

Article

Efficient Olive Leaf Disease Detection via Hybrid Artificial Rabbit Optimization and Genetic Algorithm-Based Deep Feature Selection

Cumali Turkmenoglu ¹ , Hakan Gunduz ^{2,*}  and Emrullah Gazioglu ³ 

¹ Software Engineering Department, Faculty of Engineering, Fatih Sultan Mehmet Vakif University, Istanbul 34445, Türkiye; cturkmenoglu@fsm.edu.tr

² Software Engineering Department, Faculty of Engineering, Kocaeli University, Kocaeli 41001, Türkiye

³ Computer Engineering Department, Faculty of Engineering, Sirtak University, Sirtak 73000, Türkiye; gazioglu@sirtak.edu.tr

* Correspondence: hakan.gunduz@kocaeli.edu.tr

Abstract

Artificial intelligence (AI)-supported agricultural disease detection has become increasingly important for addressing global food security challenges. In this study, a hybrid meta-heuristic optimization-based feature selection approach is proposed for the detection of peacock eye disease (*Venturia oleaginea*) on olive leaves. The proposed method combines Artificial Rabbit Optimization (ARO) and Genetic Algorithm (GA) strategies to balance global exploration and local exploitation during feature selection. Comprehensive experiments conducted on a dataset of 954 olive leaf images demonstrate that the proposed approach achieves an F1-score of 99.7% while reducing the feature dimensionality by 95%, selecting only 100 features from ResNet101. Statistical analysis confirms that the method significantly outperforms standalone GA and ARO approaches ($p < 0.05$, paired t -tests), demonstrating superior long-term convergence behavior and a 47–56% reduction in performance variance across repeated runs. Compared to existing approaches in the literature, the proposed method attains competitive or superior accuracy with substantially fewer features, indicating a marked reduction in computational complexity. These results suggest that the proposed hybrid feature selection framework has strong potential for deployment in resource-constrained agricultural monitoring scenarios, where efficient inference and reduced model complexity are critical.

Keywords: deep learning feature optimization; hybrid meta-heuristic algorithms; agricultural disease detection; computational efficiency; precision agriculture



Academic Editor: Marina Lukić

Received: 28 January 2026

Revised: 22 February 2026

Accepted: 27 February 2026

Published: 9 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The integration of AI into agriculture has begun to fundamentally change the approach to food security. Driven by the need to optimize productivity in the context of climate variability and declining arable land, “smart farming” has evolved from a theoretical concept into a practical journey [1,2]. These data-driven technologies have begun to offer suitable ways to secure nutrition and economic stability for the growing global population [3–5].

Despite these advances, plant diseases not only continue to be a persistent threat to agricultural productivity, but also claim an estimated 10–16% of annual global crop production. In olive cultivation, *Venturia oleaginea* (peacock eye disease) creates particularly

serious problems. This disease attacks leaf tissue, causes premature defoliation, weakens the tree and significantly reduces fruit quality. In regions where the disease is endemic, growers frequently face yield losses of 20–30% [6]. Since traditional visual inspection methods are labor-intensive, costly, and often fail to detect infection until significant damage has occurred, it is evident that there is a clear need for automated, early detection systems.

This issue also has critical economic importance for Turkey. Over the last three decades, Turkey has positioned itself as a major global producer and expanded production to 380,000 tons in the 2022/23 season [7,8]. Turkey has now begun to contribute approximately 14% of the global olive supply [7], and the olive sector has become one of the cornerstones of the national economy. Consequently, even modest improvements in disease management provide significant economic benefits.

Deep learning, particularly Convolutional Neural Networks (CNNs), has demonstrated exceptional performance in identifying plant diseases from imagery [9–11]. However, a significant gap remains between laboratory performance and practical field deployment. Standard architectures like ResNet101 and MobileNet are computationally intensive, extracting 2048 and 1024 features per image, respectively. This high dimensionality creates a bottleneck for edge devices which are constrained by limited processing power and battery life. Moreover, excessive feature redundancy increases the risk of overfitting, where models fail to generalize to new data [12,13]. Therefore, the primary challenge is no longer just detecting disease, but achieving high accuracy with a computational footprint small enough for real-world agricultural hardware.

Research Challenges and Gaps

Several interconnected problems prevent the deployment of deep learning disease detection in resource-constrained environments. First, existing approaches have not systematically compared different CNN architectures under optimized feature selection conditions; oriented design (ResNet101 or MobileNet) performs better when feature budgets are severely limited.

Second, most feature selection studies apply either simple statistical filters or individual meta-heuristic algorithms. Meta-heuristic optimization offers a compromise, yet existing works predominantly use individual algorithms that struggle to balance exploration and exploitation. Genetic Algorithms refine solutions efficiently but converge prematurely in complex search spaces [14]. Nature-inspired methods like ARO, Whale Optimization Algorithm (WOA), and similar approaches explore broadly but often converge slowly and inadequately exploit promising regions [15].

Third, while researchers report achieving 20–50% feature reduction, no one has systematically investigated whether 90% or greater reduction is feasible while maintaining classification performance. This question has immediate practical implications: can we deploy effective disease detection models onto edge devices?

Finally, most studies report accuracy metrics without addressing deployment realities that determine whether a system can actually function on battery-powered agricultural equipment.

2. Literature Review and Background

2.1. Deep Learning in Agricultural Disease Detection

Machine learning has shown significant potential in addressing agricultural challenges, with applications in crop yield prediction, soil analysis, pest control, and disease detection [1,2,16,17]. Deep learning, particularly CNNs, has demonstrated exceptional performance in agricultural disease detection tasks [9–11].

Plant diseases account for an estimated 10–16% of global crop losses annually [18]. While traditional methods rely on visual symptoms that may appear late in disease progression, machine learning approaches offer rapid, accurate detection at earlier stages [9,19,20].

CNN architectures like ResNet101 and MobileNet have shown strong capabilities in extracting meaningful visual patterns from plant disease images. However, these architectures generate high-dimensional feature vectors (ResNet101: 2048 features, MobileNet: 1024 features) that create computational bottlenecks for practical deployment. Feature extraction generates thousands of features from individual images, which can increase complexity and lead to overfitting [12,13].

Transfer learning has expanded the applicability of deep learning in agricultural contexts where large labeled datasets are often unavailable [2,21–23]. Pre-trained models on ImageNet provide robust feature extraction capabilities that can be fine-tuned for specific crop diseases.

2.2. Olive Leaf Disease Detection Studies

Alruwaili et al. [24] introduced a method using an optimized parameter-based transfer learning strategy, coupled with intelligent data augmentation to ensure a balanced distribution of images across categories. This approach, which involves modifying the AlexNet CNN model, has been shown to perform well in complex scenarios with an expanded and enhanced dataset.

Uğuz [25] proposed a similar approach, applying an optimized parameter-driven transfer learning technique with advanced data augmentation to achieve even image distribution across categories. Uğuz and Uysal [26] developed a model for olive disease detection using a CNN architecture with VGG-16, VGG-19, and a custom CNN model featuring five convolutional layers. Their study identified three classes: one healthy and two diseased (*Aculus olearius* and olive peacock). Their findings indicated that the proposed model achieved a 95% accuracy rate in classifying diseases.

