Zahra: Bulletin Big data, Data Science and Artificial Intelligence
Vol. 3, No. 1, [June 2024], pp. [43–55]
E-ISSN 2830-7984

Research Article Open Access (CC–BY-SA)

Particle Swarm Optimization of Handcrafted Color and Texture Features for Random Forest–Based Plant Disease Classification

Muhammad Syahrizal[1*], Agus Perdana Windarto[2], Mesran[3], Siti Nur Nazhifah Binti Jais Meah[4]
[1] *Computer Graphics Engineering Technology Program, Politeknik Cendana, Indonesia*
[2] *Department of Informatics, Master's Program, STIKOM Tunas Bangsa, Pematangsiantar, Indonesia*
[3] *Management Study Program, Sekolah Tinggi Ilmu Manajemen Sukma, Medan, Indonesia*
[4] *Faculty of Intelligent Computing, Universiti Malaysia Perlis (UniMAP), Malaysia*

[1] *m.syahrizal156@gmail.com;* [2] *agus.perdana@amiktunasbangsa.ac.id;* [3] *mesran.skom.mkom@gmail.com;* [4] *sitinurnazhifahjm@gmail.com*
* *Corresponding author*

Abstract

Visible plant diseases can alter leaf color and texture, motivating lightweight image classifiers for agricultural screening. This study examined whether Particle Swarm Optimization (PSO) of handcrafted color–texture descriptors and Random Forest hyperparameters improves multiclass disease recognition while reducing the number of input features. A pilot drawn from the public PlantDisease collection comprised 1,800 unique images from 15 pepper, potato, and tomato classes, with 1,260 training, 270 development, and 270 reserved test images. Fifty-five image descriptors included 30 RGB/HSV color statistics and 25 GLCM, local binary pattern, and Sobel texture measurements. Six configurations were compared using two-fold training cross-validation, six particles, four PSO iterations, and common development metrics. The default Random Forest achieved development macro-F1 of 0.7950, whereas joint PSO feature-and-parameter optimization obtained 0.7508 with 25 selected descriptors, a 54.55% reduction. The joint method exceeded random joint search (0.7302 macro-F1) but trailed standard parameter tuning (0.7888), indicating a feature-efficiency trade-off rather than a reliable improvement in recognition. The final test set remained locked, so the reported values are exploratory development findings and do not establish independent test superiority or field generalization.

Keywords: Plant leaf disease; Random Forest; Particle swarm optimization; Handcrafted color–texture descriptors; Feature selection

1. Introduction

Image-based plant disease recognition can complement visual inspection when symptoms appear as changes in leaf color and texture [1]. A feature-engineering review identifies color moments, texture statistics, and segmentation as interpretable sources of disease-relevant measurements [2]. Comparative crop experiments show that conventional machine-learning models remain useful alternatives to large neural networks [3]. Broader surveys describe persistent variation in datasets, target classes, and reported evaluation protocols [4]. A recent systematic review emphasizes the importance of separating classification performance from the practical conditions under which images are captured [5]. Agricultural computer-vision research increasingly considers robustness, model complexity, and data accessibility alongside predictive scores [6]. Dataset surveys show that controlled leaf photographs may reward background cues that are absent in field environments [7]. Domain-generalization experiments confirm that differences in illumination, image style, and surrounding vegetation can disrupt learned disease patterns [8]. Image augmentation is widely studied to address some of these variations, although the supporting evidence often remains concentrated in laboratory images [9]. A concise, measurable baseline therefore remains necessary before attributing a classification gain to descriptor selection or optimizer design.

Handcrafted descriptors can summarize RGB and HSV color variation together with gray-level co-occurrence and local binary pattern textures [10]. Swarm-driven feature selection has been explored as a way to remove redundant descriptors while retaining diagnostically useful variation [11]. A recent rice-disease study evaluated

particle-swarm and ant-colony search with Random Forest and support vector classifiers, illustrating the relevance of optimizer-specific comparisons [12]. Particle-swarm feature selection remains sensitive to the initialization of candidate subsets and the evaluation function [13]. Explicit, adaptive particle representations have been developed to improve subset search in high-dimensional classification [14]. Multi-objective PSO algorithms also attempt to balance predictive performance and the number of retained features [15]. Factorial analysis has shown that PSO selection performance depends on search parameters rather than optimizer labels alone [16]. Probe-based PSO variants further demonstrate that the chosen exploration mechanism can affect feature discovery [17]. Tomato-leaf work employing selected features illustrates the broader search for compact disease classifiers [18]. These findings motivate separate tests of feature selection, forest tuning, and their combination rather than treating the full pipeline as automatically superior.

The present investigation evaluates particle swarm optimization of handcrafted color and texture features for Random Forest-based classification of pepper, potato, and tomato leaf conditions. The evaluated pipeline comprises 30 color and 25 texture descriptors extracted from resized leaf images. PlantVillage-type imagery offers a controlled benchmark, yet field evaluations indicate that environmental variation can lower performance substantially [19]. Research on transfer across plant-disease image domains reinforces the need to distinguish laboratory discrimination from field generalization [20]. New in-the-wild datasets seek to represent the complex backgrounds and disease appearances missing from uniform-image benchmarks [21]. Resource-efficient disease classification also motivates examining feature reduction without assuming an accuracy benefit [22]. For the present pilot, the primary descriptive outcome is macro-averaged F1 on a development partition, supplemented by accuracy, balanced accuracy, per-class diagnostics, feature count, and fitting time. The comparisons comprise an unoptimized forest, PSO feature selection, PSO hyperparameter tuning, joint PSO, random joint search, and conventional random tuning. The central hypothesis is whether joint PSO can improve multiclass recognition or at least offer a defensible sparsity–performance trade-off. The observed evidence, including any unfavorable result, is evaluated without redefining the study objective. Recent explainable and federated agriculture research further distinguishes benchmark scores from accountable deployment settings [23].

2. Method

2.1. Related Work

Plant disease classifiers span traditional color–texture features, ensemble learning, and deep representations [24]. Texture-based diagnostic approaches emphasize that local visual structure can complement color distributions in leaf images [1]. Comparative crop research identifies Random Forest as a relevant baseline under some experimental conditions [3]. Broader machine-learning studies report that model-family rankings vary by crop, image source, and validation method [25]. CNN-based recognition systems provide strong alternatives but generally introduce different computational and data requirements [26]. Plant-disease reviews therefore caution against inferring broad superiority from one classification benchmark [27]. A recent systematic review synthesizes the breadth of deep-learning evidence for disease identification [28].

Feature-selection research provides the immediate methodological context for the proposed PSO–Random Forest combination. The salp-swarm literature demonstrates that irrelevant handcrafted descriptors can hinder disease classification and that subset selection may reduce this burden [11]. A hybrid PSO–ACO study offers a closer domain-specific reference because it examines optimization with classical plant-leaf classifiers [12]. Research on particle initialization shows that search diversity and candidate representation can influence the discovered subset [13]. Alternative PSO representations explicitly account for variable subset size during optimization [14]. General hyperparameter-optimization guidance recommends controlling search spaces, budgets, and evaluation boundaries when comparing optimizers [29]. Swarm-tuned neural architectures offer complementary optimization evidence without matching the present forest protocol [30]. Additional rice-focused swarm research highlights algorithm-dependent trade-offs [31].

