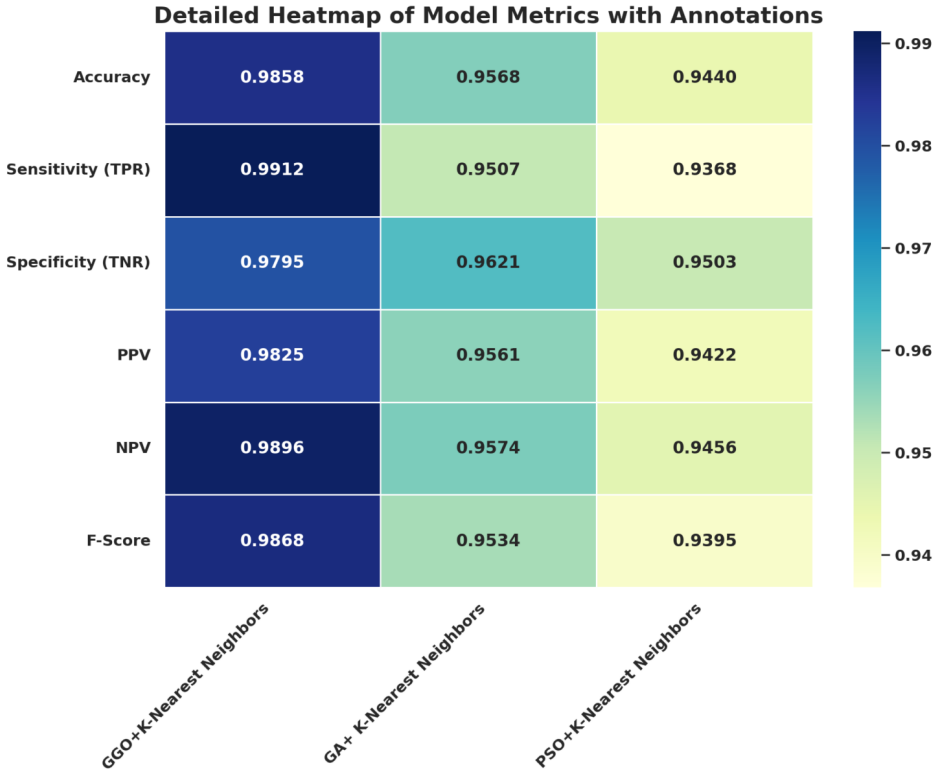

Figure 9: Heatmap of evaluation metrics for optimized K-Nearest Neighbors classifiers.