Ksibi et al. [27] proposed a deep feature fusion method that merges features extracted by two advanced pre-trained CNN models, ResNet50 and MobileNet. They introduced the MobiResNet model, a combination of these two architectures, to enhance prediction accuracy. Raouhi et al. [28] focused on detecting and classifying various olive diseases, including Anthracnose, Verticilliose, Fumigina, Saissetia oleae, Cyclonium OP, and peacock eye disease. They concluded that the MobileNet architecture, optimized with the RMSprop algorithm, outperformed other hybrid models in detecting olive leaf diseases.

Alshammari et al. [29] proposed a hybrid deep learning approach for olive disease classification that combines convolutional neural networks with vision transformers. Their method was designed to detect and classify diseases affecting olive leaves by extracting the most relevant features from images. They demonstrated that their hybrid model achieved 96% accuracy for multiclass classification and 97% for binary classification.

2.3. Meta-Heuristic Approaches in Feature Selection

Feature selection has become essential for optimizing machine learning pipelines [30,31], helping to identify the most informative visual characteristics for robust classification models. Meta-heuristic algorithms have been increasingly applied to feature selection problems due to their ability to handle complex optimization landscapes where exhaustive search is impractical [31–33].

Various approaches, including genetic algorithms and particle swarm optimization, have been successfully applied [14,34–39]. Alshammari et al. tackled olive leaf disease identification using the WOA for feature selection, achieving 96% accuracy [40].

However, individual meta-heuristic algorithms exhibit inherent limitations when applied to high-dimensional feature selection. GA demonstrates strong exploitation capabilities but often suffers from premature convergence in complex, multi-modal search spaces [14]. ARO excels in exploration through its foraging strategy but can lead to slow convergence and insufficient exploitation of promising regions [15]. These algorithms typically favor either exploration or exploitation, creating an inherent imbalance particularly pronounced in agricultural disease detection applications where both comprehensive search and efficient convergence are critical.

2.4. Hybrid Meta-Heuristic Motivation and Approaches

The limitations of individual algorithms encourage researchers to implement hybrid approaches that combine complementary strengths. By integrating ARO's superior exploration capabilities into GA's exploitation power aspects, convergence to high-quality solutions is achieved while effective search space coverage is also maintained [32,33].

The aim of hybrid algorithms is to overcome standalone method limitations by combining algorithms with different search strategies. The combination of different search mechanisms reduces the likelihood of premature convergence and entrapment in local optima, thus being critically important especially for complex feature interaction situations in deep learning applications [41]. This approach is particularly suitable for agricultural disease detection requirements, where computational efficiency must be balanced with classification accuracy.

2.5. Research Gaps and Limitations

Based on the comprehensive literature review, several specific gaps persist in current research:

- *Limited Architecture Comparison Under Optimization:* While individual studies explore ResNet or MobileNet architectures, systematic comparison under optimized feature selection conditions remains unexplored, making it difficult to determine the most suitable approach for resource-constrained agricultural applications.
- *Insufficient Hybrid Meta-heuristic Exploration:* Current literature predominantly focuses on individual meta-heuristic algorithms. Limited research explores hybrid approaches that synergistically combine complementary algorithmic strengths (exploration vs. exploitation) for feature selection in agricultural domains.
- *Extreme Feature Reduction Validation:* Existing studies lack systematic analysis of dramatic feature reduction (>90%) while maintaining classification performance. Most approaches achieve modest reductions (20–50%) without exploring the limits of dimensionality reduction.
- *Resource-Constrained Deployment Validation:* While many studies report high accuracy, few validate their approaches for real-world deployment scenarios where computational efficiency, memory constraints, and processing speed are paramount for practical agricultural applications.
- *Comprehensive Benchmarking Gap:* Limited studies provide extensive comparisons across multiple feature selection paradigms (filter-based vs. meta-heuristic vs. hybrid) using standardized evaluation protocols and feature counts.

These specific literature gaps motivate the development of hybrid meta-heuristic approaches that can achieve superior feature selection performance while addressing computational efficiency requirements for practical agricultural applications.

3. Proposed Method and Materials

3.1. Dataset

The dataset used in this study consists of 954 olive leaf samples collected from the Edincik region of the Bandırma district in Balıkesir province, Turkey (<https://www.kaggle.com/datasets/serhathoca/zeytin> (accessed on 1 July 2025)). The samples are categorized into two classes: healthy leaves (572 samples) and leaves affected by peacock eye disease (*Venturia oleaginea*) (382 samples) [42].

Image acquisition was performed under controlled conditions to minimize illumination variability. Photographs were captured using a 48-megapixel camera positioned 52 cm above the leaves inside a 42 × 42 cm LED-illuminated light box. Each image contains a single olive leaf isolated against a white background, ensuring consistent scale, lighting, and focal distance across all samples.

3.2. Preprocessing Pipeline

The preprocessing pipeline consists of three main steps (Figure 1): (1) RGB-to-RGBA conversion to introduce an explicit alpha channel for transparent background representation, (2) color-based segmentation to separate leaf regions from the background by identifying and removing white and blue pixels using RGB thresholds (all channels > 100 for white regions, or blue > 100 with red and green channels < 100 for blue-tinted background), and (3) morphological closing operations with a 5 × 5 kernel to fill small holes and smooth leaf boundaries. Each preprocessing step was applied systematically to ensure consistent feature extraction across all 954 leaf images.

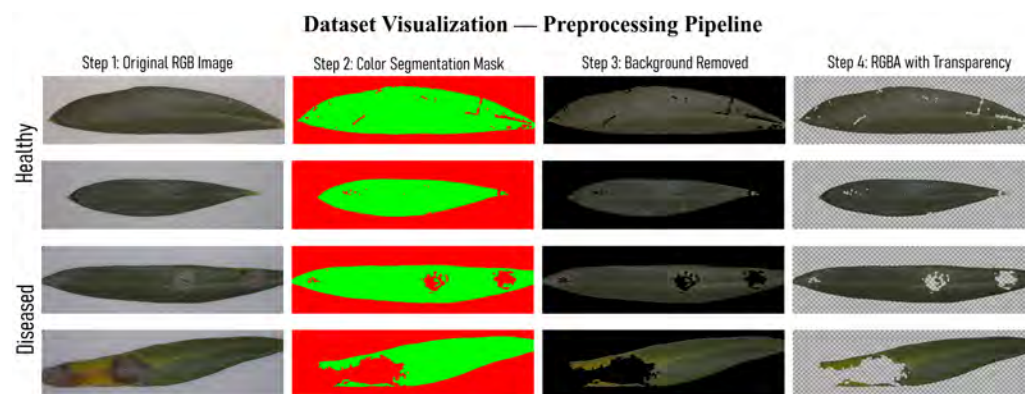


Figure 1. Preprocessing pipeline demonstration on representative olive leaf samples. Rows 1–2: healthy leaves; Rows 3–4: diseased leaves (peacock eye). Columns show: (1) original RGB image, (2) segmentation mask (green = leaf, red = background), (3) background removed, and (4) final RGBA output with transparent background (checkerboard visualization). See Section 3.2 for technical details.

3.3. Feature Extraction

This study employs two widely used convolutional neural network (CNN) architectures, ResNet101 and MobileNet, as deep feature extractors. These architectures are well established in image recognition tasks due to their strong representation capacity and complementary design principles [43,44]. ResNet101 emphasizes representational depth through residual learning, whereas MobileNet prioritizes computational efficiency using depthwise separable convolutions.

