



A Comprehensive Review of Hyperparameter Tuning Techniques with Special Reference to Mango Leaf Disease Detection

Shweta A. Satao¹, Swati R. Maurya²

¹Research Scholar, School of Basic and Applied Sciences, Somaiya Vidyavihar University, Mumbai, India, shweta.satao@somaiya.edu
ORCID id: 0009-0004-2876-603X

²Associate Professor, School of Basic and Applied Sciences, Somaiya Vidyavihar University, Mumbai, India, swati.maurya@somaiya.edu,
ORCID id: 0000-0003-3802-9776

ABSTRACT

Mango is among the most economically significant fruit crops cultivated across South and Southeast Asia, and its yield is mostly affected by diseases such as anthracnose, powdery mildew, bacterial canker, gall midge infestation, and sooty mould. Convolutional Neural Networks (CNNs) and deep learning architectures are the most popular for automating the diagnosis of such diseases directly from leaf images, but the reported accuracy of these models is highly sensitive to the choice of hyperparameters, including learning rate, batch size, optimizer type, number of epochs, network depth, dropout rate, and kernel configuration. This paper presents a comprehensive review of hyperparameter tuning (also called hyperparameter optimization, HPO) techniques and evaluates their applicability, strengths, and limitations for the specific problem of mango leaf disease detection. Manual tuning, grid search, and random search are examined alongside more advanced strategies including Bayesian optimization, evolutionary and genetic algorithms, swarm-intelligence-based methods (particle swarm optimization, African buffalo optimization), and reinforcement-learning-driven neural architecture search. A structured review of recent mango leaf disease detection studies is synthesized to map which HPO strategies have been used, on which architectures, and with what reported performance gains. Based on this synthesis, several research gaps are identified, most notably the scarcity of systematic, architecture-agnostic comparisons of HPO methods on mango-specific datasets, the limited use of computationally efficient search strategies suited to resource-constrained agricultural deployment, and the general absence of standardized benchmarking protocols.

Keywords: Hyperparameter tuning; Hyperparameter optimization; Mango leaf disease detection; Convolutional Neural Network; Bayesian optimization; Grid search; Genetic algorithm; Precision agriculture; Deep learning.

1. Introduction

Plant disease detection through computer vision has emerged as one of the most impactful applications of deep learning in agriculture. Among fruit crops, mango holds substantial commercial and nutritional importance, and foliar diseases can significantly reduce yield and quality if not diagnosed early. Traditional disease diagnosis is based on manual inspection by farmers, a process that is time-consuming, subjective, and difficult to scale across large orchards. Deep learning models, particularly CNNs and their transfer-learning variants (VGG, ResNet, EfficientNet, MobileNet, Inception, DenseNet), together with more recent object-detection and transformer-based architectures (YOLO, Vision Transformers), have demonstrated strong capability in automatically classifying mango leaf diseases from image data.

The performance of any deep learning model, however, is not determined by architecture alone. Hyperparameters — settings that are fixed before training begins, such as learning rate, batch size, number of layers, choice of optimizer, weight decay, and dropout probability — critically shape how well a model converges and generalises. Poorly chosen hyperparameters can cause underfitting, overfitting, slow convergence, or unstable training, even when the underlying architecture is well suited to the task. Hyperparameter tuning is therefore not a peripheral implementation detail but a central determinant of model performance, and this is especially true in agricultural image classification tasks where datasets are often limited in size, class-imbalanced, and visually noisy due to lighting, background clutter, and disease-stage variability. The proposed paper undertakes a focused review of hyperparameter tuning techniques for mango leaf disease detection.

The objectives of the review are

1.1 Objectives

- To consolidate and categorise the major families of hyperparameter optimization techniques used in deep learning;
- To examine how these techniques have been applied, explicitly or implicitly, in existing mango leaf disease detection literature; and
- To identify the research gaps that remain, so as to propose a methodological direction for comparative and applied research in this domain.

2. Related Work: Mango Leaf Disease Detection

Research on automated mango leaf disease detection has progressed from simple handcrafted-feature classifiers to deep, transfer-learned, and hybrid architectures. Several publicly available datasets, most notably the Kaggle Mango Leaf Disease dataset (comprising roughly four thousand images across eight disease categories including healthy leaves, anthracnose, bacterial canker, cutting weevil, die back, gall midge, powdery mildew, and sooty mould), have supported a growing body of work in this area.