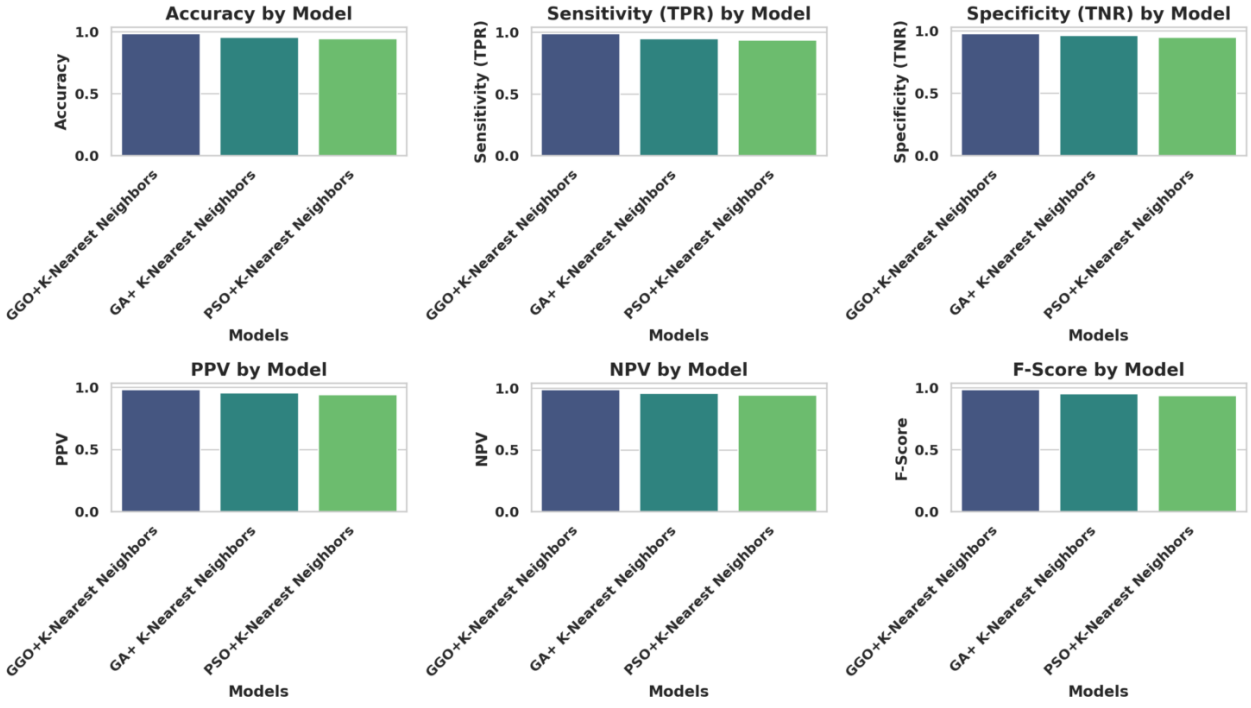

Figure 10: Bar plots of evaluation metrics across optimized K-Nearest Neighbors models.

follow a Gaussian pattern. The Positive Predictive Value (PPV) histogram, also included in Figure 11, reveals the spread and central tendency of precision in positive predictions relative to the fitted normal curve. The histogram of Negative Predictive Value (NPV) in Figure 11 highlights the concentration of reliable negative predictions, closely following the expected normal distribution. Finally, the F-Score histogram in Figure 11 shows the balance between precision and recall, visualizing how the overall performance metric is distributed across models compared to a theoretical Gaussian curve. Together, these histograms provide a deeper insight into both central tendency and distributional characteristics of evaluation metrics.

5 Discussion

The findings from this study provide compelling evidence that machine learning, when coupled with metaheuristic optimization, offers a robust and scalable approach to automatically classify apple quality. A comparison of baseline models showed that the K-Nearest Neighbors (KNN) algorithm significantly outperformed other traditional models, such as Naive Bayes, Decision Tree, and Gradient Boosting, in terms of key evaluation metrics: accuracy, sensitivity, specificity, and F-score. These