Image acquisition also affects what can be learned from handcrafted descriptors. Recent reviews of plant-disease datasets identify background and illumination differences as important sources of deployment error [7]. Laboratory-to-field evaluation confirms that a classifier trained under controlled conditions may lose discrimination when image style changes [8]. Field-oriented studies similarly argue for realistic crop imagery and independent evaluation beyond internal partitions [19]. Few-shot plant-disease research addresses a related problem in which limited representative observations constrain generalization [32]. Explainability work examines whether image classifiers attend to disease-relevant visual regions rather than incidental objects [33]. The present experiment does not contain

field images or source-group identifiers, so robustness outside the benchmark cannot be inferred. A review of diagnosis and image-driven management systems broadens the context but does not resolve the present validation limitations [34].

Optimization objectives must reflect the metric that the study intends to improve. Large-scale PSO research evaluates several trade-offs between classification score and retained subset size [17]. Other swarm methods have achieved compact feature sets in plant disease recognition, although their outcomes remain method- and dataset-dependent [11]. Work on decision-probability calibration distinguishes good classification from reliable probability estimates [35]. Augmentation reviews illustrate additional model-development strategies that do not substitute for leakage-safe testing [9]. A recent computationally efficient classifier study reinforces the importance of reporting model complexity alongside accuracy [22]. Table 1 consolidates the closest contributions and the specific gaps that remain relevant to the present PSO–Random Forest evaluation.

Table 1. Comparative synthesis of selected related studies

Author / year	Contribution	Research-specific limitation
Patil and Mandlik (2024)	Integrated handcrafted color and texture for leaf disease classification	Does not isolate PSO feature-subset and RF-parameter effects
Xie et al. (2023)	Swarm feature selection for image-based plant disease detection	Uses a different swarm optimizer and evaluation setting
Goyal et al. (2025)	Compared conventional crop-disease classifiers including Random Forest	Does not use the same 15-class handcrafted feature protocol
Kurniawan et al. (2025)	Examined PSO/ACO with classical rice-leaf classifiers	Different crop, features and validation design
Xue et al. (2023)	Investigated PSO feature initialization for classification	General datasets rather than plant disease images
Xu et al. (2024)	Analyzed benchmark-to-field generalization limitations	Does not quantify the present PSO development trade-off

The evidence in Table 1 locates the study among recent color–texture and metaheuristic classification approaches. The color–texture literature supports the chosen handcrafted representation but does not establish the benefit of PSO for these 15 classes. The comparative Random Forest study provides a plausible conventional baseline without matching the present data partition. The swarm-selection papers supply optimization precedents but use different feature sets and training budgets. The dataset work identifies external validity issues that cannot be resolved by an internal development split. The remaining research question concerns whether PSO offers a measurable gain over an unoptimized forest and standard search under the observed pilot conditions.

2.2. Dataset

The experiment used the public PlantDisease image collection associated with the approved study title (https://www.kaggle.com/datasets/emmarex/plantdisease). The recorded pilot audit selected 120 images from each of 15 available leaf-condition classes. The selected categories comprise pepper, potato, and tomato healthy or diseased conditions, yielding 1,800 distinct images after an exact decoded-image hash audit. No images failed loading, and no conflicting labels were detected among duplicate hashes. The source images were resized to 128 × 128 pixels before descriptor extraction. Color features summarize normalized RGB and HSV channels, whereas texture features derive from GLCM statistics, local binary patterns, and Sobel edge magnitudes. Table 2 specifies the source, label structure, and implemented sampling constraints.

Table 2. Planned dataset specification and audit requirements

Dataset attribute	Executed pilot specification
Public source	https://www.kaggle.com/datasets/emmarex/plantdisease
Target	15 image-folder labels for pepper, potato and tomato conditions
Analysis unit	Decoded leaf image; 1,800 unique image hashes after audit
Subset profile	120 selected images per class; seed 42; 128 × 128-pixel resize
Input representation	55 handcrafted variables (30 RGB/HSV color; 25 GLCM/LBP/Sobel texture)
Source limitation	No plant/source-level grouping or independent field-image validation

The evidence in Table 2 establishes that the analyzed sample is a balanced 15-class pilot rather than the complete image repository. Exact image hashes were used to remove duplicated decoded images before splitting. This check does not establish that visually similar photographs of the same plant were separated. The initial 55 descriptors contain 30 color and 25 texture variables. Class balance makes aggregate accuracy easier to interpret but does not remove the need for per-class recognition measures. Generalization to unobserved farms, lighting conditions, and devices remains outside the available evidence.

The experiment separates fitting and diagnosis through a fixed stratified split established before model search. The training portion contains 1,260 images, and the development portion contains 270 images. A separate set of 270 images is reserved for a final test, which the supplied notebook does not execute. All optimization fitness calculations use training-side two-fold cross-validation. The development split is used only for the recorded diagnostic comparisons after models have been fitted. Table 3 distinguishes the operations that ran from the final-test evaluation that remains unexecuted.

Table 3. Experimental data flow and leakage-prevention controls

Stage	Executed handling
A. Image audit	Decoded RGB image hashes; 1,800 unique selected images across 15 classes
B. Partition	Stratified 1,260 train / 270 development / 270 reserved test; seed 42
C. Search	Training-only two-fold CV; PSO 6 particles × 4 iterations; fitness penalty 0.01
D. Development	Fit selected forest on training data and report development predictions
E. Final test	Reserved test set NOT evaluated; RUN_FINAL_TEST = False

The evidence in Table 3 specifies the actual information boundary used for the reported results. The split proportions are 70%, 15%, and 15%, respectively. Candidate models were chosen through two-fold cross-validation on the 1,260 training examples. Development images supplied diagnostic scores for six recorded classification configurations. The 270 reserved test images were not used for a final comparison in this notebook. The scientific interpretation must therefore remain at pilot-development level until a preregistered final evaluation is executed.

2.3. Research Framework

The evidence in Figure 1 displays the original baseline-versus-proposed framework in the unchanged journal diagram. The baseline extracts the same handcrafted descriptors and fits a default Random Forest. The experimental branch adds PSO-based descriptor selection and parameter tuning before forest fitting. Both branches operate on the same selected images and fixed split. This controlled structure makes the baseline a necessary reference for any optimization claim. The executed notebook additionally evaluates the separate feature-only and tuning-only components and two conventional search controls.

