Vol. 8, No. 4, April 2026

ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

A Lightweight Classical Machine Learning Pipeline for Rice NPK Deficiency Classification Using Hand-Crafted Feature Fusion

Dhiyaussalam[1], Kun Nursyaiful Priyo Pamungkas[2], Wanvy Arifha Saputra[3], Ahmad Yusuf[4]

[1,2,3,4]Informatics Engineering, Politeknik Negeri Banjarmasin, Banjarmasin, Indonesia

Received:
October 5, 2025
Revised:
March 10, 2026
Accepted:
April 1, 2026
Published:
April 12, 2026

Corresponding Author:
Author Name*:
Dhiyaussalam
Email*:
salam@poliban.ac.id

DOI:
10.63158/journalisi.v8i2.1486

Abstract. Many high-accuracy deep learning solutions for plant nutrient deficiency remain impractical in resource-limited settings due to computational cost and limited explainability. This study proposes a lightweight classical machine learning pipeline for rice leaf NPK (nitrogen, phosphorus, potassium) deficiency classification on the publicly available Kaggle Nutrient-Deficiency-Symptoms-in-Rice dataset (1,156 images); all results should be interpreted in this dataset context rather than as field-validated performance. The pipeline applies HSV-based leaf segmentation to reduce background influence. It extracts a 126-dimensional feature set combining masked color moments, HSV histograms, vegetation indices, LBP and GLCM texture descriptors, and spatial symptom ratios. Hyperparameters are tuned via RandomizedSearchCV with 5-fold StratifiedKFold and macro-F1 scoring; final evaluation uses a held-out 80/20 stratified test set kept separate throughout tuning. XGBoost achieves the best test performance (accuracy 0.9267; macro-F1 0.9233), followed by SVM-RBF (0.9224; 0.9187) and Random Forest. Feature importance analysis confirms that color moments dominate class separability, with texture and spatial features providing complementary support. The dominant remaining error is phosphorus–potassium confusion. The novelty lies in integrating leaf-focused preprocessing with a structured, low-cost feature representation suitable for mobile or edge deployment.

Keywords: classical machine learning; hand-crafted features; feature fusion; NPK deficiency classification; rice leaf

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

1. INTRODUCTION

Rice is a staple crop that supports food security in many regions, and its productivity depends strongly on balanced plant nutrition. Among the essential macronutrients, nitrogen (N), phosphorus (P), and potassium (K) play a central role in vegetative growth and grain formation [1], [2]. Deficiency in these nutrients often produces visible symptoms on leaves, including chlorosis, marginal necrosis, and changes in leaf color and structure. These symptoms provide a practical basis for early diagnosis, especially when laboratory testing is not available [1], [2], [3].

In real farming practice, nutrient deficiencies are still frequently diagnosed by farmers or extension officers through visual inspection. However, manual diagnosis can be inconsistent because leaf appearance is influenced by lighting, camera quality, growth stage, and background conditions [4]. Moreover, different nutrient deficiencies can share overlapping symptoms, increasing the risk of misdiagnosis and inappropriate fertilizer use. This motivates the development of automated, image-based approaches to support faster, more consistent screening of rice nutrient status [5].

Publicly available datasets enable systematic research on this topic, including the Kaggle rice nutrient deficiency dataset that focuses on three classes (N, P, K) [4]. Although such datasets are useful for benchmarking, they also introduce practical challenges. The Kaggle set is relatively small and often contains real-world variation in background, illumination, and leaf positioning [5]. These factors can encourage shortcut learning, in which models rely on background cues rather than leaf symptoms [6]. In addition, phosphorus and potassium deficiencies can be visually similar, and prior studies repeatedly report confusion between these two classes, suggesting that class separability is a core difficulty rather than a purely modelling issue [7].

Early research explored classical machine learning using hand-crafted features. For example, one study applied Learning Vector Quantization (LVQ) with a compact feature set derived from leaf images to classify N, P, and K deficiencies, reporting moderate performance and demonstrating that feature-based pipelines can work with limited data [8]. This line of work supports the value of interpretable pipelines. However, it also

indicates that discriminative power may be constrained when features do not explicitly control for background variation or capture subtle symptom patterns.

More recent studies have shifted toward deep learning and transfer learning to overcome limitations in feature design. On the same Kaggle dataset, ensemble strategies combining pre-trained CNNs have been shown to improve average performance over single-model baselines [9], while deep ensemble approaches with weighted averaging and extensive augmentation have reported accuracy as high as 98.33% [2]. Hybrid configurations combining CNN feature extraction with SVM classifiers have also been reported to reach approximately 99% accuracy under certain experimental conditions [1].

However, these performance figures are not always directly comparable across studies due to differences in augmentation intensity, split definitions, and evaluation protocols. More fundamentally, deep and hybrid approaches often require substantial computational resources and offer limited interpretability, which can reduce their suitability for lightweight deployment and scientific transparency.

Taken together, these findings reveal persistent gaps that remain inadequately addressed: background variability encourages shortcut learning, phosphorus–potassium confusion persists across studies, and the trade-off between model complexity and practical deployability has received limited attention. Although deep learning and hybrid approaches have demonstrated strong performance on the Kaggle rice NPK dataset, they typically require substantial computational resources, rely on extensive data augmentation to compensate for the small dataset size, and offer limited interpretability that can hinder both practical deployment and scientific transparency. Moreover, the recurring P-K confusion suggests that feature representation quality, rather than model capacity alone, is the core bottleneck. This study addresses these limitations by proposing a hand-crafted, mask-guided classical machine learning pipeline: it explicitly reduces background influence through HSV-based segmentation, combines complementary multi-cue features in a compact and interpretable representation, and is evaluated under a transparent and reproducible protocol that reports not only accuracy but also macro precision, macro recall, and macro F1-score to reflect balanced performance across classes. This approach remains valuable not as a competitor to deep learning in terms of

peak accuracy, but as a practical, lightweight, and explainable alternative suited for resource-constrained deployment scenarios.

Novelty and contributions of this work are summarized as follows: 1. A classical machine learning pipeline for rice NPK deficiency recognition is developed, in which background influence is explicitly reduced through leaf-focused preprocessing prior to feature extraction; 2. A structured feature representation is designed by combining complementary cues, including color statistics, color distribution, vegetation-related color transformations, texture descriptors, and morphological indicators, so that symptom patterns can be captured in a compact and interpretable form; 3. A systematic comparison of multiple classical learners, including kernel-based and tree-based models, is conducted under consistent tuning and evaluation, and macro-averaged metrics are reported to reduce majority-class bias; 4. An error-oriented analysis is performed to highlight remaining confusion patterns, particularly between phosphorus and potassium, and the implications for future dataset refinement and model improvement are discussed.

Based on these motivations, the study is guided by the following research questions:

1) To what extent can competitive performance be achieved on the Kaggle rice NPK deficiency dataset when a classical machine learning pipeline with leaf-focused preprocessing is applied?
2) Under real-world image variability, which feature families are found to contribute most to separability among N, P, and K classes?
3) Which error patterns remain dominant, particularly P–K confusion, and what implications are suggested for future improvements in data and modelling?

2. METHODS

The proposed pipeline is designed as an integrated methodological contribution in which three components work together: (1) HSV-based leaf segmentation that explicitly reduces background influence before feature extraction, (2) a structured multi-cue feature representation that combines complementary color, texture, and spatial descriptors computed on the segmented leaf region, and (3) a systematic model comparison under a consistent tuning and evaluation protocol. While individual components such as HSV

thresholding, LBP, and GLCM are established techniques, their integration within a unified mask-guided pipeline specifically designed for rice NPK deficiency recognition constitutes the novelty of this study.

A comparative experimental design is employed to classify rice leaf nutrient deficiencies into three categories: potassium (K), nitrogen (N), and phosphorus (P). Hand-crafted image features and classical machine learning models are utilized. The primary objective is to assess the performance of different algorithms. Each is trained on an identical feature representation and evaluated using a consistent protocol. The classifiers evaluated include Support Vector Machine with a radial basis function kernel (SVM-RBF), Random Forest (RF), and XGBoost. To ensure comparability, all models use the same dataset splits and feature extraction pipeline. This ensures that observed performance differences primarily reflect the algorithms' learning mechanisms, rather than variations in input data or preprocessing.