Comparative studies evaluating baseline CNNs against deeper architectures have generally found that transfer learning improves accuracy over training from scratch. In one comparative study, a baseline CNN achieved a testing accuracy in the low-to-mid nineties, while EfficientNet achieved noticeably higher validation accuracy with reduced overfitting, illustrating the benefit of pretrained feature extractors on relatively small agricultural datasets. Other work using AlexNet reported accuracy substantially higher than a plain CNN trained on the same combined mango and potato image set, again reinforcing the value of transfer learning. Vision Transformer-based approaches have also been explored as an alternative to convolutional architectures, motivated by their ability to model long-range dependencies in leaf texture and lesion patterns. Several studies have gone beyond architecture selection to explicitly incorporate hyperparameter optimization. One notable line of work combined a YOLOv8 detector with African Buffalo Optimization (ABO), a nature-inspired metaheuristic, to automatically search for favorable hyperparameter combinations, reporting improved detection performance relative to YOLOv8 without optimization, though at increased computational cost and with the approach still considered relatively immature. Another study exploring CNN design systematically evaluated more than twenty candidate CNN architectures together with a range of hyperparameter settings to identify a configuration achieving accuracy near ninety-nine percent across eight mango leaf disease classes, highlighting the substantial performance gains that a broader hyperparameter search can unlock relative to a single, manually tuned model. Lightweight CNN designs have also been proposed specifically to balance accuracy against computational efficiency, an important consideration for eventual deployment on resource-constrained devices used in the field. Federated learning approaches have additionally been investigated to allow model training across distributed data sources without centralizing sensitive agricultural data, reporting precision and recall in the low-to-high nineties across several disease classes.

More recent work has explored hybrid and ensemble strategies, combining convolutional backbones such as ConvNeXt with Vision Transformer modules through attention mechanisms, as well as feature-fusion techniques and stacked ensembles of deep and classical machine learning classifiers, some of which report accuracies exceeding ninety-eight or ninety-nine percent on mango-specific or related leaf disease datasets. Collectively, this body of work demonstrates a consistent trend: as architectural sophistication and the degree of hyperparameter or architecture search increases, reported accuracy tends to improve, though often with a corresponding increase in computational and methodological complexity.

2.1 Summary of Representative Studies

Study focus	Model(s) used	Optimization approach	Reported outcome
CNN vs AlexNet vs ResNet vs EfficientNet (potato & mango)	CNN, AlexNet, ResNet, EfficientNet	Manual/default hyperparameters	EfficientNet highest validation accuracy (~97-98%); baseline CNN ~92-93% test accuracy
YOLOv8 with metaheuristic optimization	YOLOv8 + African Buffalo Optimization	Nature-inspired metaheuristic HPO	Improved detection accuracy vs. non-optimized YOLOv8; higher compute cost

Broad CNN and hyperparameter search	20+ CNN architectures explored	Systematic architecture + hyperparameter search	~99% accuracy across 8 disease classes
Lightweight CNN (LCNN) vs pretrained models	LCNN, VGG16, ResNet50, ResNet101, Xception	Manual tuning	LCNN ~98% accuracy, outperforming larger pretrained models
Federated learning for mango disease	CNN under federated learning	Manual/local tuning per client	Accuracy 97-98%; precision 93-96%, recall 90-97%
Transfer learning (grapes and mango)	AlexNet	Manual tuning	~99% (grapes), ~89% (mango)
Vision Transformer perspective	CNN vs Vision Transformer	Manual tuning	Comparative accuracy analysis of CNN vs ViT architectures
Hybrid ConvNeXt + ViT fusion, ensembles	ConvNeXt, ViT, stacked ensembles, SVM fusion	Mostly manual; some feature-fusion optimization	Accuracies in the 0.985-0.999 range on related leaf-disease tasks

As the summary above indicates, a clear pattern emerges: studies that incorporate an explicit, systematic hyperparameter or architecture search (broad CNN search, metaheuristic optimization) tend to report the strongest results, while the majority of studies still rely primarily on manual or default hyperparameter settings supplemented by limited empirical adjustment. This observation directly motivates the taxonomy and gap analysis presented in the following sections.