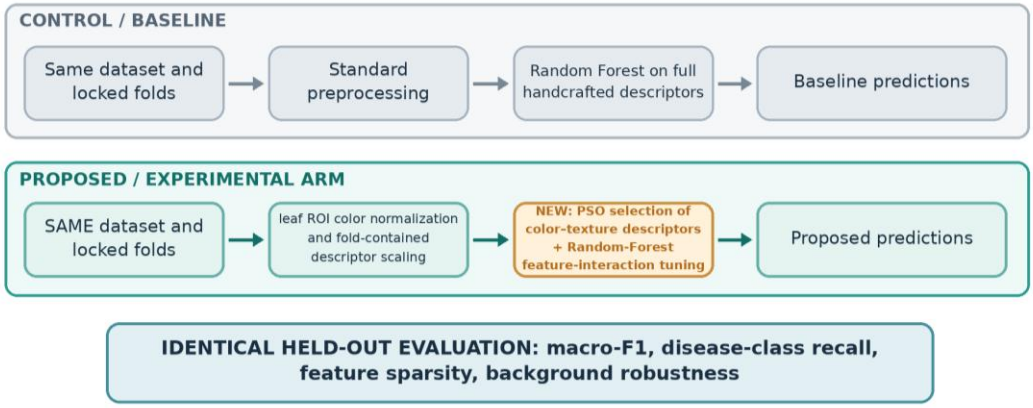

Figure 1. Controlled baseline-versus-proposed research framework

The evidence in Figure 1 separates the common image-processing pipeline from the proposed PSO interventions. The actual descriptor set comprises RGB/HSV color measurements and GLCM/LBP/Sobel texture measurements. The baseline retains all 55 input variables while the joint PSO branch selects 25. The diagram identifies a methodological comparison, not an established improvement in classification performance. The same development records underpin the recorded diagnostic metrics. The reserved test set cannot be represented as an evaluated holdout merely because it appears in the framework.

The design includes two isolated PSO interventions and their joint combination to distinguish potential component effects. PSO A searches subsets of the 55 descriptors with baseline forest settings. PSO B searches tree count, maximum depth, and minimum leaf size while retaining every descriptor. PSO A+B searches the subset and the three forest parameters together. A conventional random joint search uses an equal count of candidate requests. Table 4 defines these contrasts and the primary descriptive comparison.

Table 4. Baseline, proposed intervention and falsification design

Component	Executed comparison
Baseline	200-tree Random Forest; 55 descriptors; balanced_subsample; sqrt feature sampling
PSO A	Binary descriptor mask; default RF hyperparameters
PSO B	55 descriptors; tune trees, max_depth and min_samples_leaf
PSO A+B	Joint 55-bit mask and three RF hyperparameters
Controls	Random joint search and standard random hyperparameter search (24 requests each)

The evidence in Table 4 distinguishes the influence of feature selection from the influence of forest tuning. This separation is necessary because changing both components can obscure which intervention accounts for a performance difference. The 24 candidate requests in each pilot search represent a very small search budget. Search time varies by branch and is not identical to the final forest fit time. The random-search controls are relevant but do not constitute repeated independent experiments. The subsequent results use development macro-F1 as the central outcome without claiming an evaluated final-test advantage.

2.4. Proposed Method

The evidence in Figure 2 preserves the proposed method architecture supplied in the journal template. Each leaf image is represented by fixed-length color and texture descriptors. The particle position is decoded into a binary feature mask, with an all-feature fallback to prevent empty input sets. The tuning branch maps three continuous coordinates into Random Forest hyperparameters. Cross-validated macro-F1, reduced by a small feature-count penalty, determines particle fitness. The method is evaluated through a pilot implementation rather than an exhaustive or field-validated experiment.

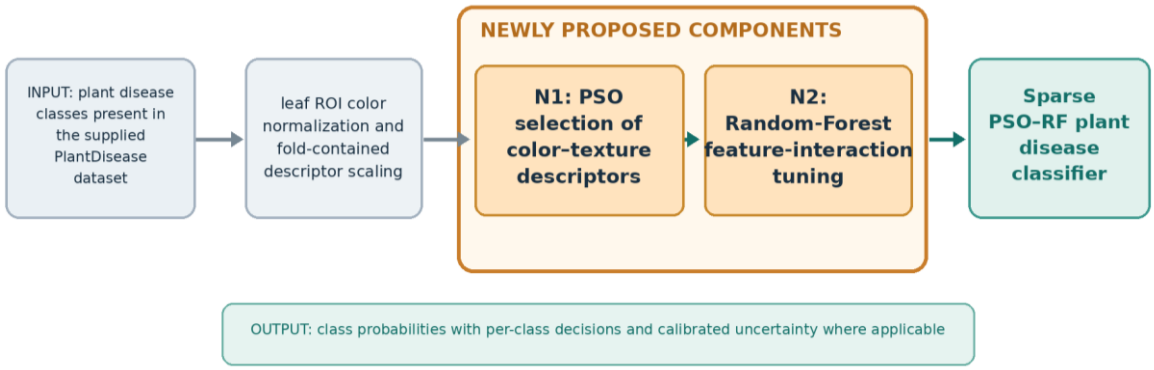

Figure 2. Proposed method architecture with explicitly highlighted novel components

The evidence in Figure 2 identifies two algorithmic interventions: feature-subset optimization and forest hyperparameter tuning. In the executed search, PSO uses an inertia coefficient of 0.72 and cognitive and social coefficients of 1.49. Particle velocities are bounded between −0.2 and 0.2, and positions are bounded between zero and one. A coordinate value of at least 0.5 selects a descriptor. The hybrid candidate retains 25 features and uses 242 trees, maximum depth 16, and minimum leaf size two. These implementation details are evidenced by the notebook and do not establish that the hybrid generalizes better than the baseline.

The selection objective must capture both predictive quality and descriptor sparsity without allowing the held-out test labels to affect optimization. The actual notebook computes two-fold macro-F1 on training-side cross-validation. It then subtracts 0.01 times the proportion of retained features. Forest fitting during candidate search uses no more than 70 trees, even when a larger forest is selected for the final training fit. The chosen candidate is subsequently refitted with its decoded tree count for development assessment. Table 5 records these settings in place of the unexecuted prospective formula.

Table 5. Proposed methodological components and verification rules

Design element	Executed operational definition
Input representation	55 handcrafted RGB/HSV, GLCM, LBP and Sobel features
Component A	PSO binary feature mask, selected by position ≥ 0.5
Component B	PSO tuning of n_estimators, max_depth, min_samples_leaf
Output model	Random Forest for 15 image-folder disease/health classes
Training objective	Mean two-fold macro-F1 − 0.01 × (number of selected features / 55)
Pilot budget	Six particles × four iterations; two CV folds; search trees capped at 70
Reported evaluation	Development macro-F1, accuracy, weighted F1, balanced accuracy, features and fit time

The evidence in Table 5 presents the actual scalarized search criterion rather than an unspecified theoretical penalty. The feature penalty coefficient equals 0.01 in the supplied execution. The optimization budget contains six particles and four iterations for each PSO branch. A two-fold stratified validation estimates candidate macro-F1 within the training partition. The final selected forest can contain more trees than the reduced-budget candidate used during search. The distinction between search-stage fitting and development-stage refitting is necessary for reproducible interpretation.