The overall workflow begins with data preparation and ends with model evaluation and error analysis. After the rice leaf images are collected and labelled, the dataset is organized into three classes and is split into training and test sets. Each image is loaded in BGR format and resized to a fixed width of 256 pixels, preserving the original aspect ratio. Next, leaf segmentation is performed in the HSV color space to obtain a leaf mask. The initial mask is generated using simple threshold rules on the saturation and value channels. Morphological operations are applied to remove small noise and fill small holes. The largest connected contour is selected as the leaf region. If segmentation fails or produces an extremely small region, a fallback strategy is applied to prevent feature-extraction errors.

Feature extraction is performed on the segmented leaf region (mask-based). The pipeline combines color statistics, color histograms, vegetation indices, texture descriptors (LBP and GLCM), and spatial symptom features. These descriptors are concatenated into a single feature vector for each image. In the modelling stage, the three classifiers are trained using RandomizedSearchCV with StratifiedKFold cross-validation. This optimizes hyperparameters based on macro-F1. After selecting the best configuration for each model, the trained model is evaluated on the test set. Finally, confusion matrices are

examined to identify dominant misclassification patterns. This supports a more detailed interpretation of the results. The pipeline flowchart is shown in Figure 1.

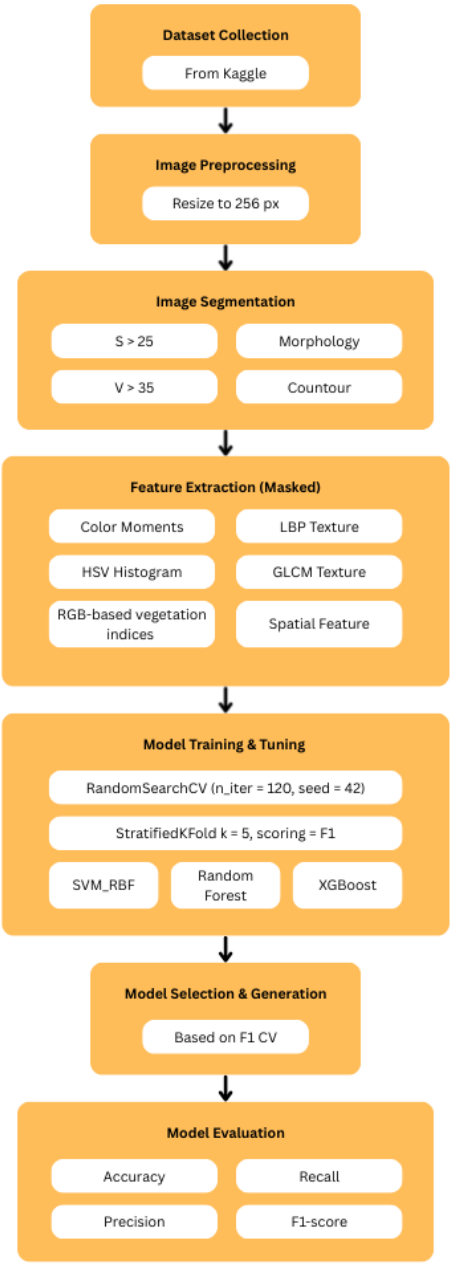

Figure 1. The Pipeline of this study.

All experiments are implemented in Python. The image processing pipeline uses OpenCV 4.13.0 for image loading, resizing, color space conversion, and morphological operations. Texture feature extraction relies on scikit-image 0.26.0 for Local Binary Pattern and Gray Level Co-occurrence Matrix computation. Feature vector construction and numerical

operations use NumPy 2.4.1 and SciPy 1.17.0, while dataset management uses Pandas 2.3.3. Model training, hyperparameter tuning, and cross-validation are performed using scikit-learn 1.8.0, and XGBoost 3.1.3 is used for the gradient boosting classifier. Trained models are serialized using Joblib 1.5.3, and result visualization is produced with Matplotlib 3.10.8.

2.1. Dataset

The dataset used in this study is obtained from a public Kaggle repository titled "Nutrient-Deficiency-Symptoms-in-Rice" [9]. The dataset contains rice leaf images depicting nutrient deficiency symptoms and provides class labels for supervised learning. Using an open dataset improves transparency and enables other researchers to reproduce the same experimental pipeline under comparable settings. After downloading the dataset, the images are filtered and organized to match the target categories in this research, namely K, N, and P deficiency. Images outside these three classes are removed to maintain consistency with the study objective. The remaining dataset contains 1,156 images, with 383 for K, 440 for N, and 333 for P. Representative sample images for each class are shown in Figure 2.

To evaluate model performance fairly, the dataset is split using stratified sampling to preserve class proportions. The split uses 80% for training and 20% for testing, resulting in 924 training images and 232 test images. The class distribution in each split is as follows:

1) Training set (924 images): K = 306, N = 352, P = 266
2) Test set (232 images): K = 77, N = 88, P = 67

The training set is used for model development and hyperparameter tuning. Instead of using a separate fixed validation set, this study uses a 5-fold StratifiedKFold within the training set, with the validation portion formed by the held-out fold in each iteration. The test set is kept separate and used only once for final performance reporting. Since the dataset is publicly shared for research and educational purposes, this study uses it only for scientific analysis and reports results in aggregate form.

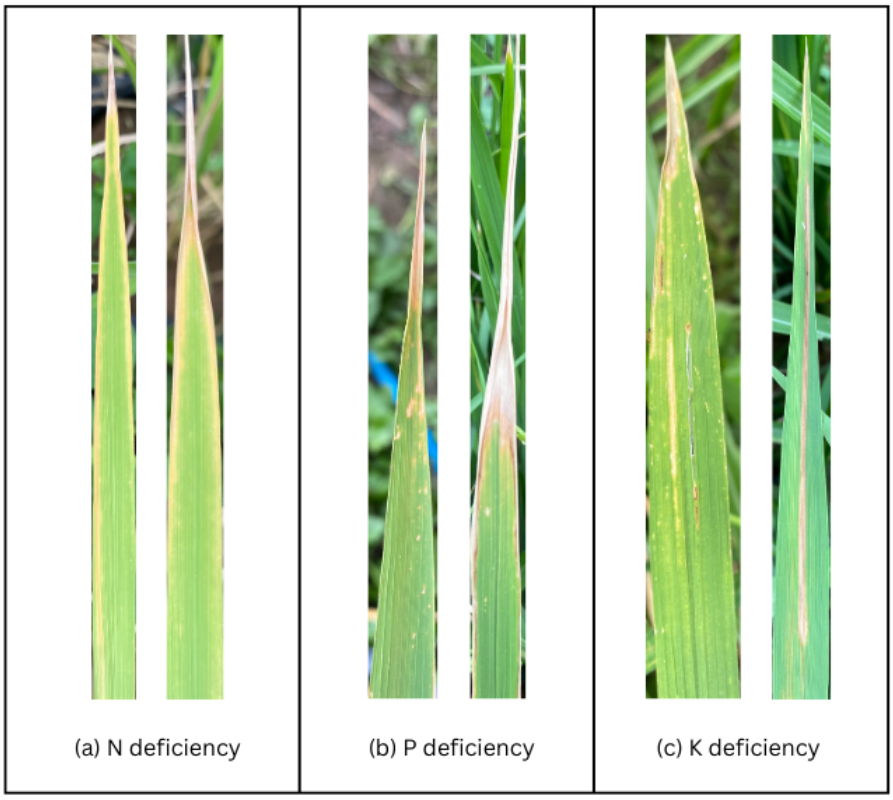

Figure 2. Representative sample images from the Kaggle Nutrient-Deficiency-Symptoms-in-Rice dataset for the three target classes: K, N, and P deficiency.

2.2. Image Preprocessing and Leaf Segmentation

1) Basic preprocessing

Once the dataset is prepared, each rice leaf image undergoes the same processing steps to ensure consistent feature extraction. The image is first loaded from its file path in BGR format, then resized to a fixed width of 256 pixels while maintaining the original aspect ratio [10]. This approach standardizes input for segmentation and feature computation, minimizing scale variation without causing geometric distortion that could alter visual symptom patterns [11].

2) HSV-leaf segmentation with morphology

Leaf segmentation is then performed in the HSV color space. HSV separates color information from brightness more clearly than BGR. This simplifies the separation between leaf and background pixels [12]. The BGR image is converted into HSV, and the S (saturation) and V (value) channels are used to build an initial foreground mask. A pixel is considered part of the leaf if $S > 25$ and $V > 35$. These thresholds help capture pixels

that are sufficiently saturated and not too dark. The initial mask is refined using morphological operations to improve its shape and remove small noise [13].