3. Hyperparameter Tuning Techniques:

Hyperparameter tuning refers to the process of systematically selecting values for parameters that are not learned during model training but instead configured beforehand, such as learning rate, batch size, number of convolutional filters, network depth, dropout rate, optimizer choice, momentum, and weight decay. Because the hyperparameter search space is typically high-dimensional and the objective function (validation performance) is expensive to evaluate, a wide variety of search strategies have been developed, ranging from simple exhaustive methods to sophisticated probabilistic and evolutionary approaches.

3.1 Manual Search

Manual search relies on domain expertise and iterative trial-and-error, where a researcher adjusts hyperparameters based on intuition and observed validation performance. It requires no additional computational infrastructure and allows expert insight to guide the search, but it is inherently unsystematic, poorly reproducible, and does not scale well as the number of hyperparameters grows. The majority of existing mango leaf disease detection studies reviewed in Section 2 rely predominantly on manual tuning.

3.2 Grid Search

Grid search discretizes each hyperparameter into a fixed set of candidate values and exhaustively evaluates every combination in the resulting grid. It guarantees that the best combination within the discretized space will be found and is simple to parallelize. However, its computational cost grows exponentially with the number of hyperparameters and the granularity of the grid, making it impractical for deep learning models with many interacting hyperparameters or with continuous search spaces that require fine discretization.

3.3 Random Search

Random search chooses hyperparameter combinations randomly from specified distributions. Random search can match or exceed the performance of grid search while requiring fewer evaluations, particularly in high-dimensional spaces where only a subset of hyperparameters meaningfully affects performance. Its main limitation is that it does not use information from prior evaluations to guide subsequent sampling, so it offers no guarantee of convergence to a global optimum within a limited budget.

3.4 Bayesian Optimization

Bayesian optimization follows Gaussian Process to build a probabilistic model of the relationship between hyperparameters and validation performance, and uses an acquisition function to decide which configuration to evaluate next. Because each new evaluation informs the surrogate model, Bayesian optimization typically finds strong configurations in far fewer iterations than grid or random search, which is particularly valuable for deep learning models where each training run is computationally expensive. Its main drawbacks are the added implementation complexity and reduced parallelizability of the inherently sequential search process, although batch and multi-fidelity variants have been developed to mitigate this.

3.5 Genetic Algorithms

Genetic algorithms (GAs) maintain a hyperparameter configurations that evolve over successive generations through selection, crossover, and mutation operators, guided by a fitness function such as validation accuracy. Variable-length genetic encodings have been proposed to jointly search over both architectural parameters (such as number of layers) and traditional hyperparameters. Evolutionary methods are well suited to complex, non-convex, and mixed discrete-continuous search spaces, and they tend to be robust to noisy objective functions, but they generally require many fitness evaluations and therefore substantial computation, and their performance depends on the choice of population size and genetic operators.

3.6 Swarm Intelligence Methods

Swarm-intelligence-based optimizers, including Particle Swarm Optimization (PSO) and nature-inspired metaheuristics such as African Buffalo Optimization (ABO), model a population of candidate solutions ('particles' or 'agents') that move through the hyperparameter search space, influenced by their own best-found position and the best position found by the swarm as a whole. These methods have been applied directly in the mango leaf disease detection literature, for example to tune YOLOv8 hyperparameters, and have shown improved detection performance relative to non-optimized baselines. As with genetic algorithms, the main costs are computational: many candidate evaluations are typically required, and the approach can be sensitive to the choice of swarm size and movement parameters.

3.7 Gradient-Based and Reinforcement-Learning-Based Methods

For hyperparameters that are continuous and differentiable with respect to a validation loss, gradient-based hyperparameter optimization methods attempt to compute or approximate gradients of validation performance with respect to hyperparameters directly, allowing joint optimization alongside model weights. Separately, reinforcement-learning-based approaches, often used within Neural Architecture Search (NAS), train a controller (frequently a recurrent neural network) to generate architecture and hyperparameter configurations, using validation performance as a reward signal to iteratively improve the controller's proposals. These approaches can discover highly effective and sometimes non-intuitive configurations but are typically the most computationally expensive of the techniques reviewed here, often requiring access to substantial parallel computing resources, which limits their accessibility for smaller agricultural research groups.