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

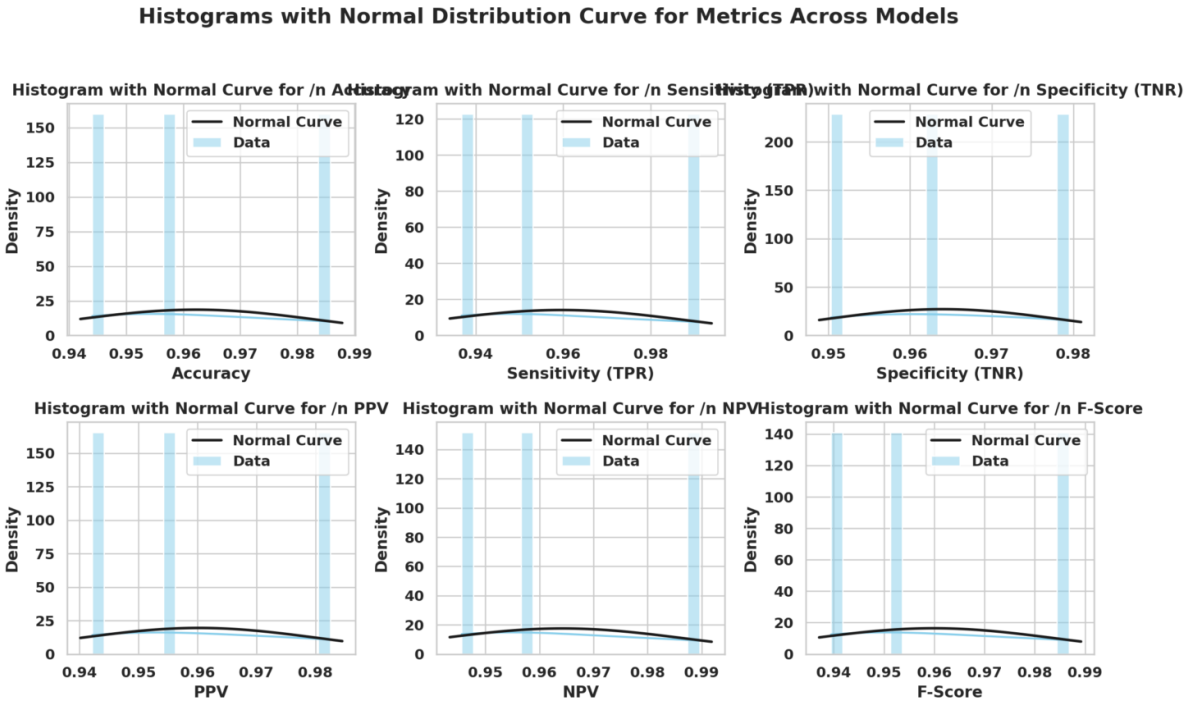

Figure 11: Histograms with normal distribution curve for evaluation metrics across models.

results corroborate previous research, which has highlighted KNN's strength in capturing local data structures effectively, especially when working with well-defined feature spaces. Its inherent simplicity, flexibility, and interpretability make KNN an excellent choice for initial classification tasks in many machine learning applications. However, while KNN showed remarkable performance at the baseline level, it was also found to be sensitive to irrelevant or redundant features, particularly in the presence of noise. This limitation underscores the importance of applying optimization techniques to fine-tune both feature relevance and hyperparameter settings, which in turn, improves the overall model performance. The optimization phase, which incorporated metaheuristic algorithms, showed substantial improvements in classification outcomes. Notably, the Guided Gorilla Optimizer (GGO) achieved the highest performance across all evaluation metrics, with an accuracy reaching 99%, as well as balanced increases in sensitivity, specificity, positive predictive value (PPV), and F-score. This indicates that the GGO algorithm's ability to balance exploration and exploitation enabled it to effectively escape local minima and optimize the parameters of the KNN model. This improvement is consistent with previous studies that demonstrated the power of evolutionary and swarm-based algorithms in addressing high-dimensional, complex optimization problems. The success of GGO, in particular, suggests that it possesses a superior capacity for generalization when compared to GA and PSO, as evidenced by its ability to handle unseen data with minimal performance degradation. This finding is crucial for real-world industrial applications, where the ability to generalize to new, previously unseen data is paramount. Moreover, the ability of GGO to optimize KNN's parameters effectively suggests that it holds significant promise for large-scale, automated quality assessment systems in the agricultural industry. The broader implications of these results extend beyond the context of apple quality classification. This study contributes to the growing body of research emphasizing the critical role of artificial intelligence (AI) in modern agriculture. The proposed framework demonstrates significant potential for improving the efficiency and consistency of quality control processes. By replacing the subjective and inconsistent methods of manual inspection, which are prone to human error and labor inefficiency, AI-powered systems offer a more scalable, accurate, and reliable alternative for quality assessment. Additionally, AI-driven systems can help minimize post-harvest losses by detecting subtle quality defects early, thus enhancing consumer satisfaction and reducing waste in the supply chain. The long-term success of GGO in optimizing KNN further highlights the importance of selecting the appropriate metaheuristic optimization method tailored to specific agricultural tasks. Proper optimization is crucial in ensuring that AI systems can handle the complexity and variability of agricultural data, ultimately contributing to more sustainable and efficient practices. However, despite the promising results, this study does have some limitations that need to be addressed in future work. One notable limitation is the reliance on a specific dataset, which may not fully capture the diversity of apple varieties, environmental conditions, and cultivation practices found in real-world scenarios. To address this, future research should explore the integration of multiple datasets that encompass a wider range of variables and conditions. Additionally, the lack of real-time deployment of the system poses a challenge in assessing its practicality for industrial-scale operations. Real-time data processing, coupled with continuous optimization, will be necessary for the framework to be truly applicable in operational environments. The future work should also focus on the integration of multimodal data sources, such as hyperspectral imaging, multispectral sensors, and sensor fusion, to further enhance the model's ability to capture subtle quality variations that might be missed by traditional methods. Real-time optimization pipelines and domain-specific adaptations should also be explored to increase both the accuracy and robustness of the model. Furthermore, while the current framework focuses on apples, extending this approach to other fruits and perishable goods could open up new avenues for automated quality assessment in a variety of agricultural sectors. In conclusion, this study provides a novel approach for apple quality assessment using machine