To refine the initial mask, morphological operations are applied using an elliptical kernel whose size is determined adaptively from the image resolution [14], as shown in Equation 1.

$$k = max\left(3, \left(\frac{\min(h,w)}{200} \times 2\right) + 1\right) \quad (1)$$

where *h* and *w* are the image height and width, and the odd kernel size ensures that the operation is properly centered. In this study, all images are resized to a fixed width of 256 pixels before segmentation. Under this preprocessing setting, *min(h,w)* does not exceed 256, and the rule yields *k=3* for all 1,156 images. Therefore, a consistent *3×3* kernel is effectively used for mask refinement in this experiment. The adaptive formulation is retained to support future use on higher-resolution inputs where kernel scaling may become relevant.

The kernel size is always odd to ensure stable morphology operations, meaning the kernel that is used for shape processing always has an odd number of pixels, so the operations are centered properly [15]. The refinement includes closing for two iterations to fill small holes inside the leaf region, followed by opening for one iteration to remove small isolated noise. 'Closing' and 'opening' are specific morphological operations: closing connects small gaps or holes, and opening removes small objects or noise [16].

Next, the contour of the segmented regions is analyzed. The largest contour is selected as the leaf region, assuming the leaf occupies the largest connected area in the image [17]. To prevent failure cases, a fallback strategy is applied. If no contour is detected or the largest contour covers less than 10 percent of the image, the mask is set to cover the entire image. The fallback rule remains in place as a safety mechanism when segmentation becomes unreliable. However, in this dataset, the fallback condition is not triggered. Across 1,156 processed images, a valid leaf region is consistently obtained from the largest contour, resulting in a fallback rate of 0.00% (0/1,156). The same outcome is observed for all classes, with 0 fallback cases in K (383 images), N (440 images), and P (333 images). The summary of segmentation robustness is shown in Table 1. These results indicate that the HSV-based thresholds and morphological refinement are stable for the

visual characteristics of the selected dataset. This fallback avoids missing feature extraction when segmentation is unreliable. Issues such as challenging lighting, background similarity, or low contrast may cause unreliability [18].

Table 1. Leaf segmentation reliability across classes

Class	Number of Images	Fallback Count	Fallback Rate (%)
N	440	0	0.00
P	333	0	0.00
K	383	0	0.00
Total	**1,156**	**0**	**0.00**

Note: Fallback is defined as using a full-image mask when no contour is detected or when the largest contour covers less than 10% of the image area.

3) ROI from the mask bounding box

After obtaining the final mask, the leaf region is localized using the mask's bounding box [19]. The image is cropped to this bounding box to form the ROI, so texture features focus on the leaf area and reduce background influence [20]. The ROI is then resized to 128×128 to standardize the input to the texture computation [21]. Finally, the ROI intensity is quantized to 32 gray levels, reducing the complexity of the gray levels and stabilizing GLCM computation by limiting the number of possible intensity transitions [22].

2.3. Feature Engineering and Feature Vector Construction

The feature set is designed based on the visual characteristics of rice NPK deficiency symptoms. Color moments and HSV histograms are included because nitrogen, phosphorus, and potassium deficiencies each produce distinct color changes on leaf surfaces. Vegetation indices are added as compact proxies for greenness and color imbalance. LBP and GLCM texture descriptors are included to capture surface irregularities and second-order patterns that accompany symptom progression. Spatial symptom features are incorporated to reflect the region-specific distribution of discoloration, which differs across deficiency types. All features except vegetation indices are computed on the masked leaf region to minimize background influence.

To compactly and measurably represent nutrient deficiency symptoms, this study extracts a set of hand-crafted features from each rice leaf image. The design integrates color statistics, color distribution, vegetation-related cues, texture patterns, and symptom distribution across leaf regions. Most features are computed on the segmented leaf area using the final mask to minimize background influence, while a few are computed globally to preserve overall color characteristics. Each image is thus represented by a 126-dimensional feature vector, described in the following sub-sections.

1) Color moments (RGB and HSV, masked leaf region)

Color moments capture color statistics that summarize the appearance of the leaf region. For both the RGB and HSV color spaces, three statistical moments are computed for each channel: the mean, standard deviation, and skewness. Since each color space has three channels, this results in 3 channels × 3 moments = 9 features for RGB and the same for HSV. Therefore, the total number of color-moment features is 18. These features are computed on the segmented leaf region (masked) to reduce background influence and to better represent deficiency symptoms on leaf surfaces. [23].

2) HSV histogram (masked, 16 bins per channel)

To capture the distribution of colors on the leaf surface more explicitly, HSV histograms are computed on the masked leaf region. The H, S, and V channels are each quantized into 16 bins, and the histogram counts are derived only from pixels inside the leaf mask. The three histograms are then concatenated, resulting in 3 channels × 16 bins = 48 features. This representation helps describe how color intensity and saturation values are distributed across the leaf area, which can reflect visible deficiency symptoms such as color shifts [24].

3) RGB-based vegetation indices (masked, mean values)

Vegetation indices derived from RGB channels are included as simple cues that highlight changes in greenness and color imbalance, which often accompany nutrient deficiency symptoms. Six indices are computed, namely ExG, ExR, ExGR, NGR, NGB, and NR. For each index, the mean value is calculated over the masked leaf pixels. This produces 6 features. These indices provide a compact summary of color dominance patterns that may be associated with yellowing, browning, or reduced greenness [25].

4) LBP texture (uniform, masked, P=16 and R=2)

Local Binary Pattern (LBP) features are used to represent fine-grained texture changes on the leaf surface. In this study, this study specifically applied uniform LBP with parameters P=16 neighbors and R=2 radius. For uniform LBP, the number of histogram bins is P + 2, resulting in 18 bins [26]. This LBP histogram is then computed on the masked leaf area and used directly as the texture descriptor, yielding 18 features. By doing so, this descriptor helps capture local texture differences that may appear as irregular patches, speckles, or subtle surface changes [27].

5) GLCM texture (masked ROI, multi-distance, multi-angle)

To capture second-order texture statistics, Gray Level Co-occurrence Matrix (GLCM) features are computed from a standardized ROI derived from the leaf mask. The ROI is obtained from the mask bounding box, resized to 128 × 128, and quantized to 32 gray levels before computing co-occurrence statistics [6]. GLCM is then computed at three distances (1,2,3) and four angles (0°,45°,90°,135°). For each distance, six common properties are extracted: contrast, dissimilarity, homogeneity, ASM, energy, and correlation [28]. To reduce directional sensitivity, the values are averaged across angles for each distance. This produces 6 properties × 3 distances = 18 features. These features describe broader texture structure and can support discrimination when color cues alone are not sufficient.

6) Spatial symptom features (masked, region-based color ratios)

Beyond global statistics, deficiency symptoms can appear more pronounced in specific leaf regions, such as the tips or margins. To capture this spatial pattern, region-based color ratios are computed on the masked leaf area. The method measures the ratios of yellow, brown, and green pixels within several predefined zones [29], [30]:

a) Vertical segments: top, middle, and bottom zones (3 zones × 3 ratios = 9 features)
b) Left-right margin: average of left and right margin areas (1 zone × 3 ratios = 3 features)
c) Center band: central region (1 zone × 3 ratios = 3 features)
d) Tip strip: upper 20 percent of the leaf region (1 zone × 3 ratios = 3 features)

The total number of spatial symptom features is 18. These features aim to reflect where color-related symptoms concentrate, which is often informative for separating visually similar deficiency classes.

7) Feature concatenation and final feature dimension (126 features)

All extracted features are concatenated into a single vector for each image. The final feature set consists of: 18 color moments, 48 HSV histogram features, 6 vegetation indices, 18 LBP features, 18 GLCM features, and 18 spatial symptom features. In total, each image is represented by 126 features. This unified feature vector is used as the input to the three classifiers (SVM-RBF, Random Forest, and XGBoost) during model training and hyperparameter tuning.

2.4. Model Development, Hyperparameter Tuning, and Evaluation Protocol

This study compares three classical machine learning classifiers for nutrient deficiency classification, all using the same 126-feature representation. To ensure fairness, all models apply the same tuning strategy, cross-validation, and scoring metric. Prior to training, the 126-dimensional feature vector is standardized using StandardScaler for SVM-RBF, which transforms each feature to zero mean and unit variance (z-score normalization. This step is applied only to SVM-RBF because kernel-based methods are sensitive to heterogeneous feature scales. In contrast, tree-based models such as Random Forest and XGBoost are scale-invariant by design. The scaler is fitted exclusively on the training set and applied to the test set to prevent data leakage. Hyperparameter tuning is performed with RandomizedSearchCV (120 iterations, fixed random seed). Model selection relies on macro-F1, computed with 5-fold StratifiedKFold cross-validation. After selecting the best configuration for each model, the final evaluation is reported on a held-out test set using accuracy and macro-averaged metrics.