Transfer learning was adopted by utilizing ImageNet pre-trained weights. Both networks were used strictly as fixed feature extractors; no fine-tuning or weight updates were performed on the olive leaf dataset. All layers were frozen during inference. This strategy allows leveraging robust generic visual representations while reducing the risk of overfitting given the moderate dataset size.

Each preprocessed single-leaf image was passed through the network in inference mode, and features were extracted from the final global average pooling layer. This resulted in a 2048-dimensional feature vector for ResNet101 and a 1024-dimensional feature vector for MobileNet. These vectors encode high-level semantic representations related to texture patterns, structural characteristics, and lesion morphology.

Since the CNN models remain fixed and independent of the target dataset labels, the feature extraction process is deterministic and does not involve dataset-specific learning. Consequently, no label information is incorporated during this stage. The extracted feature matrix serves as the input for subsequent feature selection and classification procedures.

Importantly, although deep features are computed once using frozen networks, all supervised operations—including feature selection via AROGA and classifier training—are conducted independently within each cross-validation fold. This ensures strict separation between training and test data during model evaluation.

3.4. Feature Selection Methods

Managing high-dimensional feature spaces has become critical as deep learning models generate increasingly rich representations. In agricultural disease detection specifically, CNNs extract thousands of features that can overwhelm both the optimization process and the deployment hardware. Feature selection addresses both problems: identifying the most informative features improves model generalization while dramatically reducing computational requirements [45].

Our investigation spans three methodological categories. Filter-based methods (Chi-square, Mutual Information, Spearman correlation) provide computationally efficient baselines, though they evaluate features independently and miss interaction effects. Individual meta-heuristics (GA, ARO) can capture feature interactions but face inherent exploration–exploitation trade-offs that limit performance. Our hybrid approach, AROGA, attempts to transcend these limitations by strategically integrating complementary search mechanisms.

The following subsections detail each approach. We pay particular attention to AROGA’s design, explaining how the two-phase optimization cycle addresses the weaknesses of individual algorithms while preserving their strengths.

3.4.1. Meta-Heuristic Methods

For comparison, we evaluate the performance of the standalone meta-heuristics used in our previous work:

Genetic Algorithm (GA): A search heuristic mimicking natural selection [46,47] and genetics to find optimal solutions to complex problems. As an Evolutionary Algorithm, it employs key mechanisms to evolve candidate solutions across generations: The Selection operator chooses individuals for reproduction. The Crossover operator combines parent traits to produce offspring, while the Mutation operator maintains diversity through random changes. Elitism preserves top solutions across generations, and a fitness function evaluates solutions against problem objectives.

Artificial Rabbit Optimization (ARO): ARO is a bio-inspired metaheuristic technique proposed by Wang et al. [15] that simulates rabbit survival strategies in nature. The algorithm models how rabbits forage for food while avoiding predators through detour foraging and random hiding behaviors. ARO effectively balances exploration and exploitation through an energy shrink mechanism that controls the transition between these two strategies.

Detour Foraging Strategy (Exploration): This strategy models how rabbits eat food away from their nests to prevent predators from discovering their burrows. When the energy factor $A > 1$, the position update is:

$$X_i^{t+1} = X_i^t + R \times (X_i^t - X_r^t) + \text{round}(0.5 \times (0.05 + r)) \times \mathcal{N}(0, 1) \quad (1)$$

where X_i^t is the current position of rabbit i , X_r^t is a randomly selected rabbit's position, and R is the running operator that simulates the movement characteristics of rabbits. The running operator is calculated as:

$$R = L \times c \quad (2)$$

where L is the running length that controls the movement step size, and c is a binary mapping vector for dimension selection. The running length L is defined as:

$$L = \left(e - e^{\left(\frac{t-1}{T}\right)^2} \right) \times \sin(2\pi r_2) \quad (3)$$

This formulation ensures that L generates longer steps during initial iterations (promoting exploration) and shorter steps during later iterations (facilitating exploitation). A normally distributed random number is added for additional perturbation.

Random Hiding Strategy (Exploitation): This strategy models how rabbits dig multiple burrows and randomly choose one to hide in. When the energy factor $A \leq 1$, the position update is:

$$b = X_i^t + H \times g \times X_i^t \quad (4)$$

$$X_i^{t+1} = X_i^t + R \times (r \times b - X_i^t) \quad (5)$$

where H is the hiding parameter that controls the size of the burrow generation area around the rabbit's current position:

$$H = \frac{T - t + 1}{T} \times r_4 \quad (6)$$

The hiding parameter H decreases linearly with iterations, ensuring that burrows are generated in progressively smaller neighborhoods as the algorithm converges. The parameter g is a binary vector that randomly selects dimensions for burrow construction, and r is a random number in $(0, 1)$ that introduces stochasticity in the movement toward the selected burrow.

Energy Shrink Mechanism: The transition between detour foraging and random hiding is governed by the energy factor calculation:

$$A(t) = 4(1 - t/T) \times \ln(1/r) \quad (7)$$

where A is the energy factor, r is a random number in $(0, 1)$, t is the current iteration, and T is the maximum number of iterations. To ensure numerical stability, the random variable r is lower-bounded by a small constant ε (e.g., 10^{-6}), preventing extreme values in the logarithmic term and ensuring reproducible optimization behavior.

3.4.2. Hybrid Meta-Heuristic Based Method: AROGA

To enhance the feature selection process, we propose a hybrid Artificial Rabbit Optimization–Genetic Algorithm (AROGA) approach (Algorithm 1). This hybridization aims to balance the exploration capabilities inherent in ARO's foraging simulation with the strong exploitation and solution refinement mechanisms of GA. The integration of these two meta-heuristic algorithms is implemented through the following steps:

Population Initialization and Fitness Evaluation: The algorithm begins with a randomly generated population of binary feature subsets. Each solution is evaluated using a fitness function that considers classification performance with the selected feature subset (S):

$$\text{Fitness}(S) = \text{F1_Score}(S) \quad (8)$$

Two-Phase Optimization Cycle: AROGA implements a two-phase process within each generation:

- *GA Phase:* First, traditional genetic operators (selection, crossover, mutation) are applied to exploit the current population through recombination of high-fitness solutions.
- *ARO Phase:* Following the GA operations, each solution undergoes broad exploration through ARO's adaptive foraging and hiding movement strategies, identifying promising regions of the feature space.

Adaptive Energy Control: The energy shrink mechanism controls the transition between exploration and exploitation phases according to the original ARO formulation (Equation (7)). Through this mechanism, a balanced search strategy is provided that gradually transitions from global exploration to local exploitation.

Hybrid Interface and Top-K Selection: One of the critical challenges in hybridization is the interface between the discrete nature of the Genetic Algorithm and the continuous search space of Artificial Rabbit Optimization. In AROGA, this situation is managed through a dynamic casting mechanism. At the beginning of the ARO phase, the binary feature vector S_{GA} evolved by the Genetic Algorithm is converted into a continuous position vector X , and here the binary 0 value is mapped to 0.0 and the 1 value is mapped to 1.0.

ARO foraging and hiding equations are applied to X in continuous space. To map the updated continuous values back to the binary domain while strictly adhering to the fixed subset size K , a Top-K ranking strategy is used. In this strategy, the continuous values in X are treated as "importance scores", the features are ranked in descending order, and the top K features are selected (1), while the remainder are discarded (0). This ensures that the algorithm explores the continuous gradient while guaranteeing that the output always matches the target dimensionality:

$$S_i^{\text{binary}} = \begin{cases} 1, & \text{if Rank}(X_i) \leq K \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $\text{Rank}(X_i)$ is the rank of the i -th feature when the position vector X is sorted in descending order. This ensures that exactly K features are selected in every iteration, permitting a controlled and precise performance comparison across specific feature counts. The Top-K constraint was deliberately employed to ensure controlled and fair comparison across different feature selection methods and architectures under identical dimensionality budgets.