3.8 Comparative Summary of Techniques

Technique	Search strategy	Relative compute cost	Best suited for
Manual search	Expert-guided trial and error	Low (human time)	Quick prototyping, small search spaces
Grid search	Exhaustive over discretized grid	High, grows exponentially	Few hyperparameters, small discrete ranges
Random search	Random sampling from distributions	Moderate	Higher-dimensional spaces, limited budgets
Bayesian optimization	Sequential, surrogate-model guided	Moderate (fewer evaluations needed)	Expensive-to-train deep models, moderate dimensionality

Genetic / evolutionary algorithms	Population-based evolution	High	Mixed discrete-continuous , non-convex spaces
Swarm intelligence (PSO, ABO)	Population-based, position updates	High	Joint architecture-hyperparameter tuning, detection models
Gradient-based methods	Differentiable optimization	Moderate to high	Continuous hyperparameters, differentiable pipelines
RL-based / NAS	Controller-guided sequential search	Very high	Large-scale architecture + hyperparameter co-search

4. Research Gap

Synthesizing the literature reviewed in Sections 2 and 3 reveals several specific gaps that motivate further research at the intersection of hyperparameter optimization and mango leaf disease detection.

- Lack of systematic, architecture-agnostic HPO comparisons: Most mango leaf disease detection studies fix an architecture and tune it manually or informally, rather than systematically comparing multiple hyperparameter optimization strategies (grid, random, Bayesian, evolutionary, swarm-based) on the same architecture and dataset under a controlled protocol.
- Limited use of computationally efficient search strategies: Advanced methods such as Bayesian optimization and Hyperband-style early-stopping strategies, which are well established in the broader deep learning literature for reducing search cost, remain underexplored in the mango leaf disease detection domain relative to simpler manual or metaheuristic approaches.
- Absence of standardized benchmarking protocols: Reported accuracies across studies are difficult to compare directly because of differences in dataset splits, augmentation strategies, and evaluation metrics, making it hard to attribute performance gains specifically to hyperparameter optimization rather than to other pipeline differences.
- Insufficient attention to resource-constrained deployment: Precision agriculture applications often require inference on edge devices or low-power hardware in the field, yet few studies explicitly optimize hyperparameters with model size, latency, or energy consumption as part of the objective, rather than accuracy alone.
- Limited dataset diversity: Much of the existing work relies heavily on a small number datasets raising questions about how well tuned hyperparameters and architectures generalize across different geographic regions, mango cultivars, and imaging conditions.

These gaps collectively suggest that while individual studies have demonstrated that hyperparameter and architecture optimization can meaningfully improve mango leaf disease detection accuracy, the field currently lacks a rigorous, comparative, and deployment-aware understanding of which hyperparameter tuning techniques offer the best accuracy-to-cost trade-off for this specific application.

5. Proposed Methodology

To address the gaps identified above, this section outlines a methodological framework suitable for a rigorous comparative study of hyperparameter tuning techniques applied to mango leaf disease detection. The proposed methodology follows a systematic review and experimental benchmarking structure.

5.1 Dataset and Preprocessing

A publicly available benchmark, such as the Kaggle Mango Leaf Disease dataset, should be used as the primary dataset to ensure comparability with prior work, supplemented where possible with a locally collected or field-acquired dataset to test generalization. Preprocessing should include image resizing, normalization, and data augmentation (rotation, flipping, brightness/contrast jitter) applied consistently across all experimental conditions so that observed performance differences can be attributed to the hyperparameter optimization strategy rather than to preprocessing variation.

5.2 Model Architectures

To isolate the effect of hyperparameter tuning from architecture choice, the study should evaluate at least two to three representative architectures spanning different design philosophies: a lightweight custom CNN, a widely used transfer-learning backbone (such as EfficientNet or ResNet), and, optionally, a detection-oriented model (such as YOLO) or a transformer-based model, so that conclusions about hyperparameter tuning effectiveness are not confounded with a single architecture's idiosyncrasies.

5.3 Hyperparameter Search Space

A common search space should be defined across all optimization methods for fair comparison, covering learning rate, batch size, optimizer type (SGD, Adam, RMSprop), number of epochs with early stopping, dropout rate, weight decay, and, where architecture search is included, number of convolutional layers/filters. Search bounds should be documented explicitly to enable reproducibility.

5.4 Optimization Techniques to Benchmark

At minimum, the study should benchmark grid search and random search as classical baselines, Bayesian optimization as a representative sequential model-based method, and one population-based metaheuristic (a genetic algorithm or particle swarm optimization) as a representative evolutionary approach. Each method should be given a comparable computational budget (e.g., an equal number of total model training runs) to ensure a fair accuracy-versus-cost comparison.