1) SVM-RBF

Because SVM-RBF operates in a kernel-induced space where inter-feature scale differences can disproportionately influence the margin, the feature vector is standardized with StandardScaler before this classifier is applied. Support Vector Machine with an RBF kernel (SVM-RBF) is used to model potentially non-linear class boundaries in the feature space [31]. In this setting, the RBF kernel allows the classifier to implicitly map features into a higher-dimensional space, separating classes that are

not linearly separable. Two key hyperparameters are tuned: C, which controls the penalty for misclassification and thus the margin–error trade-off, and gamma, which controls the influence range of each training sample and affects the smoothness of the decision boundary. By tuning these parameters, the model can balance underfitting and overfitting and adapt to the complexity of the hand-crafted feature representation [32].

2) Random Forest

Random Forest (RF) is included as an ensemble baseline that combines many decision trees trained on bootstrapped samples [33]. This model is robust to noisy features and can capture non-linear relationships without strong assumptions about feature distributions. In this study, the RF configuration is tuned to control both model capacity and generalization. Parameters such as the number of trees, maximum depth, feature subsampling strategy, and minimum split and leaf sizes are adjusted to reduce variance while maintaining sufficient discriminative power for the three-class problem [34].

3) XGBoost

XGBoost is a gradient-boosting approach that builds an ensemble of trees sequentially, where each new tree focuses on correcting the errors made by previous trees. This model is often effective for structured feature vectors because it can learn complex interactions and can be regularized to reduce overfitting [35]. The tuning process in this study adjusts parameters that influence boosting strength and generalization, including the number of estimators, tree depth, learning rate, subsampling ratios, and regularization terms. This configuration aims to achieve strong performance while keeping the model stable under cross-validation [36].

4) Hyperparameter tuning using RandomizedSearchCV

Hyperparameter optimization is performed using RandomizedSearchCV, which samples parameter combinations from predefined search spaces rather than exhaustively evaluating all combinations. This approach enables broader exploration within a limited computational budget. The tuning setup is fixed across models to enable comparability: n_iter = 120, random seed = 42, and scoring = macro-F1 [37]. The value of n_iter = 120 was selected to provide sufficient exploration of the defined search spaces while maintaining computational efficiency. Given the discrete and moderate-sized parameter grids used for Random Forest and XGBoost, and the continuous log-uniform distributions for SVM-

RBF, 120 iterations offer a practical balance between sampling coverage and training cost on the relatively small dataset. Each sampled configuration is evaluated using the same cross-validation split generator. The best parameter set for each classifier is selected as the configuration that maximizes macro-F1 under cross-validation [38]. For SVM-RBF, the parameters C and gamma are sampled from log-uniform distributions to cover several orders of magnitude. For Random Forest and XGBoost, discrete candidate sets are defined to control model complexity and regularization. The hyperparameter search space and the best hyperparameters obtained from RandomizedSearchCV are reported in Tables 2 and 3, respectively.

Table 2. The Best Hyperparameters for Each Algorithms

Algorithm	Parameter	Search Space (param_distributions)	Type
SVM-RBF	C	log-uniform	continuous
	gamma	log-uniform	continuous
Random Forest	n_estimators	200, 300, 500	discrete
	max_depth	None, 8, 12, 16, 20	discrete
	max_features	sqrt, log2, 0.5	discrete
	min_samples_split	2, 5, 10	discrete
	min_samples_leaf	1, 2, 4	discrete
	class_weight	None, balanced	discrete
XGBoost	n_estimators	200, 300, 500	discrete
	max_depth	3, 4, 5, 6	discrete
	learning_rate	0.01, 0.03, 0.05, 0.1	discrete
	subsample	0.6, 0.8, 0.9, 1.0	discrete
	colsample_bytree	0.6, 0.8, 1.0	discrete
	min_child_weight	1, 3, 5	discrete
	gamma	0, 0.1, 0.2, 0.3	discrete
	reg_alpha	0, 0.1, 0.5	discrete
	reg_lambda	1.0, 1.5, 2.0	discrete

Table 3. The Best Hyperparameters for Each Algorithms

Algorithm	Parameter	Value
SVM-RBF	C	15.3272
	gamma	0.0054242
Random Forest	n_estimators	500
	max_depth	16
	max_features	0.5
	min_samples_split	2
	min_samples_leaf	1
	class_weight	Balanced
XGBoost	n_estimators	300
	max_depth	4
	learning_rate	0.1
	subsample	0.8
	colsample_bytree	1.0
	min_child_weight	1
	gamma	0.2
	reg_alpha	0.1
	reg_lambda	1.0

5) Cross-validation with StratifiedKFold

Model selection is conducted using StratifiedKFold with k = 5, which preserves the class distribution within each fold. Stratification is important in multi-class settings because it reduces the risk that a fold is dominated by a single class, which can bias the tuning process and lead to unstable estimates [39]. Using 5 folds provides a balance between reliable performance estimation and computational efficiency. Cross-validation is applied only on the training portion of the dataset during tuning, while the held-out test set is kept untouched until the final evaluation stage [40].

6) Evaluation metrics

After the best configuration is obtained for each model, performance is reported on the test set using accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correctly classified samples across all classes. Precision describes how many predicted

samples of a class are correct, while recall measures how many true samples of a class are successfully detected. The F1-score is the harmonic mean of precision and recall, providing a single balanced measure [41]. In this study, macro-averaging is used for precision, recall, and F1 to give equal importance to each class (K, N, and P), which is important when class counts are not perfectly balanced. The formulas as shown in Equation 2 to 5 [42]:

$$Accuracy = \frac{\sum TP+\sum TN}{\sum TP+\sum TN+\sum FP+\sum FN} \quad (2)$$

$$Precision = \frac{\sum TP}{\sum TP+\sum FP} \quad (3)$$

$$Recall = \frac{\sum TP}{\sum TP+\sum FN} \quad (4)$$

$$F1 = \frac{2 \times (Recall \times Precision)}{(Recall+Precision)} \quad (5)$$

3. RESULTS AND DISCUSSION

3.1. Performance Evaluation

This section reports the test performance of the three tuned classical machine learning models for classifying rice leaf nutrient deficiency. The discussion interprets the practical implications of the results. To check whether the tuned models are stable during training, this study reports the 5-fold StratifiedKFold cross-validation macro-F1 as mean on the training set. This result reflects the model's consistent performance across folds and supports the reliability of the final test-set evaluation. The result of cross-validation is shown in Table 4.

Table 4. Cross-Validation Macro-F1 Results on the Training Set

Algorithm	F1
SVM-RBF	**0.9391**
Random Forest	0.9010
XGBoost	0.9133

The cross-validation results indicate that the tuned models achieve consistent performance across folds, as shown by the macro-F1 scores across all three models. SVM-RBF obtains the highest average macro-F1 during cross-validation, while XGBoost remains

competitive with stable performance. These findings confirm that the selected hyperparameter configurations are not dependent on a single split of the training data, which strengthens the credibility of the reported test-set results.

Table 5 summarizes the evaluation results using accuracy and macro-averaged precision, recall, and F1-score. Using macro-F1, which treats each nutrient deficiency equally, ensures the models do not overlook any deficiency type. Practically, this balance means farmers can address all types of nutrient issues more reliably, leading to improved crop yields and more effective resource allocation.