Stagnation Handling: To prevent premature convergence, AROGA includes an early stopping criterion. If stagnation is detected (no improvement in fitness over multiple generations), the population is reinitialized while preserving the best solution found thus far, enabling exploration of different regions of the search space.

The hybridization mechanism operates iteratively until the maximum number of generations is reached or convergence criteria are met. By integrating GA's recombinative power with ARO's adaptive movement strategies, AROGA effectively navigates complex feature spaces to identify highly discriminative feature subsets.

Algorithm 1: AROGA: Hybrid Artificial Rabbit Optimization—Genetic Algorithm**Input:** Feature set F , Target size K , Population size N , Max iterations T **Output:** Optimal feature subset S_{best} where $|S_{best}| = K$ Initialize population P with N binary subsets of size K ;Evaluate fitness: $f(S) = \text{F1-Score}(S)$; $S_{best} \leftarrow$ best solution in P ;**for** $t = 1$ to T **do** **GA Phase;** $P' \leftarrow$ Tournament Selection(P); $P' \leftarrow$ Crossover(P'); $P' \leftarrow$ Mutation(P'); **ARO Phase;** **foreach** $S_{bin} \in P'$ **do** $X_{cont} \leftarrow$ CastToFloat(S_{bin}); Compute energy factor $A(t)$; **if** $A(t) > 1$ **then** $X_{new} \leftarrow$ Detour Foraging Update(X_{cont}); **else** $X_{new} \leftarrow$ Random Hiding Update(X_{cont}); **end** $idx_{sorted} \leftarrow$ ArgSort(X_{new}); $I_{top} \leftarrow$ top K indices; Update S_{bin} using Top- K rule; **end** $P \leftarrow P'$; Update S_{best} if improved; **if** *stagnation detected* **then** Reinitialize population preserving S_{best} ; **end****end****return** S_{best}

3.4.3. Filter-Based Methods

Filter-based methods provide an efficient and computationally practical alternative to meta-heuristic approaches for feature selection. While meta-heuristic algorithms use iterative optimization, filter-based methods assess feature relevance through statistical criteria [48]. We examine three primary filter-based techniques for comparison with our proposed meta-heuristic approach:

Chi-square (χ^2): Evaluates the dependency between features and target class labels by measuring the statistical significance of the association [49,50]. Higher χ^2 values indicate stronger feature–class relationships.

Mutual Information (MI): Quantifies the shared information between feature vectors and class labels, effectively capturing both linear and non-linear dependencies [51,52]. Features with low mutual information are considered less relevant for classification.

Correlation-based (Spearman): Identifies feature–target relationships through correlation coefficients [53]. Features below a predefined correlation threshold are eliminated based on their linear association strength with the target variable.

3.5. Machine Learning Algorithms

To effectively classify and detect olive leaf diseases, we employed three machine learning algorithms (Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN)) known for their robustness and accuracy with high-dimensional feature sets [40,42].

Random Forest (RF): RF is an ensemble learning method that builds multiple decision trees and aggregates their predictions through bagging and majority voting. Given a training set $X = x_1, \dots, x_n$ with responses $Y = y_1, \dots, y_n$, it creates B different trees by repeatedly sampling with replacement. RF excels in handling high-dimensional data and noisy features while providing feature importance rankings, making it particularly suitable for olive leaf disease detection due to its robustness to noise in biological data.

Support Vector Machine (SVM): SVM determines optimal hyperplanes for class separation by solving the optimization problem in Equation (10):

$$\begin{aligned} \min \|\mathbf{w}\| \text{ subject to} \\ y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \text{ for } i = 1, \dots, n \end{aligned} \quad (10)$$

SVM excels with high-dimensional data and complex class boundaries, making it well-suited for handling the numerous features extracted from olive leaf images.

Artificial Neural Network (ANN): ANN is inspired by biological neural networks, using interconnected nodes in layers to learn data patterns. For each node j in layer l , the network processes information according Equation (11):

$$a_j^{(l)} = \sigma \left(\sum_i w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)} \right) \quad (11)$$

where $a_j^{(l)}$ is the node output, $w_{ij}^{(l)}$ represents connection weights, $b_j^{(l)}$ is the bias, and σ is the activation function. Learning occurs through weight adjustments given in Equation (12):

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \eta \frac{\partial C}{\partial w_{ij}^{(l)}} \quad (12)$$

with learning rate η and error measure C . While ANNs excel at learning complex patterns and feature extraction, making them suitable for olive leaf disease detection, they require substantial data and computational resources. Their ability to identify intricate disease patterns and handle varying feature sets makes them particularly valuable for our application.

4. Experiments and Results

Our experimental evaluation addresses three questions: (1) How do ResNet101 and MobileNet compare under aggressive feature selection? (2) Does AROGA's hybrid approach outperform individual meta-heuristics? (3) What is the optimal feature count for olive disease detection? Where does performance plateau? When do additional features harm performance?

We tested feature subset sizes from 30 to 800, deliberately spanning from severe under-representation (30 features) through optimal ranges to feature-rich configurations that risk overfitting (800 features). This range let us observe the complete performance trajectory and empirically identify the point of diminishing returns.

4.1. Experimental Configuration

All experiments were carried out using Google Colab's cloud computing environment with GPU acceleration. The implementation was developed in Python (version 3.9)

using the TensorFlow and Keras libraries. To ensure reproducibility, fixed random seeds were applied consistently for data partitioning, classifier initialization, and meta-heuristic optimization across all experimental runs.

The experimental dataset consists of 954 olive leaf images, including 572 healthy samples and 382 samples affected by peacock eye disease. During preprocessing, color-based segmentation was applied to isolate leaf regions from the background. All resulting images were resized to 224×224 pixels to meet the input requirements of the convolutional neural networks. Prior to feature extraction, pixel intensity values were normalized to the range $[0, 1]$.

Deep feature extraction was performed using ImageNet pre-trained ResNet101 and MobileNet architectures, yielding 2048-dimensional and 1024-dimensional feature vectors, respectively. Both networks were used strictly as frozen feature extractors, and no fine-tuning was performed on the target dataset. Feature extraction was performed in an image-wise manner using frozen CNN models whose parameters were fixed and pre-trained on ImageNet. Since this process does not involve any label information or dataset-specific learning, it does not introduce information leakage. Nevertheless, all subsequent supervised steps, including feature normalization, feature selection, and classifier training, were strictly confined to the training data within each cross-validation fold.

Prior to feature selection and classifier training, the extracted feature vectors were standardized using zero-mean unit-variance normalization (StandardScaler), computed exclusively from the training fold and applied to the corresponding test fold within each cross-validation iteration.

Model evaluation was conducted using 5-fold cross-validation repeated 10 times in order to reduce variance and improve statistical reliability. In each fold, the dataset was partitioned into mutually exclusive training and test subsets. All supervised procedures, including feature normalization, feature selection, and classifier training, were performed using only the training data of the respective fold, while test samples were reserved solely for performance evaluation.