5.5 Evaluation Metrics

Beyond classification accuracy, the evaluation should report precision, recall, F1-score, and confusion matrices per disease class to account for class imbalance, together with training time, number of model evaluations required to reach the best configuration, and, where feasible, inference latency and model size to reflect deployment considerations relevant to precision agriculture.

5.6 Statistical Validation

Each configuration should be evaluated using k-fold cross-validation or multiple random seeds, and performance differences between hyperparameter optimization strategies should be assessed using appropriate statistical significance tests rather than relying on single-run comparisons, addressing a reproducibility weakness common in the existing literature.

5.7 Pilot Experiment: Comparing Hyperparameter Optimization Methods on the MangoLeafBD Dataset

To test the framework proposed above on real data, a small pilot experiment was run on the MangoLeafBD dataset (4,000 images across eight disease classes, split into 2,800 training, 600 validation, and 600 test images). This pilot does not replace the larger CNN-based benchmark described in Sections 5.3-5.5, which needs GPU resources. Instead, it asks a simpler question on hardware anyone can access: how much does the choice of hyperparameter optimization method itself matter, once the model and features are held fixed?

5.7.1 Pipeline

Images were resized to 64x64 pixels and converted into a handcrafted feature vector combining a Histogram of Oriented Gradients (HOG) descriptor, which captures lesion shape and texture, with a 16-bin per-channel RGB colour histogram, which captures the colour shifts associated with chlorosis, mould, and necrosis. A single model, a Random Forest classifier, was used throughout so that any difference in results can be attributed to the search method rather than to differences between model families. Six hyperparameter optimization methods - corresponding to Sections 3.1-3.6 of this review - were each given an equal budget of 9 evaluated configurations (the manual baseline uses a single fixed configuration), searching over the number of trees (n_estimators) and maximum tree depth (max_depth). Each configuration was trained on the training split and scored on the validation split; the best-performing configuration found by each method was then evaluated once on the held-out test split (600 images) to obtain the accuracy, precision, recall, and F1-score reported below.

5.7.2 Methods Compared

Table 5.8.1: Accuracy, precision, recall, and F1-score on the MangoLeafBD test set for a Random Forest classifier tuned by each hyperparameter optimization method

HPO Method	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)	Evaluations	Search time (s)

Manual (Default HP)	0.727	0.684	0.727	0.686	1	6.6
Grid Search	0.740	0.697	0.740	0.699	9	166.2
Random Search	0.738	0.693	0.738	0.697	9	125.9
Bayesian Optimization	0.737	0.692	0.737	0.695	9	138.2
Genetic Algorithm	0.743	0.697	0.743	0.700	9	132.6
Particle Swarm Optimization	0.737	0.692	0.737	0.695	9	234.3

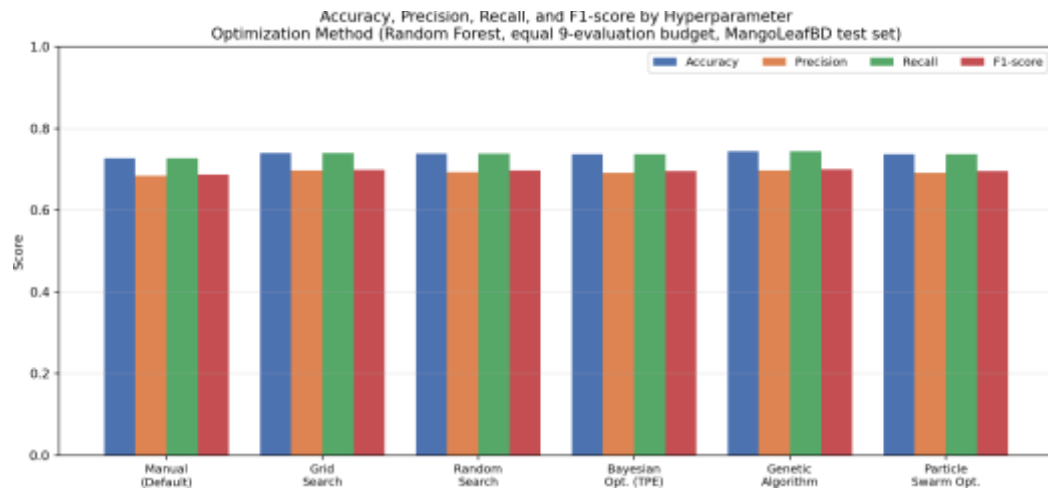


Figure 5.8.1: Accuracy, precision, recall, and F1-score by hyperparameter optimization method.