Table 5. Test Set Performance of the Three Classifiers

Algorithm	Accuracy	Precision	Recall	F1
SVM-RBF	0.9224	0.9200	0.9179	0.9187
Random Forest	0.8965	0.8926	0.8894	0.8906
XGBoost	**0.9267**	**0.9246**	**0.9223**	**0.9233**

Table 6. Robustness Check (Masked vs Unmasked)

Algorithm	Accuracy (Masked)	Accuracy (Unmasked)	Δ Accuracy	F1 (Masked)	F1 (Unmasked)	Δ F1
SVM-RBF	0.9224	0.9224	**0.0000**	0.9187	0.9180	**+0.0007**
Random Forest	0.8965	0.9052	-0.0087	0.8906	0.9004	-0.0098
XGBoost	**0.9267**	**0.9310**	-0.0043	**0.9233**	**0.9282**	-0.0049

To examine whether the models rely on background color cues, a robustness check is conducted by replacing the masked color moments with global (unmasked) color moments computed from the entire image. Table 6 shows that the performance differences between the two configurations are small across all models, with absolute changes in macro-F1 not exceeding 1%. This indicates that the overall classification results are not strongly driven by background color. The slight advantage of unmasked features in Random Forest and XGBoost may reflect incidental background correlations rather than meaningful symptom information, as the differences remain within expected variation for this dataset size. This check reduces the risk of shortcut learning and

strengthens confidence that the extracted features reflect leaf symptoms rather than acquisition artefacts. Overall, XGBoost provides the best generalization performance on the test set, followed by SVM-RBF, while Random Forest achieves slightly lower scores. This pattern is consistent with gradient boosting's ability to model complex, non-linear feature interactions and handle mixed feature types effectively, especially when features combine mask-based color statistics, masked descriptors, texture measures, and spatial symptom ratios.

The confusion matrices in Figure 3 provide a detailed view of where the models succeed and fail. They also help confirm whether the models make consistent mistakes across classes.

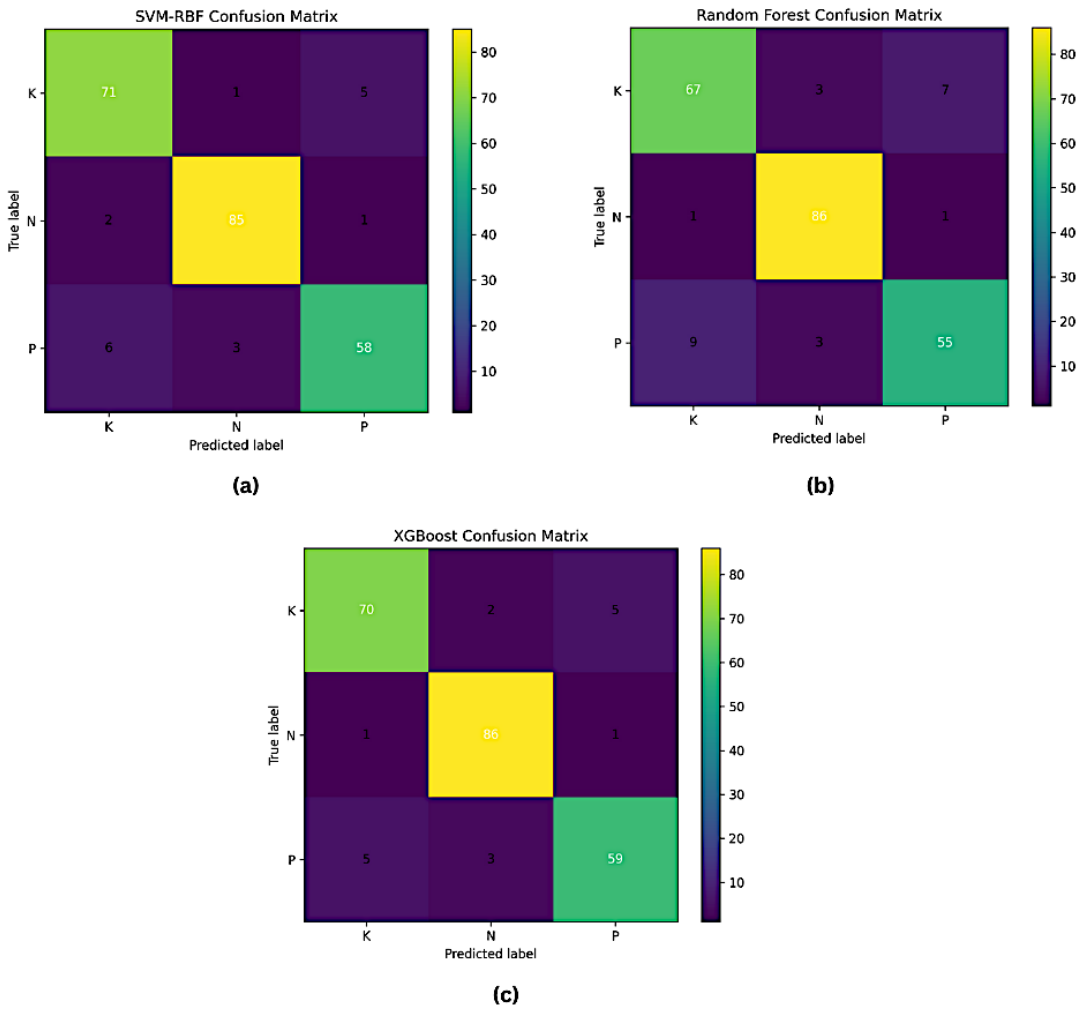

Figure 3. The Confusion Matrix of (a) SVM-RBF, (b) Random Forest, and (c) XGBoost

Across all models, N deficiency is most reliably recognized (85 – 86 correct predictions out of 88 test samples per model). In contrast, K and P show higher confusion. For example, in XGBoost, the main errors occur when K is predicted as P (5 samples) and when P is predicted as K (5 samples). This indicates that symptom similarity between K and P remains a key challenge, even after segmentation and feature engineering.

This confusion is consistent with the agronomic literature, which documents that phosphorus and potassium deficiencies can produce visually overlapping symptoms on rice leaves. Both deficiencies may cause marginal discoloration, browning at leaf tips, and reduced overall greenness, particularly during early to mid-growth stages [1]. These shared visual characteristics make the two classes inherently difficult to separate using color and texture features alone, suggesting that the observed confusion is partly a data-level limitation rather than a purely algorithmic shortcoming.

Although SVM-RBF achieves the highest mean macro-F1 during 5-fold cross-validation, XGBoost provides the best performance on the held-out test set. This difference is not contradictory, as cross-validation assesses stability across multiple splits of the training data. In contrast, the test set evaluates generalization to samples that are completely unseen during tuning. In other words, a model can be consistently strong across folds yet still be slightly less adaptive to the specific distribution shift between the training folds and the final test subset. To make the cross-validation and test behavior more explicit, Table 7 summarizes the macro-F1 achieved during cross-validation (mean ± std) and on the held-out test set, together with their difference.

Table 7. Comparison of Cross-Validation and Test Set Macro-F1

Model	CV F1	Test F1	Gap (Test – CV mean)
SVM-RBF	**0.9391**	0.9187	-0.0203
Random Forest	0.9010	0.8906	-0.0104
XGBoost	0.9133	**0.9233**	+0.0100

Note: Cross-validation values are computed on the training set using 5-fold StratifiedKFold, while test values are computed once on the held-out test set.

Because hyperparameters are selected to maximize cross-validation macro-F1, the model ranking observed in cross-validation may not always match the ranking on the held-out test set, especially when the test subset contains symptom variations that are underrepresented in some folds. A plausible explanation lies in the models' learning mechanisms and the heterogeneous feature design used in this study. SVM-RBF operates as a margin-based classifier in a kernel-induced space and often performs well when class separation can be achieved with a smooth decision boundary. This can lead to strong, stable cross-validation results, especially when feature scaling is consistent. In contrast, XGBoost builds an ensemble of trees that can capture non-linear interactions and conditional patterns across different feature groups, such as mask-based color statistics, masked histograms, vegetation indices, texture descriptors, and spatial symptom ratios. Such interactions may become more important when subtle variations appear in the held-out test set, which can explain why XGBoost generalizes slightly better.