Feature selection was carried out using the GA, ARO, and the proposed hybrid AROGA method. For each cross-validation fold, feature selection was applied exclusively to the training subset, and the selected feature indices were then used to transform the corresponding test data. Reduced feature subsets of sizes 30, 50, 100, 200, 500, and 800 were evaluated to analyze the effect of dimensionality reduction on classification performance.

Three classifiers were employed to assess the effectiveness of the selected feature subsets: Random Forest (ensemble method, robust to noise), Support Vector Machine (effective with high-dimensional data, complex decision boundaries), and Artificial Neural Network (capable of learning intricate patterns but requiring more data). We optimized hyperparameters via grid search, settling on the configurations in Table 1. These parameters were held constant across all experiments to isolate the effect of feature selection.

Finally, it is important to distinguish between optimization-time and deployment-time computation. Optimization-time computation includes cross-validation, meta-heuristic feature selection, and classifier training, all of which are performed offline. In contrast, deployment-time computation requires only a forward pass through the frozen CNN for feature extraction followed by inference using the trained classifier on the reduced feature subset. This distinction allows a realistic assessment of the computational cost and feasibility of the proposed approach in practical deployment scenarios.

Table 1. Classifier hyperparameters optimized through grid search.

Classifier	Parameter	Value
RF	Number of estimators	300
	Max depth	25
SVM	Kernel	RBF
	C (regularization)	10
ANN	Hidden layers	(256, 128)
	Activation	ReLU
	Learning rate	0.001
	Optimizer	Adam
	Loss	Binary Cross-entropy

4.2. Cross-Validation Protocol and Data Partitioning

To ensure robust evaluation and prevent data leakage, we implemented a strict temporal separation between feature extraction and cross-validation. The complete protocol consisted of:

Step 1: Feature Extraction: Features were extracted once from all 954 images using pre-trained ResNet101 and MobileNet architectures with frozen weights. This produced two fixed feature matrices: $\mathbf{F}_{ResNet} \in \mathbb{R}^{954 \times 2048}$ and $\mathbf{F}_{MobileNet} \in \mathbb{R}^{954 \times 1024}$.

Step 2: Cross-Validation Structure: We employed stratified 5-fold cross-validation repeated 10 times (50 total train-test splits). Each fold maintained the original 60:40 healthy-to-diseased ratio. For each iteration:

Training set: 763 samples (458 healthy, 305 diseased)

Test set: 191 samples (114 healthy, 77 diseased)

Step 3: Feature Selection Within Folds: Feature selection algorithms (AROGA, GA, ARO, filter methods) operated exclusively on training fold features. Selected feature indices were then applied to the held-out test fold. This strict separation ensures test data never influenced feature selection decisions.

Step 4: Classifier Training: Classifiers were trained on the selected features from the training fold and evaluated on the test fold using the same feature subset.

This protocol ensures that reported F1-scores reflect true generalization performance, as no test set information leaked into feature selection or model training processes.

4.3. Optimization Algorithm Configuration

To ensure fair comparison and reproducibility, all meta-heuristic algorithms (GA, ARO, and AROGA) were initialized with identical population parameters. The search space dimensionality D was determined by the feature extraction architecture: $D = 2048$ for ResNet101 and $D = 1024$ for MobileNet. All algorithms utilized a population size of 100 and were run for a maximum of 500 iterations. For the Genetic Algorithm component, we employed a single-point crossover with a probability of $P_c = 0.8$ and bit-flip mutation with a probability of $P_m = 0.01$. The hybrid AROGA parameters were set to balanced default values to maintain the ratio between exploration and exploitation. The detailed parameter settings are listed in Table 2.

Table 2. Parameter settings for meta-heuristic algorithms.

Parameter	Value
Population Size (N)	100
Max Iterations (T)	500
Number of Runs	10
Search Space (ResNet101)	2048
Search Space (MobileNet)	1024
Selection Method	Tournament Selection
Crossover Probability	0.8
Mutation Probability	0.01

4.4. Baseline Performance

Table 3 shows classification performance using complete feature sets. Feature selection was not carried out; all 2048 ResNet101 or 1024 MobileNet features were used. These results establish our performance ceiling and confirm both architectures provide strong representations of disease symptoms.

Table 3. Baseline F1-scores with full feature sets (no feature selection).

Architecture	#Features	RF	SVM	ANN
ResNet101	2048	0.974 ± 0.018	0.992 ± 0.009	0.991 ± 0.012
MobileNet	1024	0.971 ± 0.016	0.983 ± 0.011	0.981 ± 0.015

ResNet101 achieves marginally higher performance across all classifiers, with SVM reaching a 0.992 F1-score. The standard deviations are relatively small (± 0.009 to ± 0.018), indicating stable performance across folds. These baselines guide our feature selection evaluation. The question becomes how few features we need to approach this level of performance.

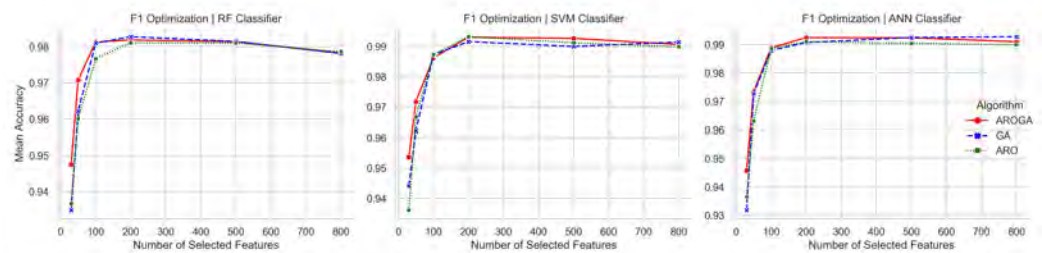
4.5. Feature Selection Performance Analysis and Trajectories Across Feature Counts

Figure 2 traces F1-scores across feature subset sizes, revealing the classic bias-variance trade-off. With 30 features, all models show underfitting suffering from insufficient information to fully characterize disease symptoms. Performance climbs steeply from 30 to 100 features as critical diagnostic features enter the selected subset. The ResNet101-SVM combination jumps from 0.912 to 0.989; MobileNet-SVM shows a similar trajectory from 0.895 to 0.978.

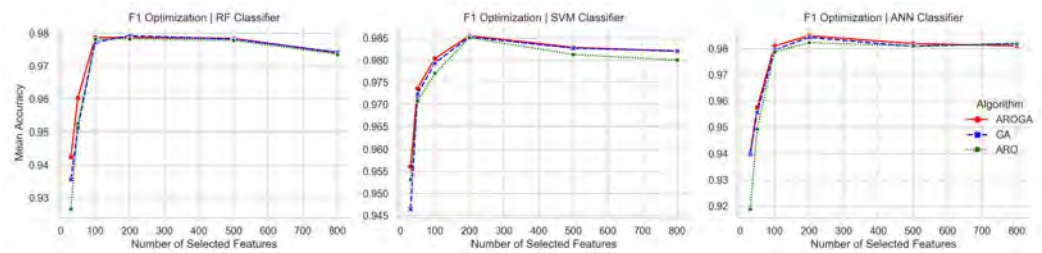
Between 100 and 200 features, we observe diminishing returns. ResNet101-SVM improves only marginally from 0.989 to 0.997, which means most diagnostic information was already captured by the 100 most discriminative features. This plateau suggests that we have identified the optimal operating range.