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

learning and metaheuristic optimization. The integration of KNN with GGO optimization demonstrates a scalable, accurate, and practical solution for automated quality classification, offering significant advantages over manual inspection systems. This research serves as a foundation for further advancements in AI-driven agricultural technologies and sets the stage for future work aimed at improving the robustness, scalability, and real-time deployment of automated quality assessment systems in the agriculture industry. By leveraging AI and optimization techniques, this research paves the way for a more efficient, data-driven, and sustainable agricultural supply chain that can meet the growing demands of global food production while reducing waste and improving product quality.

6 Conclusion and Future Work

This paper proposes a computerized system for evaluating apple quality using machine learning classifiers and metaheuristic optimization. It compared four baseline models: K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Gradient Boosting, and their results indicated that KNN had the best baseline accuracy of 93.6%. To further improve performance, three metaheuristic optimization methods — Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Guided Gorilla Optimizer (GGO) — were utilized in the tuning of hyperparameters. Among them, GGO was always better than GA and PSO, increasing the accuracy of KNN by 98.6% and its sensitivity by 99.1% and F-Score by 98.7%. The results indicate that the KNN-GGO solution is reliable, accurate, and scalable for automatically classifying apple quality. In addition to enhancing classification accuracy, it overcomes the constraints of traditional manual inspection via a more data-driven and uniform technique that can be implemented in large-scale industries. Despite the promising results, several avenues for future exploration remain. To begin with, the existing work was performed using a structured dataset with predetermined physical and sensory characteristics; it may be more robust and generalized to multimodal data sources, such as hyperspectral imaging, RGB images, or IoT sensor data. Second, although GGO proved to be more optimized, a future study on a hybrid or adaptive metaheuristic approach would enhance the speed of convergence and minimize computational cost. Third, implementing the suggested framework in practice in industrial settings would involve addressing scalability and latency issues, such as integrating automated grading machines and supply chain systems. Lastly, the domain-specific adoptions, including adapting the models to various types of apples or applying the method to other perishable fruits, would lead to the increased generalizability of the framework. Following these guidelines, future work could make the use of intelligent optimization in agricultural quality control more significant, leading to more sustainable and efficient food supply chains.

Acknowledgment

Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R308), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

Data Availability Statement

The smart home energy consumption and weather data used in this study are publicly available at: `https://www.kaggle.com/datasets/zeesolver/apple-quality/data`.

References

[1] Li, X., Gao, W., Wang, L., Chen, Y., Cai, Z., Wu, D., Chen, N., Jiang, Q., Zheng, Z., Zhu, J., Wang, Z.: Co-fermentation of lactobacillus plantarum and lactobacillus casei improves in vitro antioxidant capacity and quality of apple juice. Fermentation **11**(4) (2025) https://doi.org/10.3390/fermentation11040161

[2] Akter, T., Faqeerzada, M.A., Kim, Y., Pahlawan, M.F.R., Aline, U., Kim, H., Kim, H., Cho, B.-K.: Hyperspectral imaging with multivariate analysis for detection of exterior flaws for quality evaluation of apples and pears. Postharvest Biology and Technology **223**, 113453 (2025) https://doi.org/10.1016/j.postharvbio.2025.113453