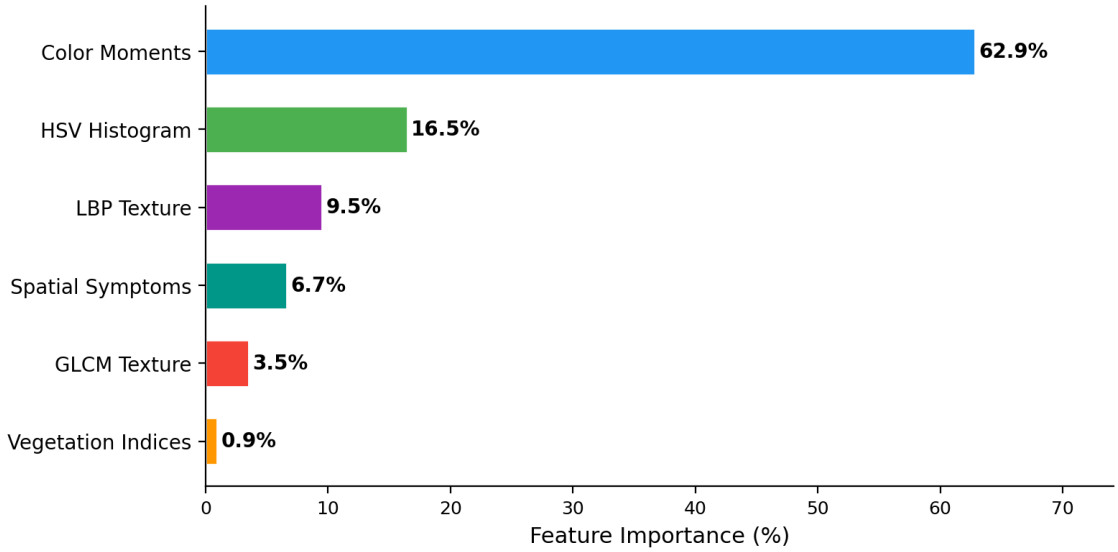

Figure 4. XGBoost Feature Importance by Feature Group

The observed error structure also supports this interpretation. The dominant confusion occurs between K and P across all models, and the test-set advantage of XGBoost suggests that boosting-based trees may better exploit complementary cues across multiple feature groups to reduce this confusion on unseen samples. Therefore, the cross-validation results highlight the stability of SVM-RBF during tuning, whereas the test results highlight XGBoost's stronger generalization with this feature representation.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

To address the second research question regarding which feature families contribute most to class separability, feature importance analysis was conducted on the tuned XGBoost model using split-count-based importance, which reflects how frequently each feature is used for splitting across all trees. The results, summarized in Figure 4, show that Color Moments account for the largest share of importance at 62.85%, followed by HSV Histogram (16.50%), LBP Texture (9.53%), Spatial Symptoms (6.66%), GLCM Texture (3.51%), and Vegetation Indices (0.95%).

It should be noted that split-count-based importance reflects the usefulness of features within the XGBoost decision tree ensemble and does not necessarily generalize to other model architectures, such as SVM-RBF, where feature contributions are mediated through a kernel-induced space. Nevertheless, this analysis provides a useful and interpretable proxy for understanding which feature groups carry the most discriminative information in this dataset.

Among individual features, the mean value of the red channel (RGB Mean R) emerges as the single most informative feature, contributing approximately 54.89% of total split importance. This finding is consistent with the nature of NPK deficiency symptoms in rice leaves, where nitrogen deficiency produces widespread yellowing (chlorosis), phosphorus deficiency causes purplish or dark green discoloration, and potassium deficiency manifests as marginal browning and leaf tip necrosis, all of which involve distinct shifts in the red channel intensity. The dominance of a single-color statistic also explains why Color Moments as a group are far more decisive than texture or spatial features in this dataset.

HSV Histogram features contribute the second largest share (16.50%), with hue and value histogram bins appearing among the top individual features. This is coherent with the HSV-based segmentation strategy employed in this study: because the leaf region is isolated using saturation and value thresholds, the remaining HSV distribution captures genuine leaf color variation rather than background noise. LBP Texture (9.53%) and Spatial Symptoms (6.66%) provide complementary discriminative information, likely supporting the separation of cases where color alone is insufficient, such as the P–K confusion identified in the confusion matrix analysis. By contrast, Vegetation Indices contribute only 0.95% of total importance, suggesting that the six RGB-derived index

means, while conceptually motivated, do not provide additional discriminative power beyond what is already captured by the raw color moment and histogram features in this dataset.

These findings collectively answer the second research question from the perspective of the best-performing model: color-based features, particularly masked color moments, are the dominant contributors to separability among N, P, and K classes within XGBoost, while texture and spatial features play a supporting role. Compared with prior works that also use the Kaggle rice nutrient deficiency dataset, the proposed classical ML pipeline performs competitively.

1) A transfer learning and ensemble study reports that, on the same dataset, the best single-model accuracy is around 90%. The study also shows that ensemble strategies improve average performance across models [43].
2) In addition, a deep ensemble approach reports higher accuracy (up to 98.33%) under an augmented setting with weighted ensembling [2].
3) These comparisons should be interpreted with caution. Even when the same public dataset is used, results may not be directly comparable due to differences in split definition, preprocessing steps, augmentation intensity, and evaluation protocol, such as cross-validation versus a held-out test set. Therefore, the above values are provided only to contextualize the range of reported performance rather than to claim a strict ranking across methods.
4) Importantly, prior research on early-stage rice NPK diagnosis reports that K and P are easily confused, which is consistent with the dominant error pattern observed in this study [1].

Notably, the proposed pipeline achieves 92.67% accuracy and 0.9233 macro-F1 without any data augmentation, using only the original 1,156 images. In contrast, the deep ensemble approach in Talukder, et al. relies on extensive augmentation to expand the effective training set size before reaching its reported 98.33% accuracy [2]. This distinction is significant: augmentation-free pipelines are simpler to maintain, easier to validate, and less prone to introducing artificial distribution shifts that may not generalize to real field conditions. Furthermore, the single-model accuracy of approximately 90% reported in Sharma, et al. is comparable to the Random Forest result in this study (89.65%) [43]. At the same time, the proposed XGBoost pipeline surpasses it by approximately 3

percentage points using only hand-crafted features. These comparisons suggest a clear trade-off. Deep learning ensembles can achieve higher peak accuracy, but the proposed approach offers strong performance at lower computational cost and simpler deployment, which is valuable for practical systems that need fast inference and easier model maintenance.

The main benefit of this study is that it demonstrates a lightweight, explainable pipeline for nutrient-deficiency classification using segmentation-guided, hand-crafted features (126) and tuned classical ML models. This design is suitable for applied scenarios where computing resources are limited, such as:

1) mobile or edge-based decision support for field monitoring,
2) integration with smart farming platforms that already collect visual data,
3) rapid screening to prioritize which plants require closer inspection or follow-up actions.

3.2. Discussion

The overall findings indicate that the proposed feature-engineering and classical machine learning framework is not only accurate, but also practically meaningful for rice leaf nutrient deficiency classification. Among the evaluated models, XGBoost achieved the strongest generalization on the held-out test set, with the highest accuracy (0.9267) and macro-F1 (0.9233), while SVM-RBF showed the strongest average performance during cross-validation (macro-F1 = 0.9391). This pattern suggests that SVM-RBF provides highly stable behavior across the training folds, whereas XGBoost is slightly more adaptable when facing unseen samples in the final evaluation. Such a result is reasonable given the heterogeneous feature representation used in this study. Because the input combines several complementary feature families, including masked color statistics, masked histograms, vegetation-related information, texture descriptors, and spatial symptom ratios, the classification task likely depends on non-linear interactions between visual cues rather than on any single dominant feature type. In this context, the superior test-set performance of XGBoost supports the view that boosting-based ensembles are effective for modeling complex conditional relationships among handcrafted symptom features.

The robustness analysis also strengthens confidence in the validity of the pipeline. When masked color moments were replaced with unmasked color moments computed from the entire image, the performance changes remained very small across all three models, with absolute macro-F1 differences below 1%. This result suggests that the classifiers are not relying heavily on background color cues or obvious acquisition artefacts. Instead, the predictive signal appears to come primarily from symptom-related visual patterns present on the leaf itself. This is an important finding because it reduces the concern that the model may be exploiting shortcut correlations unrelated to nutrient status. Although Random Forest and XGBoost showed a slight numerical advantage under the unmasked setting, the differences were minor and remain within the range of expected variation for a dataset of this size. Therefore, the robustness check supports the conclusion that the extracted features capture biologically meaningful symptom information rather than merely reflecting differences in background appearance.

The confusion matrix analysis provides further insight into the practical strengths and weaknesses of the approach. Across all models, nitrogen deficiency was the easiest class to recognize, with consistently high numbers of correct predictions. This implies that nitrogen deficiency symptoms may produce stronger or more distinctive visual signatures in the extracted feature space, making them easier to separate from the other classes. By contrast, the main classification challenge lies in distinguishing phosphorus (P) and potassium (K) deficiencies. The repeated confusion between these two classes across all three classifiers, particularly in XGBoost where K was often predicted as P and P as K, indicates that their visual manifestations substantially overlap. This finding is consistent with agronomic observations that both nutrient deficiencies can produce similar patterns such as reduced greenness, tip damage, marginal discoloration, and browning. As a result, the remaining classification errors should not be interpreted solely as model weakness; rather, they also reflect an inherent difficulty in the visual discrimination problem itself. Even with segmentation and multi-group feature extraction, some degree of ambiguity is expected when two physiological stress conditions generate closely related leaf symptoms.