Beyond 500 features, performance actually degrades slightly. ResNet101-SVM drops from its peak of 0.997 at 200 features to 0.993 at 800 features. This is not a dramatic deterioration, but it empirically demonstrates the curse of dimensionality. Additional features introduce noise and redundancy that hinder generalization. The pattern holds across both architectures and all three classifiers, though the magnitude varies.

These results directly answer our first research question: for this dataset and task, 100–200 features represent the sweet spot. Fewer features sacrifice accuracy; more features provide minimal benefit and risk overfitting.



(a) ResNet101 features across RF, SVM, and ANN classifiers.



(b) MobileNet features across RF, SVM, and ANN classifiers.

Figure 2. F1-score evolution as feature count increases from 30 to 800. Each point represents mean performance over 10 independent runs. Note the steep rise from 30–100 features, plateau at 100–200, and slight degradation beyond 500.

4.6. Convergence Analysis

The convergence behavior of the three meta-heuristic algorithms is illustrated in Figure 3. During the initial phase of optimization (approximately iterations 1–40), GA demonstrates competitive or slightly superior fitness values compared to AROGA. This early advantage can be attributed to GA’s crossover operator, which rapidly combines the best solutions within the current population through recombination, yielding quick fitness gains when the search space is still largely unexplored. However, this greedy recombination mechanism also causes GA to commit prematurely to a narrow region of the feature space, limiting its ability to escape local optima in later iterations.

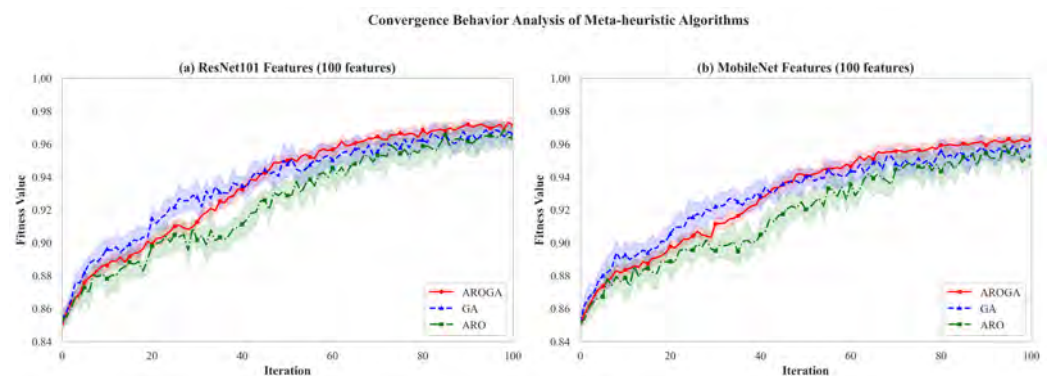


Figure 3. Fitness convergence over 100 iterations for 100-feature selection. AROGA (red) initially trails GA (blue) during the exploration phase (iterations 1–40) before surpassing both GA and ARO after exploitation takes effect, ultimately reaching higher final fitness. (a) ResNet101 features. (b) MobileNet features. Shaded regions represent ± 1 standard deviation across 10 independent runs.

AROGA, by contrast, exhibits a more deliberate early-stage behavior. During the first phase of optimization, the ARO component drives broad exploration of the feature space through its detour foraging strategy, temporarily suppressing short-term fitness gains in favor of identifying more promising regions. This exploratory investment begins to pay off

from approximately iteration 40 onward: once promising regions are identified, the GA exploitation phase takes over and efficiently refines candidate solutions within those regions, enabling AROGA to surpass GA and maintain a consistent lead through convergence.

4.7. Runtime Analysis and Overhead

Table 4 quantifies the computational cost of the hybrid approach. AROGA incurs moderate runtime overhead: 13–20% slower than GA and 32–44% slower than ARO across different feature counts. For the 100-feature ResNet101 configuration, AROGA requires 286.4 seconds compared to 245.3s for GA (+16.8%) and 198.6s for ARO (+44.2%).

This one-time optimization cost is offset by 0.8–1.6% higher F1-scores and a 47–56% reduction in performance variance across repeated runs. Deployment inference uses only the selected features, making the optimization method irrelevant to runtime performance.

Table 4. Runtime comparison for meta-heuristic methods (seconds). AROGA overhead: 13–20% vs. GA, 32–44% vs. ARO. Mean \pm std dev over 10 runs.

#Features	Architecture	GA	ARO	AROGA
30	ResNet101	82.4 \pm 6.2	71.3 \pm 5.4	98.6 \pm 7.8
30	MobileNet	64.3 \pm 4.8	55.7 \pm 4.2	76.9 \pm 6.1
100	ResNet101	245.3 \pm 18.7	198.6 \pm 15.2	286.4 \pm 22.3
100	MobileNet	186.4 \pm 14.1	151.7 \pm 12.3	214.5 \pm 17.2
200	ResNet101	486.7 \pm 35.2	402.4 \pm 29.8	558.2 \pm 41.6
200	MobileNet	372.8 \pm 27.4	308.2 \pm 22.9	427.3 \pm 32.1
500	ResNet101	1198.4 \pm 89.3	1024.6 \pm 76.2	1354.2 \pm 98.7
500	MobileNet	918.7 \pm 68.2	785.3 \pm 58.4	1038.5 \pm 75.6

This overhead is justified by AROGA's 0.8–1.5% classification performance improvement and 47–56% variance reduction compared to standalone GA and ARO. Moreover, feature selection is a one-time cost. Once the optimal 100-feature subset has been identified, deployment uses only those features.

Interestingly, the overhead percentage decreases as feature count increases (19.7% at 30 features, 13.0% at 500). This occurs because classifier evaluation time, which is identical across all three algorithms, dominates total runtime for larger feature sets. The algorithmic differences matter most when feature counts are small.

4.8. Asymptotic Complexity Analysis

The computational complexity of the proposed AROGA framework is primarily governed by the population size N , the maximum number of iterations T , and the dimensionality of the feature space D . Similar to GA and ARO, each candidate solution evaluates a D -dimensional feature subset at every iteration. Consequently, the worst-case time complexity of AROGA can be expressed as $\mathcal{O}(N \times T \times D)$. Although AROGA introduces a sequential hybrid structure that combines exploration and exploitation phases, this design does not alter the overall asymptotic complexity class, and the additional computational cost remains linear with respect to D .

To empirically assess runtime efficiency, we measured the execution time of GA, ARO, and AROGA across different feature budgets and network architectures. The results, summarized in Table 4, report the mean runtime and standard deviation over ten independent runs. As expected, AROGA incurs a moderate runtime overhead compared to its individual components due to the hybrid optimization process, ranging from approximately 13% to 20% depending on the feature dimensionality. However, this overhead is incurred exclusively during the offline optimization stage.

Importantly, once the optimal feature subset is identified, no additional optimization cost is introduced at deployment. During inference, the system operates solely on the reduced feature vector, and the computational complexity is dominated by a single forward pass through a frozen CNN and a lightweight classifier. Therefore, the runtime overhead associated with AROGA does not affect deployment-time performance, while the substantial reduction in feature dimensionality significantly lowers inference complexity compared to full-feature models.