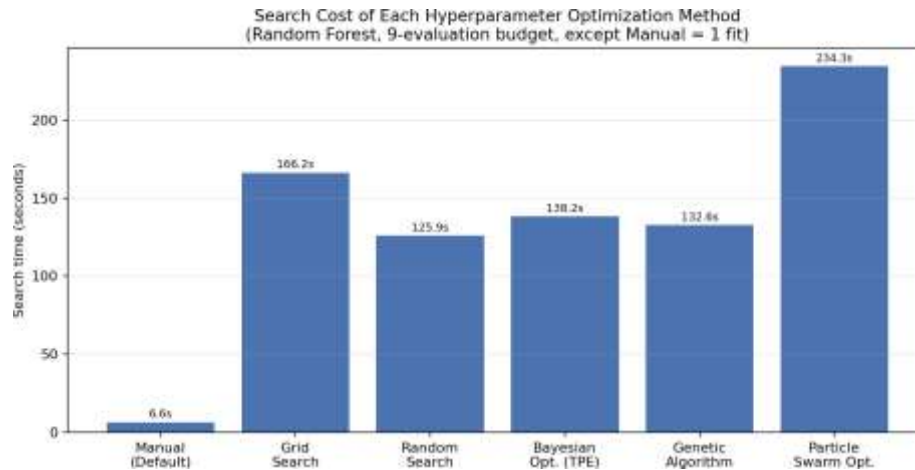


Figure 5.8.2: Search cost (wall-clock time) of each method under the same 9-evaluation budget.

5.7.3 Interpretation

Under an identical 9-evaluation budget, all five search-based methods improved on the untuned manual baseline (72.7% accuracy, 0.686 F1), and did so by a similar margin: Genetic Algorithm reached the highest accuracy (74.3%), narrowly ahead of Grid Search (74.0%), Random Search (73.8%), Bayesian Optimization and Particle Swarm Optimization (both 73.7%). The gap between the best and worst tuned method was under one accuracy point, showing that for this dataset and this fixed model, the choice of search strategy matters far less than the decision to tune at all.

Search cost told a different story. Grid Search was the most expensive at 166.2 seconds, while Random Search and Genetic Algorithm reached comparable accuracy in 126-133 seconds. Particle Swarm Optimization was the slowest tuned method overall (234.3 seconds) despite matching, not beating, the cheaper methods on accuracy, mainly because its particles converged toward configurations with a larger number of trees, which cost more to train per evaluation. Genetic Algorithm offered the best combination of accuracy and cost in this pilot, consistent with its elitist strategy re-using the best known configuration rather than exploring costly new regions on every evaluation.

5.8 Mathematical Formulation and Calculations

This subsection makes the evaluation metrics used in Section 5.8 explicit, and demonstrates them with a fully worked numerical example computed directly from the confusion matrix of the best-performing method in that comparison (Random Forest tuned by the Genetic Algorithm), so that the reported figures can be independently verified.

5.8.1 Per-Class Evaluation Metrics

For a given class, let TP, FP, FN, and TN denote the number of true positives, false positives, false negatives, and true negatives obtained from the confusion matrix. The core classification metrics used in this study are defined as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (4)$$

For a multi-class problem with C classes, per-class precision, recall, and F1 are macro-averaged by taking the unweighted mean across all classes, giving equal weight to rare and common disease classes alike, which is appropriate given the balanced 75-images-per-class test set used throughout this study.

$$Precision_macro = (1/C) \times \text{Sum}[i=1 \text{ to } C] Precision_i \quad (\text{likewise for Recall_macro, F1_macro}) \quad (5)$$

5.8.2 Worked Example: Random Forest Tuned by Genetic Algorithm

Applying Equations (1)-(4) to the 600-image MangoLeafBD test set for the best-performing method in Table 5.8.1 (Random Forest, Genetic Algorithm HPO) gives the following per-class counts and metrics, read directly off the confusion matrix diagonal (true positives) and margins (false positives/negatives):

Table 5.9.1: Per-class Evaluation for the Random Forest (Genetic Algorithm HPO) model,