[3] Okba, S.K., Abo Ogiela, H.M., Mehesen, A., Mikhael, G.B., Alam-Eldein, S.M., Tubeileh, A.M.S.: Influence of compost and biological fertilization with reducing the rates of mineral fertilizers on vegetative growth, nutritional status, yield and fruit quality of 'anna' apples. Agronomy **15**(3) (2025) https://doi.org/10.3390/agronomy15030662

[4] Iyoubi, E.M., Boq, R.E., Tetouani, S., Cherkaoui, O., Soulhi, A.: Advanced apple conformity detection through fuzzy logic: A novel approach to post-harvest quality control. Engineering Proceedings **97**(1) (2025) https://doi.org/10.3390/engproc2025097051

[5] Nisenbaum, M., Agustinelli, S.P., Guzmán, M.N., Murialdo, S.E.: Biospeckle laser for real-time monitoring of water and food samples: Enhancing contamination detection and quality control in field applications. Instrumentation Science & Technology, 1–35 (2025) https://doi.org/10.1080/10739149.2025.2506110

[6] Liu, X., Le Bourvellec, C., Yu, J., Zhao, L., Wang, K., Tao, Y., Renard, C.M.G.C., Hu, Z.: Trends and challenges on fruit and vegetable processing: Insights into sustainable, traceable, precise, healthy, intelligent, personalized and local innovative food products. Trends in Food Science & Technology **125**, 12–25 (2022) https://doi.org/10.1016/j.tifs.2022.04.016

[7] Wang, X., Feng, Y., Wang, Y., Zhu, H., Song, D., Shen, C., Luo, Y.: Enhancing optical non-destructive methods for food quality and safety assessments with machine learning techniques: A survey. Journal of Agriculture and Food Research **19**, 101734 (2025) https://doi.org/10.1016/j.jafr.2025.101734

[8] He, Z., Cao, L., Zhao, L., Zhou, Y., Gong, J., Zhu, C., Tan, C.: Effects of the mixed fermentation of apples and yeast on the sensory evaluation, physicochemical composition and flavor of coffee beans. Food Bioscience **68**, 106337 (2025) https://doi.org/10.1016/j.fbio.2025.106337

[9] Rojas Santelices, I., Cano, S., Moreira, F., Pe
na Fritz, A.: Artificial vision systems for fruit inspection and classification: Systematic literature review. Sensors **25**(5) (2025) https://doi.org/10.3390/s25051524

[10] Jia, X., Schneider, F., Ning, M., Ding, J.: Aesthetic grading causes food losses without financially benefiting farmers: Micro-level evidence from china's fresh apple supply chain. Waste Management & Research **43**(6), 957–968 (2025) https://doi.org/10.1177/0734242X241280097

[11] Rosati, R., Romeo, L., Cecchini, G., Tonetto, F., Viti, P., Mancini, A., Frontoni, E.: From knowledge-based to big data analytic model: A novel iot and machine learning based decision support system for predictive maintenance in industry 4.0. Journal of Intelligent Manufacturing **34**(1), 107–121 (2023) https://doi.org/10.1007/s10845-022-01960-x

[12] Chhetri, K.B.: Applications of artificial intelligence and machine learning in food quality control and safety assessment. Food Engineering Reviews **16**(1), 1–21 (2024) https://doi.org/10.1007/s12393-023-09363-1

[13] Taner, A., Mengstu, M.T., Selvi, K.Ç., Duran, H., Gür, İ., Ungureanu, N.: Apple varieties classification using deep features and machine learning. Agriculture **14**(2), 2 (2024) https://doi.org/10.3390/agriculture14020252

[14] Albahra, S., Gorbett, T., Robertson, S., D'Aleo, G., Kumar, S.V.S., Ockunzzi, S., Lallo, D., Hu, B., Rashidi, H.H.: Artificial intelligence and machine learning overview in pathology & laboratory medicine: A general review of data preprocessing and basic supervised concepts. Seminars in Diagnostic Pathology **40**(2), 71–87 (2023) https://doi.org/10.1053/j.semdp.2023.02.002