Another important contribution of this study is the interpretability of the modeling pipeline. Because the system is based on explicit, symptom-oriented features rather than fully end-to-end image classification, the outputs can be linked more directly to

agronomic reasoning. Features such as color statistics can reflect chlorosis intensity or changes in leaf greenness, masked histograms can capture the distribution of symptomatic discoloration, texture descriptors can encode surface irregularity or patchiness, and spatial symptom ratios can represent how damage is distributed across the leaf area. This structure offers a more transparent basis for decision-making than black-box models in which the relationship between image content and prediction is harder to explain. In an agricultural setting, such transparency is valuable because users may be more willing to trust model predictions when the decision process can be connected to recognizable visual symptoms. It also creates opportunities for future diagnostic tools that do not merely output a class label, but also provide symptom-level explanations that can support extension workers, agronomists, or farmers in interpreting model recommendations.

At the same time, two limitations should be acknowledged. First, the dataset was collected from an online source, which means it may not fully represent the diversity encountered under real field conditions. Variations in illumination, camera angle, camera-to-leaf distance, background clutter, growth stage, and cultivar appearance are likely to be broader in operational environments than in curated internet images. Although the current results are promising, real-world deployment requires caution because performance may decrease when the model is exposed to field images with more heterogeneous acquisition conditions. Second, the segmentation stage depends on fixed HSV thresholds and morphology-based operations, which may become less reliable under extreme illumination, shadowing, or when the leaf color is visually similar to the background. Under such circumstances, imperfect masking could distort downstream features and reduce classification performance. These limitations do not invalidate the present findings, but they do indicate that the current pipeline should be viewed as a strong proof of concept rather than a fully field-ready solution.

Future work can improve both robustness and practical utility while preserving computational efficiency. A first priority is to collect a larger and more diverse field dataset covering different locations, cultivars, growth stages, and image acquisition conditions. Such data would allow stronger evaluation of generalization and help reduce the domain gap between controlled internet images and field deployment. A second direction is to move beyond categorical diagnosis toward severity estimation and mixed-

deficiency recognition, since nutrient stress in practice often appears along a continuum and may involve overlapping deficiencies rather than a single isolated class. A third improvement is to enhance segmentation robustness through methods such as color constancy preprocessing, adaptive thresholding, or other illumination-aware masking strategies. These additions could reduce sensitivity to lighting variation while keeping the pipeline lightweight enough for deployment on mobile or edge-based agricultural systems. Overall, the study demonstrates that interpretable handcrafted features combined with tuned classical machine learning models, especially XGBoost, provide a strong and efficient baseline for rice nutrient deficiency classification, while also highlighting the technical and data-related steps needed to make the approach more reliable in real farming environments.

4. CONCLUSION

This study demonstrates that a lightweight, interpretable classical machine learning pipeline can achieve competitive performance in recognizing rice leaf NPK deficiency on the public Kaggle dataset. By combining HSV-based leaf segmentation with a 126-dimensional hand-crafted feature representation, covering masked color moments, HSV histograms, RGB-based vegetation indices, LBP and GLCM texture descriptors, and spatial symptom ratios, and evaluating under a consistent 80/20 stratified split with 5-fold StratifiedKFold-based RandomizedSearchCV tuning, XGBoost achieves the best generalization on the held-out test set (accuracy 0.9267; macro-F1 0.9233), followed closely by SVM-RBF (accuracy 0.9224; macro-F1 0.9187), while Random Forest performs slightly lower. Feature importance analysis confirms that color moments, particularly the red channel mean, are the dominant contributors to class separability within XGBoost, while texture and spatial features provide complementary discriminative support. The segmentation stage achieves a 0.00% fallback rate, and the masked feature robustness check indicates that background cues do not strongly drive classification results. The dominant remaining challenge is confusion between phosphorus and potassium, confirming that inter-class symptom similarity persists even after leaf-focused preprocessing and multi-cue feature design. The study is limited by its reliance on a public Kaggle dataset that may not fully represent field variability, and by fixed HSV thresholds that may degrade under extreme illumination. Future work should validate the pipeline on diverse field images, extend toward severity or mixed-deficiency recognition,

and improve robustness using adaptive segmentation while maintaining deployment efficiency.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Center for Research and Community Service (P3M), Politeknik Negeri Banjarmasin, for the support and facilitation provided for this research.

REFERENCES

[1] X. Liao and H. Yang, "Diagnosis of early nitrogen, phosphorus and potassium deficiency categories in rice based on multimodal integration and knowledge distillation," *Sci. Rep.*, vol. 15, no. 1, p. 13014, Apr. 2025, doi: 10.1038/s41598-025-97585-0.

[2] M. S. H. Talukder and A. K. Sarkar, "Nutrients deficiency diagnosis of rice crop by weighted average ensemble learning," *Smart Agric. Technol.*, vol. 4, no. 100155, 2023, doi: 10.1016/j.atech.2022.100155.

[3] L. Chen *et al.*, "Identification of nitrogen, phosphorus, and potassium deficiencies in rice based on static scanning technology and hierarchical identification method," *PLoS One*, vol. 9, no. 11, pp. 1–17, 2014, doi: 10.1371/journal.pone.0113200.

[4] A. Y. Reddy and T. R. Balaga, "Enhancing Precision Agriculture Based on Explainable AI for Automated Nutrient Deficiency Diagnosis in Rice Using Attention SqueezeNet," *Ing. des Syst. d'Information*, vol. 30, no. 1, pp. 181–190, 2025, doi: 10.18280/isi.300115.

[5] M. Sobhana, V. R. Sindhuja, V. Tejaswi, and Durgesh, "Deep Ensemble Mobile Application for Recommendation of Fertilizer Based on Nutrient Deficiency in Rice Plants Using Transfer Learning Models," *Int. J. Interact. Mob. Technol.*, vol. 16, no. 16, pp. 100–112, 2022, doi: 10.3991/ijim.v16i16.31497.

[6] N. Ahmad, H. M. S. Asif, G. Saleem, M. U. Younus, S. Anwar, and M. R. Anjum, "Leaf Image-Based Plant Disease Identification Using Color and Texture Features," *Wirel. Pers. Commun.*, vol. 121, no. 2, pp. 1139–1168, 2021, doi: 10.1007/s11277-021-09054-2.

[7] J. Mkhatshwa, T. Kavu, and O. Daramola, "Analysing the Performance and Interpretability of CNN-Based Architectures for Plant Nutrient Deficiency Identification," *Computation*, vol. 12, no. 6, 2024, doi: 10.3390/computation12060113.

[8] M. J. Sulastri, D. R. Sulistyaningrum, and H. Nurhadi, "Detection of Nutrient Deficiency in Rice Plants Based on Leaf Image," *2021 Int. Conf. Adv. Mechatronics, Intell. Manuf. Ind. Autom. ICAMIMIA 2021 - Proceeding*, pp. 143–148, 2021, doi: 10.1109/ICAMIMIA54022.2021.9807811.

[9] S. Supreetha, R. Premalathamma, and M. S. H., "Deep learning techniques to detect nutrient deficiency in rice plants," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Apr. 2024, pp. 699–705.

[10] B. Naresh Kumar and S. Sakthivel, "Rice leaf disease classification using a fusion vision approach," *Sci. Rep.*, vol. 15, no. 1, p. 8692, Mar. 2025, doi: 10.1038/s41598-025-87800-3.

[11] N. Islam and P. Richhariya, "Deep Learning-Based Classification of Rice Leaf Diseases Using Hybrid Ensemble Models," *Int. J. Innov. Sci. Eng. Manag.*, pp. 34–41, Dec. 2024, doi: 10.69968/ijisem.2024v3i434-41.

[12] A. Purnama, E. Fauzi, and B. A. Prasetyo, "Implementing PSO-based Image Segmentation for Detecting Sweet Potato Leaf Disease," *Int. J. Multidiscip. Approach Res. Sci.*, vol. 3, no. 02, pp. 447–457, 2025, doi: 10.59653/ijmars.v3i02.1482.

[13] B. S. Riza, R. Rosnelly, and V. H. S. Edy, "Application of Digital Image Processing for Orchid Image Segmentation in Morphological Plant Analysis," *J. Appl. Sci. Eng. Technol. Educ.*, vol. 7, no. 1, pp. 94–101, 2025, doi: 10.35877/454RI.asci3772.