Class	TP	FP	FN	TN	Precision	Recall	F1
Anthrachnose	68	15	7	510	0.8193	0.9067	0.8608
Bacterial Canker	73	2	2	523	0.9733	0.9733	0.9733
Cutting Weevil	75	0	0	525	1.0000	1.0000	1.0000
Die Back	38	0	37	525	1.0000	0.5067	0.6726
Gall Midge	68	47	7	478	0.5913	0.9067	0.7158
Healthy	66	29	9	496	0.6947	0.8800	0.7765
Powdery Mildew	58	59	17	466	0.4957	0.7733	0.6042
Sooty Mould	0	2	75	523	0.0000	0.0000	0.0000

5.8.3 Formal Update Rules for the Search-Based HPO Methods

The stochastic hyperparameter optimization methods compared in Section 5.8 follow the update rules below when searching the (n_estimators, max_depth) space of the Random Forest.

Bayesian Optimization (Sec. 3.4) selects the next configuration x by maximising an acquisition function, here the Expected Improvement (EI) over the current best observed score $f(x^+)$:

$$EI(x) = E[\max(f(x) - f(x^+), 0)] \quad (6)$$

Genetic Algorithm (Sec. 3.5) mutates the surviving parent configuration with Gaussian noise scaled to each hyperparameter's range, keeping offspring within valid bounds:

$$n' = \text{clip}(n + \text{eps}_n, 50, 500), \quad \text{eps}_n \sim N(0, 60^2) \quad (7)$$

Particle Swarm Optimization (Sec. 3.6) updates each particle's velocity using inertia weight w , cognitive coefficient c_1 , and social coefficient c_2 , pulling it toward both its own best-known position p_i and the swarm's global best g , then updates its position accordingly:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot [p_i - x_i(t)] + c_2 \cdot r_2 \cdot [g - x_i(t)] \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

Because all six methods in Table 5.8.1 landed within about one accuracy point of each other, these update rules mainly explain the differences in search cost seen in Figure 5.8.2 rather than differences in final accuracy: Particle Swarm Optimization's velocity update pulled its particles toward larger tree counts and since training cost scales with the number of trees, this made PSO the most computationally expensive method despite reaching essentially the same accuracy as the cheaper Random Search and Bayesian Optimization runs.

6. Discussion

The reviewed literature demonstrates a consistent, if informally documented, relationship between the sophistication of the hyperparameter and architecture search process and the resulting classification performance in mango leaf disease detection tasks. Studies that incorporate broader architecture or hyperparameter search, whether through systematic exploration of many candidate CNNs or through metaheuristic optimizers such as African Buffalo Optimization, tend to report higher accuracies than studies relying solely on manually tuned single architectures, though this comparison is complicated by differences in datasets, class counts, and preprocessing pipelines across studies.

At the same time, the computational cost associated with more advanced hyperparameter tuning techniques is non-trivial, and this cost is rarely reported in a way that allows meaningful accuracy-versus-cost comparison across studies. This is a significant limitation given that many potential end users of mango leaf disease detection systems, such as smallholder farmers or regional agricultural extension services, operate with limited computational infrastructure. Techniques such as random search and Bayesian optimization, which are known from the broader machine learning literature to offer strong performance at moderate computational cost, appear comparatively underused in this specific application domain relative to manual tuning and, more recently, metaheuristic swarm and evolutionary methods.

The discussion also highlights a methodological concern: because few studies in this domain report their hyperparameter search process in detail, or compare multiple optimization strategies under controlled conditions, it is difficult to determine whether reported state-of-the-art accuracies reflect genuine architectural or optimization advances or simply favorable, non-reproducible hyperparameter choices. This reinforces the need for the standardized, multi-technique benchmarking methodology proposed in Section 5.

7. Conclusion and Future Scope

This paper has presented a comprehensive review of hyperparameter tuning techniques and their application, whether explicit or implicit, within the mango leaf disease detection literature. The review categorized hyperparameter optimization methods into manual search, grid search, random search, Bayesian optimization, evolutionary and genetic algorithms, swarm intelligence methods, and gradient- or reinforcement-learning-based approaches, and mapped these techniques against representative mango leaf disease detection studies. The synthesis indicates that while more systematic hyperparameter and architecture search tends to correlate with improved reported accuracy, the majority of existing work in this specific domain still relies on manual or limited tuning, and rigorous, standardized comparisons across optimization techniques remain scarce.