[15] Ghazal, S., Qureshi, W.S., Khan, U.S., Iqbal, J., Rashid, N., Tiwana, M.I.: Analysis of visual features and classifiers for fruit classification problem. Computers and Electronics in Agriculture **187**, 106267 (2021) https://doi.org/10.1016/j.compag.2021.106267

[16] Ahmad, T., Madonski, R., Zhang, D., Huang, C., Mujeeb, A.: Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: Key developments, challenges, and future research opportunities in the context of smart grid paradigm. Renewable and Sustainable Energy Reviews **160**, 112128 (2022) https://doi.org/10.1016/j.rser.2022.112128

[17] Lara-Montaño, O.D., Gómez-Castro, F.I., Gutiérrez-Antonio, C., Dragoi, E.N.: Success-based optimization algorithm (sboa): Development and enhancement of a metaheuristic optimizer. Computers & Chemical Engineering **194**, 108987 (2025) https://doi.org/10.1016/j.compchemeng.2024.108987

[18] Sharma, P., Raju, S.: Metaheuristic optimization algorithms: a comprehensive overview and classification of benchmark test functions. Soft Computing-A Fusion of Foundations, Methodologies & Applications **28**(4) (2024) https://doi.org/10.1007/s00500-023-09276-5

[19] Elshewey, A.M., Abed, A.H., Khafaga, D.S., Alhussan, A.A., Eid, M.M., El-kenawy, E.-S.M.: Enhancing heart disease classification based on greylag goose optimization algorithm and long short-term memory. Scientific Reports **15**(1), 1277 (2025) https://doi.org/10.1038/s41598-024-83592-0

[20] Kılıçarslan, S., Dönmez, E., Kılıçarslan, S.: Identification of apple varieties using hybrid transfer learning and multi-level feature extraction. European Food Research and Technology **250**(3), 895–909 (2024) https://doi.org/10.1007/s00217-023-04436-1

[21] Bhargava, A., Bansal, A.: Classification and grading of multiple varieties of apple fruit. Food Analytical Methods **14**(7), 1359–1368 (2021) https://doi.org/10.1007/s12161-021-01970-0

[22] Biffi, L.J., Mitishita, E., Liesenberg, V., Santos, A.A., Gonçalves, D.N., Estrabis, N.V., Silva, J.d.A., Osco, L.P., Ramos, A.P.M., Centeno, J.A.S., Schimalski, M.B., Rufato, L., Neto, S.L.R., Marcato Junior, J., Gonçalves, W.N.: Atss deep learning-based approach to detect apple fruits. Remote Sensing **13**(1), 1 (2021) https://doi.org/10.3390/rs13010054

[23] Çetin, N., Karaman, K., Kavuncuoğlu, E., Yıldırım, B., Jahanbakhshi, A.: Using hyperspectral imaging technology and machine learning algorithms for assessing internal quality parameters of apple fruits. Chemometrics and Intelligent Laboratory Systems **230**, 104650 (2022) https://doi.org/10.1016/j.chemolab.2022.104650

[24] Chao, X., Sun, G., Zhao, H., Li, M., He, D.: Identification of apple tree leaf diseases based on deep learning models. Symmetry **12**(7), 7 (2020) https://doi.org/10.3390/sym12071065

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025

[25] Chen, J., Han, J., Liu, C., Wang, Y., Shen, H., Li, L.: A deep-learning method for the classification of apple varieties via leaf images from different growth periods in natural environment. Symmetry **14**(8), 8 (2022) https://doi.org/10.3390/sym14081671

[26] Chu, P., Li, Z., Lammers, K., Lu, R., Liu, X.: Deep learning-based apple detection using a suppression mask r-cnn. Pattern Recognition Letters **147**, 206–211 (2021) https://doi.org/10.1016/j.patrec.2021.04.022

[27] Gao, F., Fang, W., Sun, X., Wu, Z., Zhao, G., Li, G., Li, R., Fu, L., Zhang, Q.: A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. Computers and Electronics in Agriculture **197**, 107000 (2022) https://doi.org/10.1016/j.compag.2022.107000