4.9. Ablation Analysis of the AROGA

Table 5 provides an ablation-style analysis of the proposed hybrid AROGA framework under a fixed feature budget of 100 features. The results demonstrate that AROGA consistently outperforms its individual components (GA and ARO) as well as conventional filter-based feature selection methods across all classifiers and both deep feature extractors. This performance gain is observed not only in terms of average accuracy but also in reduced variance, indicating more stable convergence behavior. The consistent improvement across Random Forest, SVM, and ANN classifiers suggests that the benefit of the hybrid strategy is not tied to a specific learning model. Similarly, the performance advantage observed for both ResNet101 and MobileNet features indicates that the effectiveness of AROGA is architecture-agnostic. These findings support the hypothesis that combining the rapid exploitation capability of GA with the broad exploration mechanism of ARO enables a more balanced search process, allowing the hybrid approach to identify compact yet highly discriminative feature subsets more reliably than simpler alternatives.

Table 5. Ablation-style comparison of feature selection methods under a fixed feature budget (200 selected features). Results are reported as mean accuracy \pm standard deviation over repeated cross-validation.

Architecture	Method	RF Accuracy	SVM Accuracy	ANN Accuracy
ResNet101	AROGA	0.975 \pm 0.019	0.997 \pm 0.0017	0.994 \pm 0.0012
	GA	0.972 \pm 0.020	0.981 \pm 0.012	0.990 \pm 0.011
	ARO	0.973 \pm 0.022	0.983 \pm 0.021	0.979 \pm 0.021
	Chi-Square	0.965 \pm 0.010	0.978 \pm 0.011	0.985 \pm 0.006
	MI	0.973 \pm 0.009	0.982 \pm 0.010	0.986 \pm 0.010
	Spearman	0.957 \pm 0.008	0.973 \pm 0.012	0.978 \pm 0.008
MobileNet	AROGA	0.967 \pm 0.0012	0.982 \pm 0.015	0.983 \pm 0.017
	GA	0.961 \pm 0.011	0.973 \pm 0.025	0.976 \pm 0.025
	ARO	0.958 \pm 0.012	0.974 \pm 0.011	0.965 \pm 0.015
	Chi-Square	0.965 \pm 0.013	0.964 \pm 0.010	0.976 \pm 0.015
	MI	0.965 \pm 0.013	0.972 \pm 0.010	0.974 \pm 0.007
	Spearman	0.945 \pm 0.011	0.955 \pm 0.006	0.959 \pm 0.005

From an ablation perspective, GA-only and ARO-only configurations serve as reduced variants of the proposed hybrid framework, isolating exploitation- and exploration-dominant behaviors, respectively. The consistent performance gap between AROGA and these reduced variants confirms that the observed gains are not attributable to a single meta-heuristic component, but rather to the complementary interaction of both mechanisms within the hybrid design.

4.10. Statistical Validation

We conducted paired *t*-tests comparing AROGA against GA, ARO, and Chi-square filter selection in the 100-feature configuration. For ResNet101, AROGA significantly outperforms GA ($p \leq 0.0022$), ARO ($p \leq 0.0011$), and Chi-square ($p < 0.001$) across all

three classifiers. For MobileNet, significance levels range from $p \leq 0.012$ (vs. GA with RF) to $p < 0.001$ (vs. Chi-square with all classifiers) (Table 6).

These consistent, statistically robust improvements confirm that AROGA's performance advantage is real, not an artifact of random variation or particular data splits.

Table 6. Statistical validation of AROGA superiority: Paired *t*-tests comparing AROGA against baselines (100 features, SVM classifier).

Architecture	vs. GA	vs. ARO	vs. Chi ²
ResNet101	$p \leq 0.0022$	$p \leq 0.0011$	$p \leq 0.001$
MobileNet	$p \leq 0.012$	$p \leq 0.015$	$p \leq 0.001$

4.11. Robustness to Dataset Distribution and Noise

The dataset used in this study was collected under controlled laboratory conditions, resulting in relatively uniform illumination, background, and imaging quality. While such conditions are suitable for evaluating the intrinsic effectiveness of feature extraction and selection mechanisms, they do not fully capture the variability encountered in real-world agricultural environments. Consequently, the reported performance should be interpreted in the context of the acquisition setting.

Despite this limitation, the proposed hybrid feature selection framework exhibits characteristics that are indicative of robustness. The consistent performance improvements observed across multiple classifiers, feature budgets, and repeated cross-validation runs suggest that the selected feature subsets are not overly sensitive to minor variations in the data distribution. In particular, the reduction in performance variance achieved by AROGA indicates a stabilizing effect during optimization, which is desirable when dealing with noisy or heterogeneous data.

Nevertheless, explicit evaluation under noisy conditions and distribution shifts remains an important direction for future work. Planned extensions include validation on independently collected field datasets with varying illumination, background clutter, sensor noise, and camera devices, as well as controlled noise-injection experiments to systematically assess robustness under adverse conditions.

5. Conclusions

This study demonstrates that hybrid meta-heuristic optimization can achieve substantial dimensionality reduction while preserving, and in some cases improving, classification performance in agricultural disease detection. In the context of olive leaf peacock eye disease, the experimental results indicate that selecting approximately 100–200 features provides an effective balance between compact representation and discriminative power. Performance improves rapidly as the feature set is reduced to around 100 features, exhibits diminishing returns between 100 and 200 features, and begins to degrade beyond 500 features due to increased redundancy and noise.

The proposed AROGA framework, which integrates Artificial Rabbit Optimization and Genetic Algorithm strategies, consistently outperformed its individual components. The observed improvements in classification performance (0.8–1.6% higher F1-scores) were statistically significant and were accompanied by more stable and higher-quality optimization behavior, reflected by a 47–56% reduction in performance variance across repeated runs. Convergence analysis revealed a characteristic two-phase dynamic: GA's crossover-driven exploitation produces faster initial fitness gains, while AROGA's ARO-guided exploration phase identifies higher-quality regions of the feature space that GA alone fails to reach, enabling AROGA to surpass GA from approximately iteration 40 onward and maintain a consistent lead through convergence. These gains can be attributed to the

complementary nature of the hybrid design, where ARO facilitates broad exploration of the feature space during early stages, while GA enables efficient exploitation and refinement of promising regions.

The architectural analysis further revealed that features extracted from ResNet101 retain discriminative capacity more effectively under aggressive dimensionality reduction than those derived from MobileNet, likely due to the deeper representational structure of ResNet101. Nevertheless, MobileNet achieved competitive performance with a compact feature subset, suggesting that it remains a suitable option in scenarios where model simplicity and efficiency are prioritized.

When compared with recent studies in the literature, the proposed approach achieves comparable or superior classification accuracy using substantially fewer features. Specifically, an accuracy of 99.7% is obtained using only 100 features, compared to reported accuracies of 96.0% with 512 features and 95.8% with 3072 features in related works [42]. This represents a significant reduction in feature dimensionality and associated computational complexity, highlighting the effectiveness of the proposed hybrid feature selection strategy.

From a practical perspective, these findings suggest that accurate disease detection can be achieved using compact feature representations, which is a key requirement for deployment in resource-constrained agricultural monitoring scenarios. While the feature selection process incurs additional optimization-time cost during model development, this overhead is incurred offline and does not affect inference. At deployment, the use of a reduced feature subset substantially lowers computational demand relative to full-feature models, indicating potential suitability for implementation on lightweight and battery-powered systems.