Implementation followed a fixed seed of 42, with branch-specific offsets for the PSO and random-search initializations. The RGB and HSV channels contributed 30 statistical color descriptors. Texture extraction contributed 12 GLCM summary variables, 11 local binary pattern bins, and two Sobel edge summaries. Candidate Random Forest models used balanced-subsample class weights and square-root feature sampling. The selected joint candidate contained 25 features, a 54.55% reduction from the original 55 variables. Equation (1) defines the combined feature representation, and Equations (2)–(5) formalize the implemented macro-F1 objective and PSO updates. The mathematical specifications are derived from the notebook rather than assuming unrecorded optimizer components.

$$\varphi(I) = [\varphi_RGB/HSV(I)_{1:30}, \varphi_GLCM/LBP/Sobel(I)_{1:25}] \in \mathbb{R}^{55} \quad (1)$$

$$m_j = \mathbb{1}[x_j \geq 0.5]; \ \|m\|_0 = \Sigma_j m_j \quad (2)$$

$$F_1, macro = (1/15)\, \Sigma^{c}_{=1}{}^{15} [2 \cdot Precision^c \cdot Recall^c / (Precision^c + Recall^c)] \quad (3)$$

$$J(m, \theta) = (1/2)\, \Sigma^{k}_{=1}{}^{2} F_1, macro\hat{}(k)(RF\theta, m) - 0.01 \cdot \|m\|_0/55 \quad (4)$$

$$v_i\hat{}(t+1) = clip[0.72 v_i^t + 1.49 r_1(p_i - x_i^t) + 1.49 r_2(g - x_i^t), -0.2, 0.2] \quad (5)$$

$$x_i\hat{}(t+1) = clip[x_i^t + v_i\hat{}(t+1), 0, 1] \quad (6)$$

$$Reduction\ (\%) = 100(55 - 25)/55 = 54.55\% \quad (7)$$

Equations (1) and (2) describe the fixed feature vector and the binary mask used by the search. Equation (3) defines the macro-F1 score across the 15 evaluation classes. Equation (4) gives the exact feature-count penalty and two-fold fitness average implemented in the pilot. Equations (5) and (6) express the recorded PSO velocity and position updates with their corresponding bounds. Equation (7) quantifies the observed reduction in the hybrid descriptor set. These mathematical definitions do not imply that the final test was run or that additional calibration or background robustness experiments were performed.

3. Results and Discussion

The recorded experiment compared six classifier configurations on the same 270-image development partition. All methods used the 55-dimensional source descriptor representation or a subset of it. Macro-F1 was the principal descriptive measure of balanced multiclass recognition. Accuracy, weighted F1, balanced accuracy, feature count, and measured forest fitting time were also retained. Only the training data were available for candidate fitness estimation, and the final test remained locked. The results below therefore describe preliminary development evidence rather than an independent evaluation of the optimized classifier.

3.1. Dataset Profile and Development Protocol

The evidence in Figure 3 displays the input-class distribution and the train, development, and reserved-test partitions recorded in the notebook. The 15 classes were sampled at 120 images each to create a balanced pilot. The image-hash audit removed exact duplicates before the stratified split. The training partition contains 1,260 images.

The development and reserved-test partitions each contain 270 images. The plot documents the analyzed profile rather than the size or variability of the complete public repository.

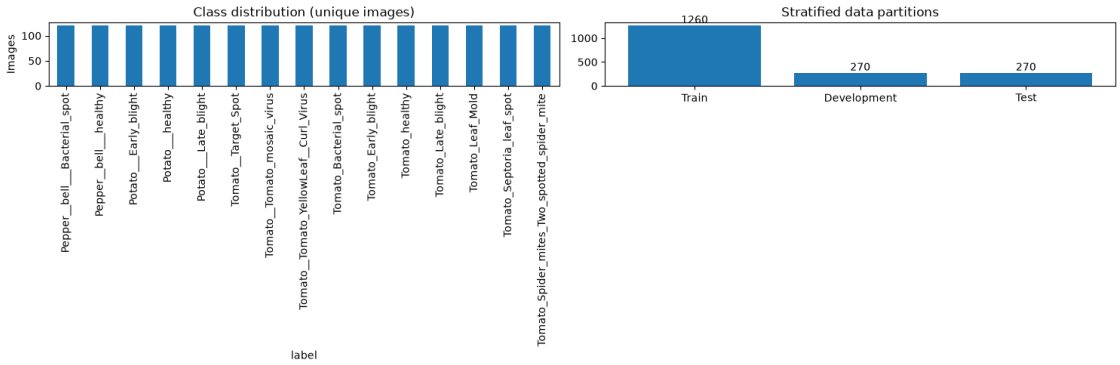

Figure 3. Image-class distribution and fixed stratified partitions

The evidence in Figure 3 confirms equal selected class counts and a 70/15/15 partition of the 1,800 unique images. This balance reduces the extent to which majority-class prevalence can dominate aggregate development accuracy. It does not ensure that samples from the same original plant occur in only one partition. Exact hash filtering captures identical decoded images but not near duplicates or alternative photographs of a plant. No explicit acquisition-domain labels were available for group separation. The subsequent analysis therefore concerns the selected balanced pilot rather than independent agricultural field data.

3.2. Baseline and PSO Model Comparison

The evidence in Table 6 reports the original development metrics for the default Random Forest and five search-based alternatives. All results concern the same 270 development images. Macro-F1 is prioritized because each of the 15 classes contributes equally to its average. Balanced accuracy and overall accuracy are included to assess different aggregations of class recognition. Feature count measures representation sparsity, while fitting time describes the selected model fit and not total optimization time. The proposed joint PSO arm is compared with the default model and both random-search controls.

Table 6. Development-stage baseline and optimization comparison

Method	Accuracy	Macro-F1	Balanced acc.	Features	Fit (s)
Baseline RF	0.7926	0.7950	0.7926	55	0.8915
PSO A (features)	0.7926	0.7934	0.7926	54	0.8549
PSO B (tuning)	0.7741	0.7732	0.7741	55	0.8538
PSO A+B (joint)	0.7519	0.7508	0.7519	25	1.1075
Random joint	0.7333	0.7302	0.7333	22	1.0452
Standard tuning	0.7889	0.7888	0.7889	55	0.9851

The evidence in Table 6 identifies the default Random Forest as the highest-scoring method on development macro-F1 (0.7950). Feature-only PSO nearly matches that score (0.7934), but removes only one of 55 variables. PSO hyperparameter tuning lowers macro-F1 to 0.7732 with all original features retained. Joint PSO lowers it further to 0.7508 while reducing the feature count to 25. The conventional random joint search obtains 0.7302 with 22 variables, and standard parameter tuning obtains 0.7888 with 55. The hybrid thus improves over its random-joint search control but not over the unoptimized forest or standard tuning. The differences are observed development outcomes, not final-test estimates.

The evidence in Figure 4 summarizes the comparative development metrics in the notebook for the six candidate models. The plot includes accuracy, macro-F1, weighted F1, and balanced accuracy. Its bars represent performance on the same development partition. The default forest appears as the strongest macro-F1 configuration. The proposed joint method retains a lower level across the displayed predictive indicators. These comparisons are descriptive because no independent final-test evaluation was executed.