[28] Khan, A.I., Quadri, S.M.K., Banday, S., Latief Shah, J.: Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. Computers and Electronics in Agriculture **198**, 107093 (2022) https://doi.org/10.1016/j.compag.2022.107093

[29] Li, Y., Feng, X., Liu, Y., Han, X.: Apple quality identification and classification by image processing based on convolutional neural networks. Scientific Reports **11**(1), 16618 (2021) https://doi.org/10.1038/s41598-021-96103-2

[30] Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., Whiting, M.D.: Deep learning based segmentation for automated training of apple trees on trellis wires. Computers and Electronics in Agriculture **170**, 105277 (2020) https://doi.org/10.1016/j.compag.2020.105277

[31] Mirbod, O., Choi, D., Heinemann, P.H., Marini, R.P., He, L.: On-tree apple fruit size estimation using stereo vision with deep learning-based occlusion handling. Biosystems Engineering **226**, 27–42 (2023) https://doi.org/10.1016/j.biosystemseng.2022.12.008

[32] Tian, K., Zhu, W., Wang, M., Chen, T., Li, F., Xie, J., Peng, Y., Sun, T., Zhou, G., Hu, D.: Qualitative and quantitative assessment of apple quality using bulk optical properties in combination with machine learning and chemometrics techniques. LWT **211**, 116894 (2024) https://doi.org/10.1016/j.lwt.2024.116894

[33] Villacrés, J., Viscaino, M., Delpiano, J., Vougioukas, S., Auat Cheein, F.: Apple orchard production estimation using deep learning strategies: A comparison of tracking-by-detection algorithms. Computers and Electronics in Agriculture **204**, 107513 (2023) https://doi.org/10.1016/j.compag.2022.107513

[34] Wang, X.A., Tang, J., Whitty, M.: Deepphenology: Estimation of apple flower phenology distributions based on deep learning. Computers and Electronics in Agriculture **185**, 106123 (2021) https://doi.org/10.1016/j.compag.2021.106123

[35] Wu, J., Yang, G., Yang, H., Zhu, Y., Li, Z., Lei, L., Zhao, C.: Extracting apple tree crown information from remote imagery using deep learning. Computers and Electronics in Agriculture **174**, 105504 (2020) https://doi.org/10.1016/j.compag.2020.105504

[36] Xuan, G., Gao, C., Shao, Y., Zhang, M., Wang, Y., Zhong, J., Li, Q., Peng, H.: Apple detection in natural environment using deep learning algorithms. IEEE Access **8**, 216772–216780 (2020) https://doi.org/10.1109/ACCESS.2020.3040423

[37] Yu, F., Lu, T., Xue, C.: Deep learning-based intelligent apple variety classification system and model interpretability analysis. Foods **12**(4), 4 (2023) https://doi.org/10.3390/foods12040885

[38] Zhan, C., Mao, H., Fan, R., He, T., Qing, R., Zhang, W., Lin, Y., Li, K., Wang, L., Xia, T., Wu, Y., Kang, Z.: Detection of apple sucrose concentration based on fluorescence hyperspectral image system and machine learning. Foods **13**(22), 22 (2024) https://doi.org/10.3390/foods13223547

[39] Zou, Z., Long, T., Wang, Q., Wang, L., Chen, J., Zou, B., Xu, L.: Implementation of apple's automatic sorting system based on machine learning. Food Science and Technology **42**, 24922 (2022) https://doi.org/10.1590/fst.24922

[40] Zhang, X., He, L., Zhang, J., Whiting, M.D., Karkee, M., Zhang, Q.: Determination of key canopy parameters for mass mechanical apple harvesting using supervised machine learning and principal component analysis (pca). Biosystems Engineering **193**, 247–263 (2020) https://doi.org/10.1016/j.biosystemseng.2020.03.006

DOI: https://doi.org/10.54216/JAIM.090106

Received: October 30, 2024 Revised: December 11, 2024 Accepted: January 14, 2025