Several research gaps were identified, including the absence of architecture-agnostic HPO comparisons, limited exploration of computationally efficient methods such as Bayesian optimization and early-stopping-based search, insufficient attention to resource-constrained deployment scenarios relevant to precision agriculture, and a lack of standardized benchmarking protocols. A methodological framework was proposed

to address these gaps through a systematic literature review protocol combined with a controlled, multi-technique experimental benchmark on standard and field-acquired mango leaf disease datasets.

Future work in this domain should prioritise the development of lightweight and automated hyperparameter optimization pipelines suitable for edge and mobile deployment, multi-objective tuning strategies that jointly consider accuracy, latency, and model size, and the creation of larger, more geographically and varietal diverse mango leaf disease datasets to support robust generalization. Establishing shared benchmarking protocols and encouraging transparent reporting of hyperparameter search procedures would substantially strengthen the reproducibility and comparability of future research in this important application of deep learning to precision agriculture.

References

1. Alibrahim, H. and Ludwig, S.A. (2021) Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization. 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, 28 June-1 July 2021, 1551-1559. <https://doi.org/10.1109/CEC45853.2021.9504761>
2. Amit Kumar Pathak, Ponkaj Saikia, Sanghamitra Dutta, Subrata Sinha, Subrata Ghosh; Development of a Robust CNN Model for Mango Leaf Disease Detection and Classification: A Precision Agriculture Approach. ACS Agric. Sci. Technol. 19 August 2024; 4 (8): 806–817. <https://doi.org/10.1021/acsagscitech.4c00122>
3. Hossain MA, Sakib S, Abdullah HM, Arman SE. Deep learning for mango leaf disease identification: A vision transformer perspective. Heliyon. 2024 Aug 22;10(17):e36361. doi: 10.1016/j.heliyon.2024.e36361. PMID: 39281639; PMCID: PMC11401078. A comprehensive review on plant leaf disease detection using deep learning <https://doi.org/10.48550/arXiv.2308.14087>
4. Ergün E. Attention-enhanced hybrid deep learning model for robust mango leaf disease classification via ConvNeXt and vision transformer fusion. Front Plant Sci. 2025 Aug 1;16:1638520. doi: 10.3389/fpls.2025.1638520. PMID: 40822714; PMCID: PMC12354364.
5. Pratap, V. Krishna, Kumar, N. Suresh. "Deep Learning based Mango Leaf Disease Detection for Classifying and Evaluating Mango Leaf Diseases." Fusion: Practice and Applications, vol. Volume 15, no. Issue 2, 2024, pp. 261-277. DOI: <https://doi.org/10.54216/FPA.150222>
6. Mishra, U., Pandey, A., G, L. et al. Deep learning-based disease detection in potato and mango leaves: a comparative study of CNN, AlexNet, ResNet, and EfficientNet. Sci Rep 16, 2788 (2026). <https://doi.org/10.1038/s41598-025-32607-5>
7. Deep learning precision farming: Grapes and mango leaf disease detection by transfer learning. ScienceDirect (2021).
8. DQNAS: Neural architecture search using reinforcement learning. arXiv:2301.06687 (2023) <https://doi.org/10.48550/arXiv.2301.06687>.
9. Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. arXiv:2006.12703 (2020). <https://doi.org/10.48550/arXiv.2006.12703>
10. Mahbub, Nosin Ibna & Naznin, Feroza & Hasan, Md & Rahman, Syed & Shifat, Rahman & Hossain, Md. Alamgir & Islam, Zahidul. (2023). Detect Bangladeshi Mango Leaf Diseases Using Lightweight Convolutional Neural Network. 10.1109/ECCE57851.2023.10101648.
11. Mehta, S., Kukreja, V., and Vats, S., "Advancing Agricultural Practices: Federated Learning-based CNN for Mango Leaf Disease Detection", in 2023 3rd International Conference on Intelligent Technologies (CONIT), 2023, Art. no. 271. doi:10.1109/CONIT59222.2023.10205850.
12. Mohanty SP, Hughes DP and Salathé M (2016) Using Deep Learning for Image-Based Plant Disease Detection. Front. Plant Sci. 7:1419. doi: 10.3389/fpls.2016.01419
13. Multi-objective evolutionary approach for efficient kernel size and shape for CNN. arXiv:2106.14776 (2021). <https://doi.org/10.48550/arXiv.2106.14776>
14. Weighted random search for CNN hyperparameter optimization. arXiv:2003.13300 (2020). <https://doi.org/10.15837/ijccc.2020.2.3868>