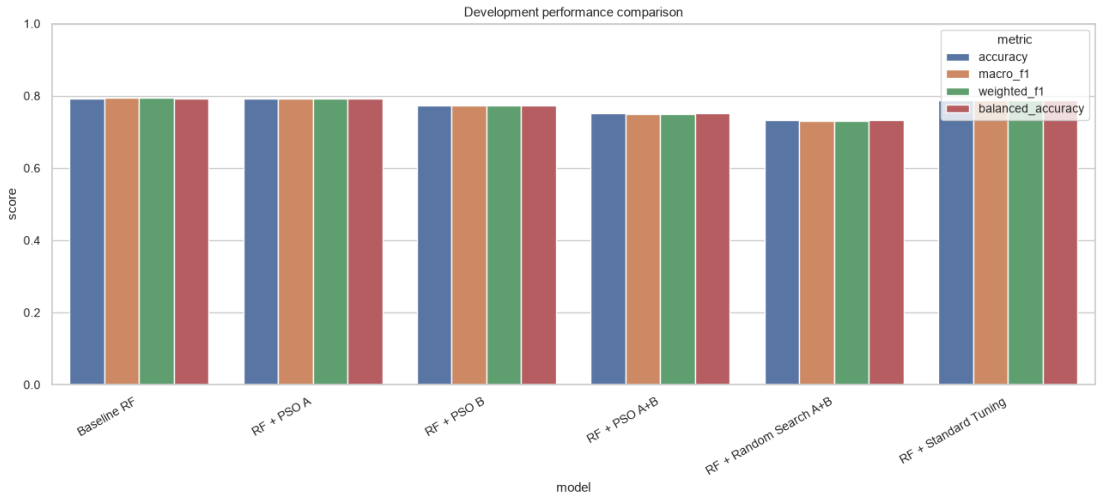

Figure 4. Development metric comparison across six classifiers

The evidence in Figure 4 makes the ranking of the candidate models visible without relying on one metric alone. The baseline records macro-F1 0.7950 and accuracy 0.7926. Joint PSO records macro-F1 0.7508 and accuracy 0.7519. Standard tuning comes closer to the baseline than either the joint PSO or random joint search. Weighted F1 and balanced accuracy follow similar broad patterns in this balanced subset. The joint optimizer therefore does not yield a hidden advantage in the reported aggregate classification indicators. The evidence in Figure 5 compares the number of retained descriptors and measured selected-model fitting times. All six methods use the same starting pool of 55 handcrafted variables. The joint PSO model retains 25 variables and the random joint model retains 22. The feature-only PSO approach retains 54 variables, indicating little compression in that branch. Fitting durations range approximately from 0.85 to 1.11 seconds in the recorded environment. These timings exclude the complete computational search budget and cannot be interpreted as inference latency.

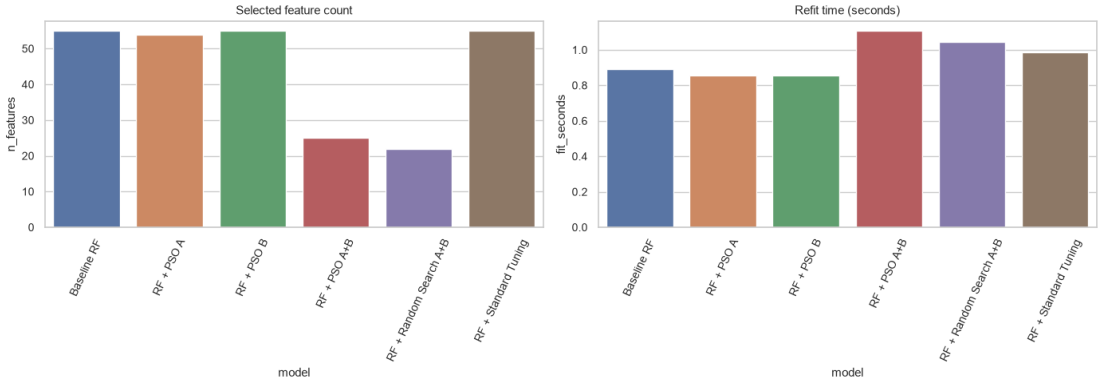

Figure 5. Selected feature counts and individual forest fitting times

The evidence in Figure 5 shows that substantial input compression is achievable even when predictive performance declines. Relative to the 55-feature baseline, the joint PSO branch removes 30 descriptors. Its selected forest fitting time of 1.1075 seconds exceeds the baseline time of 0.8915 seconds despite the smaller feature set. The larger 242-tree configuration and other parameter changes plausibly influence this timing, but causality was not isolated. The random joint search has fewer features and also lower macro-F1 than the proposed joint search. Feature count, training time, and multiclass recognition are therefore distinct study outcomes.

3.3. Optimization Trajectory and Error Structure

The evidence in Figure 6 presents the best penalized training-cross-validation fitness observed during four PSO iterations. It compares the feature-only, parameter-only, and joint PSO branches. Each branch uses six particles per iteration. The plotted objective includes the 0.01 feature-ratio penalty and is not a held-out F1 score. The feature-only branch attains the highest plotted terminal fitness among the three searches. The short trajectory documents search behavior but cannot establish convergence to a stable or global optimum.

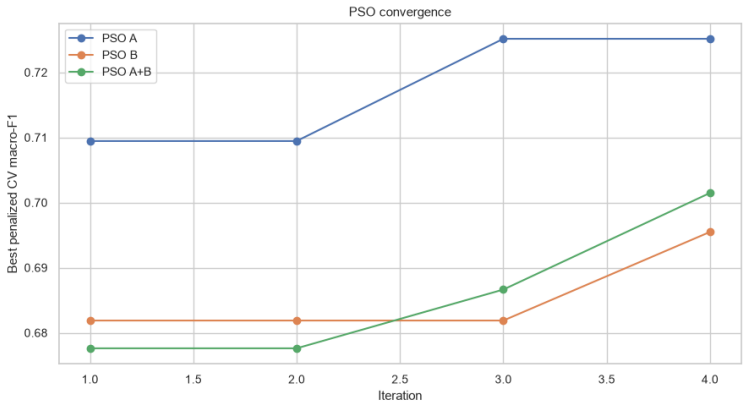

Figure 6. PSO convergence across four pilot iterations

The evidence in Figure 6 shows changes in the best observed penalized fitness over the limited search budget. PSO A rises to approximately 0.7252, while PSO B reaches about 0.6955. Joint PSO reaches about 0.7015 by the fourth iteration. These scores are training-cross-validation objectives and cannot be compared numerically with development macro-F1 without accounting for their different evaluation data and penalty. The final joint search improvement within its own trajectory is not sufficient to beat the baseline on the development set. Longer repeated searches would be required to assess whether the observed optimizer ranking is stable. The evidence in Figure 7 displays the archived confusion matrices for the baseline and three PSO variants. The matrices describe predictions on the same 270 development images. Their diagonal entries represent correctly classified cases, while off-diagonal counts indicate confusions among disease and health categories. The figure complements macro-F1 by showing where changes in recognition arise. Its four panels reflect the stored development predictions rather than independently measured final-test outcomes. The figure is retained directly from the executed notebook without altering any classification counts.