[14] C. Fang, X. Wang, and Q. Wang, "Adaptive morphology structural element construction algorithm based on local pixel density and symmetry," *Multimed. Tools Appl.*, vol. 82, no. 1, pp. 195–215, 2023, doi: 10.1007/s11042-022-13259-3.

[15] J. Yan, K. Liang, C. Liu, and M. Gao, "Citrus recognition in orchard scene based on modified HSV-morphology method," *Appl. Comput. Eng.*, vol. 88, no. 1, pp. 69–76, 2024, doi: 10.54254/2755-2721/88/20241637.

[16] T. Xu, L. Yao, L. Xu, Q. Chen, and Z. Yang, "Image Segmentation of Cucumber Seedlings Based on Genetic Algorithm," *Sustainability*, vol. 15, no. 4, 2023, doi: 10.3390/su15043089.

[17] M. A. Hutchison and C. M. Koepferl, "Contour Analysis Tool: An Interactive Tool for Background and Morphology Analysis," *Astrophys. J.*, vol. 975, no. 1, p. 27, 2024, doi: 10.3847/1538-4357/ad779f.

[18] S. Zhiwei *et al.*, "Image-based Pretreatment Study of Rice Blast Disease," *J. Adv. Appl. Sci. Res.*, vol. 6, no. 3, pp. 1–15, 2024, doi: 10.46947/joaasr632024945.

[19] H. Rezvan, M. J. Valadan Zoej, F. Youssefi, and E. Ghaderpour, "Automated Rice Seedling Segmentation and Unsupervised Health Assessment Using Segment Anything Model with Multi-Modal Feature Analysis," *Sensors*, vol. 25, no. 17, 2025, doi: 10.3390/s25175546.

[20] M. Keskar and D. D. Maktedar, "Hybrid deep-spatio textural feature model for medicinal plant disease classification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 30, no. 1, pp. 356–365, 2023, doi: 10.11591/ijeecs.v30.i1.pp356-365.

[21] M. J. Hoque, M. S. Islam, and M. Khaliluzzaman, "AI-Powered Precision in Diagnosing Tomato Leaf Diseases," *Complexity*, vol. 2025, no. 1, Jan. 2025, doi: 10.1155/cplx/7838841.

[22] R. S. Krishnan and E. G. Julie, "Computer aided detection of leaf disease in agriculture using convolution neural network based squeeze and excitation network," *Automatika*, vol. 64, no. 4, pp. 1038–1053, 2023, doi: 10.1080/00051144.2023.2241792.

[23] B. Luna-Benoso, J. C. Martínez-Perales, J. Cortés-Galicia, R. Flores-Carapia, and V. M. Silva-García, "Detection of diseases in tomato leaves by color analysis," *Electron.*, vol. 10, no. 9, pp. 1–16, 2021, doi: 10.3390/electronics10091055.

[24] N. S. B. Mat Said, H. Madzin, S. K. Ali, and N. S. Beng, "Comparison of color-based feature extraction methods in banana leaf diseases classification using SVM and K-NN," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 24, no. 3, pp. 1523–1533, 2021, doi: 10.11591/ijeecs.v24.i3.pp1523-1533.

[25] B. Yue *et al.*, "Research on SPAD Inversion of Rice Leaves at a Field Scale Based on Machine Vision and Leaf Segmentation Techniques," *Agric.*, vol. 15, no. 12, pp. 1–24, 2025, doi: 10.3390/agriculture15121270.

[26] P. Pradhan, B. Kumar, K. Kumar, and R. Bhutiani, "Plant leaf disease detection using local binary pattern and deep convolutional neural networks," *Environ. Conserv. J.*, vol. 26, no. 1, pp. 66–78, 2025, doi: 10.36953/ECJ.29292943.

[27] J. D. Thiagarajan *et al.*, "Analysis of banana plant health using machine learning techniques," *Sci. Rep.*, vol. 14, no. 1, pp. 1–23, 2024, doi: 10.1038/s41598-024-63930-y.

[28] C. Nyasulu *et al.*, "A comparative study of Machine Learning-based classification of Tomato fungal diseases: Application of GLCM texture features," *Heliyon*, vol. 9, no. 11, p. e21697, 2023, doi: 10.1016/j.heliyon.2023.e21697.

[29] Z. Hou *et al.*, "Selection of Optimal Diagnostic Positions for Early Nutrient Deficiency in Cucumber Leaves Based on Spatial Distribution of Raman Spectra," *Plants*, vol. 14, no. 8, pp. 1–17, 2025, doi: 10.3390/plants14081199.

[30] D. Ma, L. Wang, L. Zhang, Z. Song, T. U. Rehman, and J. Jin, "Stress distribution analysis on hyperspectral corn leaf images for improved phenotyping quality," *Sensors*, vol. 20, no. 13, pp. 1–13, 2020, doi: 10.3390/s20133659.

[31] M. Saberi Anari, "A Hybrid Model for Leaf Diseases Classification Based on the Modified Deep Transfer Learning and Ensemble Approach for Agricultural AIoT-Based Monitoring," *Comput. Intell. Neurosci.*, vol. 2022, no. 1, 2022, doi: 10.1155/2022/6504616.

[32] F. Budiman and E. Sugiarto, "Non-linear Multiclass SVM Classification Optimization using Large Datasets of Geometric Motif Image," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 9, pp. 284–290, 2021, doi: 10.14569/IJACSA.2021.0120932.

[33] Dhiyaussalam, A. Wibowo, F. A. Nugroho, E. A. Sarwoko, and I. M. A. Setiawan, "Classification of Headache Disorder Using Random Forest Algorithm," in *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, IEEE, Nov. 2020, pp. 1–5. doi: 10.1109/ICICoS51170.2020.9299105.

[34] Dhiyaussalam and S. Uyun, "Optimization of Random Forest Hyperparameters with Genetic Algorithm in Classification of Lung Cancer," in *2023 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2023, pp. 82–88. doi: 10.1109/ISRITI60336.2023.10467686.

[35] D. Boldini, F. Grisoni, D. Kuhn, L. Friedrich, and S. A. Sieber, "Practical guidelines for the use of gradient boosting for molecular property prediction," *J. Cheminform.*, vol. 15, no. 1, pp. 1–13, 2023, doi: 10.1186/s13321-023-00743-7.

[36] A. Eley, T. T. Hlaing, D. Breininger, Z. Helforoush, and N. N. Kachouie, "Monte Carlo Gradient Boosted Trees for Cancer Staging: A Machine Learning Approach," *Cancers (Basel)*, vol. 17, no. 15, pp. 1–27, 2025, doi: 10.3390/cancers17152452.

[37] J. Edin *et al.*, "Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study," *SIGIR 2023 - Proc. 46th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 2572–2582, 2023, doi: 10.1145/3539618.3591918.

[38] Y. Liu *et al.*, "Development and external validation of machine learning models for the early prediction of malnutrition in critically ill patients: a prospective observational study," *BMC Med. Inform. Decis. Mak.*, vol. 25, no. 1, 2025, doi: 10.1186/s12911-025-03082-9.

[39] C. Chen *et al*, "Application of GA-WELM Model Based on Stratified Cross-Validation in Intrusion Detection," *Symmetry (Basel).*, vol. 15, no. 9, pp. 1–14, 2023, doi: 10.3390/sym15091719.

[40] V. Teodorescu and L. Obreja Brașoveanu, "Assessing the Validity of k-Fold Cross-Validation for Model Selection: Evidence from Bankruptcy Prediction Using Random Forest and XGBoost," *Computation*, vol. 13, no. 5, 2025, doi: 10.3390/computation13050127.

[41] O. M. Alyasiri and Y. N. Cheah, "Multi-Class Text Classification using Machine Learning Techniques," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 3, pp. 22598–22604, 2025, doi: 10.48084/etasr.9994.

[42] M. Amudha and K. Brindha, "Rice Leaf Nutrient Deficiency Classification System Using CAR-Capsule Network," *IEEE Access*, vol. 12, pp. 169518–169532, 2024, doi: 10.1109/ACCESS.2024.3498606.

[43] M. Sharma, K. Nath, R. K. Sharma, C. J. Kumar, and A. Chaudhary, "Ensemble Averaging of Transfer Learning Models for Identification of Nutritional Deficiency in Rice Plant," *Electron.*, vol. 11, no. 1, 2022, doi: 10.3390/electronics11010148.