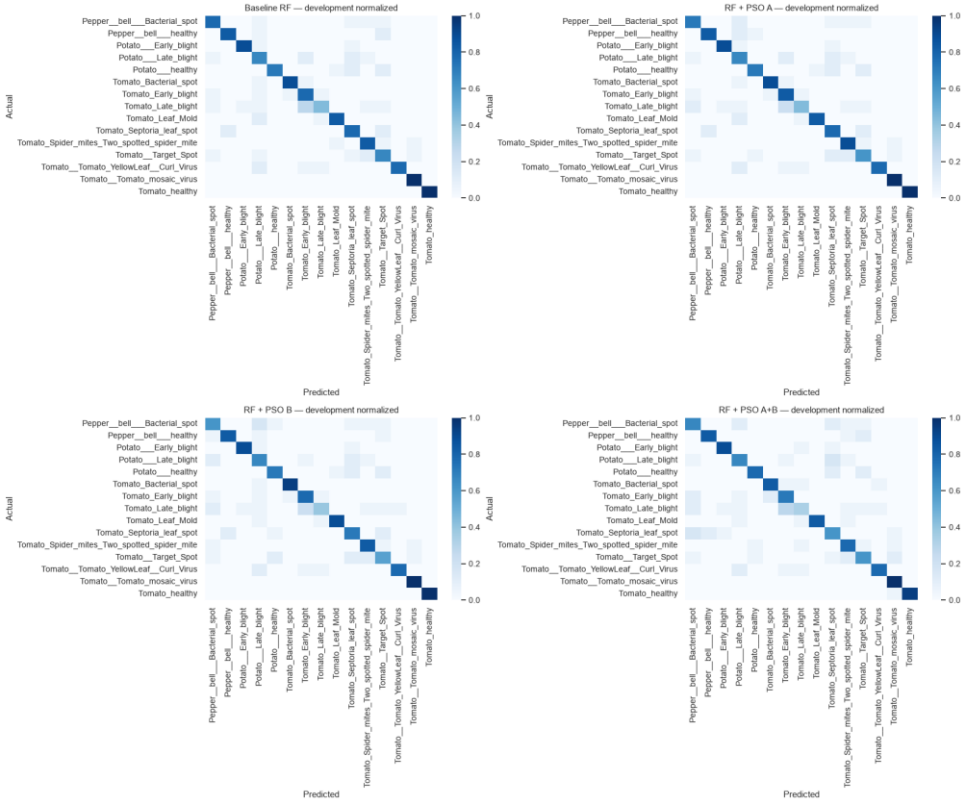

Figure 7. Selected development confusion matrices

The evidence in Figure 7 shows class-specific confusion patterns that are hidden by a single overall accuracy score. The baseline matrix is not uniformly perfect even though its aggregate macro-F1 is the strongest among the reported models. The hybrid alters the distribution of false classifications after the descriptor set is reduced. Visual

differences across confusion panels should not be interpreted as statistical significance. Small per-class development counts make several category rates sensitive to individual images. These limitations support examining class-level recall and F1 before endorsing a sparse model for disease screening.

The evidence in Figure 8 provides a per-class development F1 comparison for the baseline and the PSO configurations. Fifteen categories are included, spanning healthy and diseased pepper, potato, and tomato leaves. This breakdown prevents favorable performance on easily discriminated categories from masking other errors. It illustrates that optimization changes different classes by different amounts. The primary macro-F1 metric averages these class-specific outcomes equally. The plotted values come from the same fixed development predictions used in Table 6.

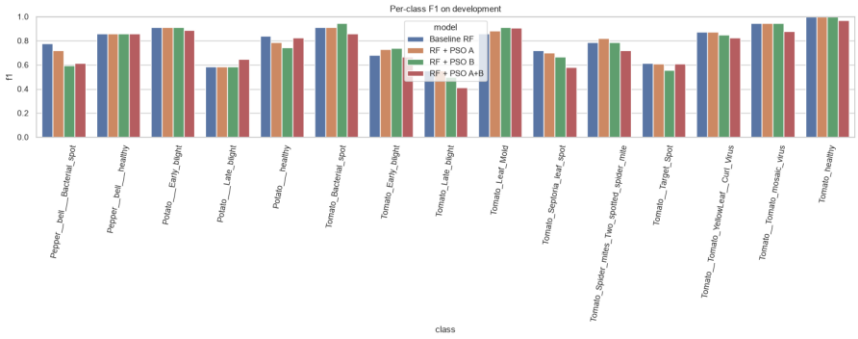

Figure 8. Class-level development F1 for baseline and PSO variants

The evidence in Figure 8 demonstrates that a reduced descriptor set does not produce a consistent gain across all diseases. Several categories are recognized more readily than others, irrespective of the optimizer. The joint PSO branch sometimes approaches the baseline but does not reverse the aggregate ranking. These differences are relevant because crop disease screening may assign distinct consequences to missed classes. The notebook does not supply independent error costs or deployment prevalences for a cost-sensitive recommendation. Per-class F1 therefore serves as a diagnostic rather than evidence of operational superiority.

3.4. Feature-Selection Interpretation and Limitations

The evidence in Figure 9 displays the feature-selection masks retained by the PSO A and joint PSO A+B models. Each row corresponds to one of the handcrafted color or texture descriptors. A selected coordinate represents a retained input feature in the fitted Random Forest. The feature-only PSO model selects 54 of 55 descriptors. The joint PSO model selects 25, indicating a substantially different search outcome. Mask differences can be interpreted as model configuration differences but do not establish causal feature importance.

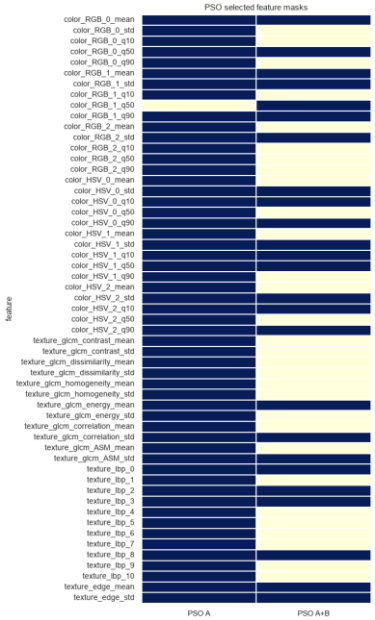

Figure 9. Descriptor-selection masks for PSO feature and joint search

The evidence in Figure 9 identifies where the main sparsity gain occurs. PSO A essentially preserves the complete representation, consistent with its macro-F1 remaining near the baseline. Joint PSO removes more than half of the descriptors, but its development macro-F1 falls by 0.0442 absolute points. The diagram does not show whether selected features would remain stable across repeated random seeds. Highly correlated color and texture measurements can produce alternative subsets with similar performance. The saved masks are therefore one observed configuration rather than a universally optimal feature inventory.

The evidence in Figure 10 compares ranked Random Forest importance estimates for the full-feature baseline and the joint PSO model. The displayed importance scores are computed from the fitted forest configurations rather than external field measurements. Both panels include color and texture descriptors. Their ranking differs because the optimized branch is trained on a reduced subset and modified forest hyperparameters. Feature-importance values are not directly comparable to causal effects of disease symptoms. The figure provides model interpretation only within the recorded development experiment.

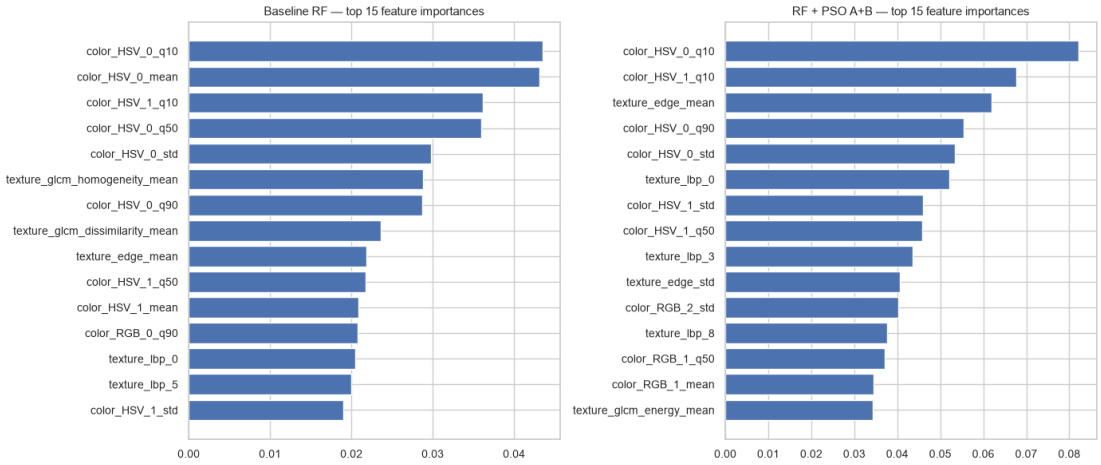

Figure 10. Random Forest feature importance before and after joint PSO

The evidence in Figure 10 highlights how the reduced subset changes which image summaries are emphasized by the forest. Several HSV and texture-derived terms remain among the highest-ranked features. This pattern is compatible with the use of color and local texture cues in leaf-condition recognition. The values cannot reveal whether the forest relies partly on background color or other acquisition artifacts. No counterfactual background perturbation or field-domain test was executed. The interpretive benefit of a smaller feature set must therefore be weighed against the recorded deterioration in macro-F1.

The development-stage evidence identifies a specific trade-off: the joint PSO configuration retains 25 rather than 55 descriptors but obtains macro-F1 0.7508 instead of 0.7950. Its 0.0206 absolute macro-F1 advantage over random joint search does not establish improvement against the stronger default forest. A naive paired bootstrap from the fixed development predictions gives a joint-minus-baseline difference near −0.0442 and a 95% interval of approximately [−0.0792, −0.0116]. This interval assumes independently resampled images and excludes plant-level dependence, optimizer variability, and retraining, so it is descriptive rather than confirmatory. The pilot also uses only four iterations of six particles, which may limit exploration of descriptor masks and forest configurations. The principal study limitation is the unexecuted 270-image final test, followed by the absence of plant-disjoint and field-image validation. Future experiments should lock optimizer selection before testing and report repeated-run uncertainty for both predictive quality and feature sparsity.

4. Conclusion

This pilot assessed PSO selection and Random Forest tuning for classification of 15 pepper, potato, and tomato leaf conditions. The baseline achieved development macro-F1 of 0.7950 using all 55 descriptors. The proposed joint PSO–Random Forest achieved 0.7508 while reducing the input set to 25 descriptors, corresponding to a 54.55% feature reduction. The joint search improved upon its random-joint comparator (0.7302 macro-F1) but underperformed the baseline and standard random tuning (0.7888). A training-side fitness penalty and a small particle budget may contribute to the observed trade-off, although their causal effects were not separately established. Per-class and confusion-matrix diagnostics demonstrate that the reduction does not translate into uniformly improved recognition. The reserved 270-image test set was not evaluated, and results from the limited

pilot cannot substantiate external superiority or field readiness. Future work should preregister candidate selection, run full-budget repeated experiments, use plant- or acquisition-group-disjoint field images, and report independent test metrics and uncertainty. The scientifically supported conclusion is feature compression with a development-stage predictive loss, not a verified classification improvement.

References

[1] J. K. Patil and V. S. Mandlik, "Plant leaf disease detection using integrated color and texture features," *Nature Environment and Pollution Technology*, vol. 23, pp. 2101–2113, 2024, doi: 10.46488/nept.2024.v23i04.016.

[2] S. M. Javidan, A. Banakar, K. Rahnama, K. A. Vakilian, and Y. Ampatzidis, "Feature engineering to identify plant diseases using image processing and artificial intelligence: A comprehensive review," *Smart Agricultural Technology*, vol. 8, p. 100480, 2024, doi: 10.1016/j.atech.2024.100480.

[3] S. R. Goyal, V. S. Kulkarni, R. Choudhary, and R. Jain, "A comparative analysis of efficacy of machine learning techniques for disease detection in some economically important crops," *Crop Protection*, vol. 190, p. 107093, 2025, doi: 10.1016/j.cropro.2024.107093.

[4] P. Sajitha, A. D. Andrushia, N. Anand, and M. Z. Naser, "A review on machine learning and deep learning image-based plant disease classification for industrial farming systems," *J. Ind. Inf. Integr.*, vol. 38, p. 100572, 2024, doi: 10.1016/j.jii.2024.100572.

[5] I. Pacal *et al.*, "A systematic review of deep learning techniques for plant diseases," *Artif. Intell. Rev.*, vol. 57, p. 304, 2024, doi: 10.1007/s10462-024-10944-7.

[6] A. Upadhyay *et al.*, "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture," *Artif. Intell. Rev.*, vol. 58, p. 92, 2025, doi: 10.1007/s10462-024-11100-x.

[7] M. Xu, J.-E. Park, J. Lee, J. Yang, and S. Yoon, "Plant disease recognition datasets in the age of deep learning: Challenges and opportunities," *Front. Plant Sci.*, vol. 15, p. 1452551, 2024, doi: 10.3389/fpls.2024.1452551.

[8] K. Zhan, Y. Peng, and M. Liao, "Domain generalization plant leaf disease recognition: Toward from laboratory to field," *Eng. Appl. Artif. Intell.*, vol. 156, p. 111168, 2025, doi: 10.1016/j.engappai.2025.111168.

[9] K. Antwi, K. E. Bennin, D. K. P. Asiedu, and B. Tekinerdogan, "On the application of image augmentation for plant disease detection: A systematic literature review," *Smart Agricultural Technology*, vol. 9, p. 100590, 2024, doi: 10.1016/j.atech.2024.100590.

[10] A. Rachmad *et al.*, "Classification of diseases on corn stalks using a random forest based on a combination of the feature extraction (local binary pattern and color histogram)," *Technium: Romanian Journal of Applied Sciences and Technology*, vol. 16, pp. 303–309, 2023, doi: 10.47577/technium.v16i.10002.

[11] X. Xie *et al.*, "A novel feature selection strategy based on salp swarm algorithm for plant disease detection," *Plant Phenomics*, vol. 5, p. 39, 2023, doi: 10.34133/plantphenomics.0039.

[12] A. Kurniawan, T. R. Soeprobowati, and B. Warsito, "Hybrid PSO-ACO optimization for rice leaf disease classification using random forest and support vector machines," *International Journal of Advanced Computer Science and Applications*, vol. 16, pp. 388–397, 2025, doi: 10.14569/IJACSA.2025.0160638.

[13] Y. Xue, X. Cai, and W. Jia, "Particle swarm optimization based on filter-based population initialization method for feature selection in classification," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, pp. 7355–7366, 2023, doi: 10.1007/s12652-022-04444-1.

[14] L. Qu, W. He, J. Li, H. Zhang, C. Yang, and B. Xie, "Explicit and size-adaptive PSO-based feature selection for classification," *Swarm Evol. Comput.*, vol. 77, p. 101249, 2023, doi: 10.1016/j.swevo.2023.101249.

[15] C. Zhang, Y. Xue, F. Neri, X. Cai, and A. Slowik, "Multi-objective self-adaptive particle swarm optimization for large-scale feature selection in classification," *Int. J. Neural Syst.*, vol. 34, p. 2450014, 2024, doi: 10.1142/S012906572450014X.

[16] E. Koçak and H. H. Örkcü, "Particle swarm optimization based feature selection using factorial design," *Hacettepe Journal of Mathematics and Statistics*, vol. 53, pp. 879–896, 2024, doi: 10.15672/hujms.1346686.

[17] H. Zhang, X. Qin, and X. Gao, "Probe mechanism based particle swarm optimization for feature selection," *Cluster Comput.*, vol. 27, pp. 8393–8411, 2024, doi: 10.1007/s10586-024-04408-4.

[18] M. Amudha and K. Brindha, "Effective feature selection based HOBS pruned-ELM model for tomato plant leaf disease classification," *PLoS One*, vol. 19, p. e0315031, 2024, doi: 10.1371/journal.pone.0315031.

[19] Q. Tian *et al.*, "Enhancing practicality of deep learning for crop disease identification under field conditions: Insights from model evaluation and crop-specific approaches," *Pest Manag. Sci.*, vol. 80, pp. 5864–5875, 2024, doi: 10.1002/ps.8317.

[20] M. H. Tunio *et al.*, "Advancing plant disease classification: A robust and generalized approach with transformer-fused convolution and Wasserstein domain adaptation," *Comput. Electron. Agric.*, vol. 227, p. 109574, 2024, doi: 10.1016/j.compag.2024.109574.

[21] T. Wei, Z. Chen, X. Yu, S. Chapman, P. Melloy, and Z. Huang, "A large-scale in-the-wild dataset for plant disease segmentation," *Sci. Data*, vol. 13, p. 205, 2026, doi: 10.1038/s41597-025-06513-4.

[22] Y. Natij *et al.*, "A resource-efficient framework for plant disease classification: Integrating reduced-order modeling with treatment-based label engineering," *Front. Plant Sci.*, vol. 17, p. 1766704, 2026, doi: 10.3389/fpls.2026.1766704.

[23] S. M. Shawon *et al.*, "Integrating federated learning and explainable AI for plant disease detection: A systematic review of recent advances in precision agriculture," *International Journal of Computational Intelligence Systems*, vol. 19, p. 138, 2026, doi: 10.1007/s44196-026-01219-w.

[24] D.-C. Rodríguez-Lira, D.-M. Córdova-Esparza, J. M. Álvarez-Alvarado, J. Terven, J.-A. Romero-González, and J. Rodríguez-Reséndiz, "Trends in machine and deep learning techniques for plant disease identification: A systematic review," *Agriculture*, vol. 14, p. 2188, 2024, doi: 10.3390/agriculture14122188.

[25] A. Singla *et al.*, "Exploration of machine learning approaches for automated crop disease detection," *Curr. Plant Biol.*, vol. 40, p. 100382, 2024, doi: 10.1016/j.cpb.2024.100382.

[26] Y. Zhao, Y. Li, N. Wu, and X. Xu, "Neural network based on convolution and self-attention fusion mechanism for plant leaves disease recognition," *Crop Protection*, vol. 180, p. 106637, 2024, doi: 10.1016/j.cropro.2024.106637.

[27] Ramanjot *et al.*, "Plant disease detection and classification: A systematic literature review," *Sensors*, vol. 23, p. 4769, 2023, doi: 10.3390/s23104769.

[28] P. K. Das, S. S. Rupa, S. Pumrin, U. C. Das, and M. K. Hossen, "Deep learning for plant disease detection and classification: A systematic analysis and review," *Curr. Appl. Sci. Technol.*, vol. 24, p. e0259016, 2024, doi: 10.55003/cast.2024.259016.

[29] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Mining and Knowledge Discovery*, vol. 13, p. e1484, 2023, doi: 10.1002/widm.1484.

[30] V. A. Metre and S. D. Sawarkar, "Convolutional neural network optimisation for discovering plant leaf diseases with particle swarm optimiser," *International Journal of Computational Science and Engineering*, vol. 27, pp. 443–457, 2024, doi: 10.1504/IJCSE.2024.139710.

[31] R. S. Jesie and M. S. G. Premi, "Improved tunicate swarm optimization based hybrid convolutional neural network for classification of leaf diseases and nutrient deficiencies in rice (Oryza)," *Agronomy*, vol. 14, p. 1851, 2024, doi: 10.3390/agronomy14081851.

[32] X. Jiang *et al.*, "PlantCaFo: An efficient few-shot plant disease recognition method based on foundation models," *Plant Phenomics*, vol. 7, p. 100024, 2025, doi: 10.1016/j.plaphe.2025.100024.

[33] R. P. Porfírio, R. N. Madeira, and P. A. Santos, "Exploring explainable AI techniques for plant disease classification in digital agriculture," *Procedia Comput. Sci.*, vol. 265, pp. 175–182, 2025, doi: 10.1016/j.procs.2025.07.170.

[34] F. O. Isinkaye, M. O. Olusanya, and P. K. Singh, "Deep learning and content-based filtering techniques for improving plant disease identification and treatment recommendations: A comprehensive review," *Heliyon*, vol. 10, p. e29583, 2024, doi: 10.1016/j.heliyon.2024.e29583.

[35] T. de Menezes e Silva Filho, H. Song, M. Perello-Nieto, R. Santos-Rodriguez, M. Kull, and P. Flach, "Classifier calibration: A survey on how to assess and improve predicted class probabilities," *Mach. Learn.*, vol. 112, pp. 3211–3260, 2023, doi: 10.1007/s10994-023-06336-7.