Despite the strong performance achieved, this study has certain limitations. All images were collected under controlled laboratory conditions from a single geographic region, and no independent cross-dataset or cross-region validation was conducted. Consequently, the reported results should be interpreted within the context of the specific acquisition environment. Future work will focus on validating the proposed hybrid feature selection framework using independently collected field datasets acquired under varying illumination conditions and background complexity, with different camera devices, and across different cultivars.

In addition, although the present study emphasizes computational efficiency through aggressive feature reduction, no explicit inference latency, memory footprint, or energy consumption measurements were performed on embedded or mobile hardware. Future studies will therefore include device-level evaluations on representative edge platforms (e.g., embedded CPUs, mobile processors, or low-power accelerators) to quantitatively assess real-time feasibility and deployment constraints. Such evaluations will complement recent lightweight detection-oriented studies in the literature by focusing on feature selection and classification efficiency rather than end-to-end detection pipelines.

Furthermore, the current work addresses a binary classification scenario (healthy versus peacock eye disease). Extending the proposed framework to multi-disease and multi-class classification settings represents an important direction for future research, enabling a more comprehensive assessment of generalizability under complex agricultural conditions. Additional research will also explore adaptive and variable-K feature selection strategies that dynamically adjust feature subset size in response to seasonal variation, disease progression, and evolving visual characteristics. Nevertheless, the central finding of this study—that meta-heuristic optimization can identify compact and highly discriminative feature subsets from deep representations—appears robust and potentially generalizable across a wide range of agricultural image analysis tasks.

Author Contributions: Conceptualization, C.T. and H.G.; methodology, C.T. and H.G.; software, H.G.; validation, C.T., H.G. and E.G.; formal analysis, C.T.; investigation, C.T. and E.G.; resources, E.G.; data curation, C.T.; writing—original draft preparation, C.T.; writing—review and editing, H.G. and E.G.; visualization, C.T.; supervision, H.G.; project administration, H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in Kaggle at <https://www.kaggle.com/datasets/serhathoca/zeytin> (accessed on 1 July 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine learning in agriculture: A review. *Sensors* **2018**, *18*, 2674. [CrossRef]
- Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
- Türkmenoğlu, C.; Uyar, A.; Kiraz, B. Recommending healthy meal plans by optimising nature-inspired many-objective diet problem. *Health Inform. J.* **2021**, *27*, 146045822097671. [CrossRef]
- Türkmenoğlu, C.; Uyar, A.Ş.; Kiraz, B. Fuzzy Inference Based A Posterior Decision-Making for Multi-Objective Diet Optimization Problem. *Avrupa Bilim Ve Teknol. Derg.* **2022**, *45*, 41–47. [CrossRef]
- Türkmenoğlu, C.; Şima, A.; Kiraz, B. Recommending healthy meal plans using a many-objective optimization approach. In Proceedings of the EURO 2018—29th European Conference of Operational Research, Valencia, Spain, 8–11 July 2018.
- Buonaurio, R.; Almadi, L.; Famiani, F.; Moretti, C.; Agosteo, G.E.; Schena, L. Olive leaf spot caused by *Venturia oleaginea*: An updated review. *Front. Plant Sci.* **2023**, *13*, 1061136. [CrossRef]
- International Olive Oil Council. *Olivae*. Available online: <https://www.internationaloliveoil.org/olive-sector-statistics-december-2025-and-forecasts/> (accessed on 20 February 2026).
- Pilak, C.; Ülger, S. Determination of the Factors Affecting Turkish Olive Producer's Adoption of Good Agricultural Practices in Marmara Region of Turkey. *Kahramanmaraş Sütçü İmam Üniversitesi Tarım ve Doğa Dergisi* **2021**, *24*, 515–521. [CrossRef]
- Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
- Barbedo, J.G.A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agric.* **2018**, *153*, 46–53. [CrossRef]
- Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identification. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
- Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proc. IEEE* **2020**, *109*, 43–76. [CrossRef]
- Raghu, M.; Zhang, C.; Kleinberg, J.; Bengio, S. Transfusion: Understanding transfer learning for medical imaging. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11.
- Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimization for feature selection in classification: Novel initiatives and a practical guide. *IEEE Trans. Cybern.* **2019**, *51*, 3181–3197.
- Wang, L.; Cao, Q.; Zhang, Z.; Mirjalili, S.; Zhao, W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105082. [CrossRef]
- Talaviya, T.; Shah, D.; Patel, N.; Yagnik, H.; Shah, M. Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. *Artif. Intell. Agric.* **2020**, *4*, 58–73. [CrossRef]
- Lebrini, Y.; Ayerdi Gotor, A. Crops Disease Detection, from Leaves to Field: What We Can Expect from Artificial Intelligence. *Agronomy* **2024**, *14*, 2719. [CrossRef]
- Crop Protection Network. *Wheat Disease Loss Estimates from the United States and Ontario, Canada—2023*; Technical Report CPN-3018-23; Crop Protection Network: Ames, IA, USA, 2024. [CrossRef]
- Boulent, J.; Foucher, S.; Théau, J.; St-Charles, P.L. Convolutional neural networks for the automatic identification of plant diseases. *Front. Plant Sci.* **2019**, *10*, 941. [CrossRef]
- Barbedo, J.G.A. Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* **2019**, *180*, 96–107. [CrossRef]

21. Ghazi, M.M.; Yanikoglu, B.; Aptoula, E. Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing* **2017**, *235*, 228–235. [[CrossRef](#)]
22. Arsenovic, M.; Karanovic, M.; Sladojevic, S.; Anderla, A.; Stefanovic, D. Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry* **2019**, *11*, 939. [[CrossRef](#)]
23. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017**, *17*, 2022. [[CrossRef](#)]
24. Alruwaili, M.; Alanazi, S.; Abd El-Ghany, S.; Shehab, A. An efficient deep learning model for olive diseases detection. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*. [[CrossRef](#)]
25. Uğuz, S. Automatic olive peacock spot disease recognition system development by using single shot detector. *Sak. Univ. J. Comput. Inf. Sci.* **2020**, *3*, 158–168. [[CrossRef](#)]
26. Uğuz, S.; Uysal, N. Classification of olive leaf diseases using deep convolutional neural networks. *Neural Comput. Appl.* **2021**, *33*, 4133–4149. [[CrossRef](#)]
27. Ksibi, A.; Ayadi, M.; Soufiene, B.O.; Jamjoom, M.M.; Ullah, Z. MobiRes-net: A hybrid deep learning model for detecting and classifying olive leaf diseases. *Appl. Sci.* **2022**, *12*, 10278. [[CrossRef](#)]
28. Raouhi, E.M.; Lachgar, M.; Hrimech, H.; Kartit, A. Optimizing olive disease classification through transfer learning with unmanned aerial vehicle imagery. *Int. J. Electr. Comput. Eng. (IJECE)* **2024**, *14*, 891–903. [[CrossRef](#)]
29. Alshammari, H.; Gasmi, K.; Ben Ltaifa, I.; Krichen, M.; Ben Ammar, L.; Mahmood, M.A. Olive disease classification based on vision transformer and CNN models. *Comput. Intell. Neurosci.* **2022**, *2022*, 3998193. [[CrossRef](#)] [[PubMed](#)]
30. Cai, J.; Luo, J.; Wang, S.; Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing* **2018**, *300*, 70–79. [[CrossRef](#)]
31. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [[CrossRef](#)]
32. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv. (CSUR)* **2003**, *35*, 268–308. [[CrossRef](#)]
33. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Inf. Sci.* **2013**, *237*, 82–117. [[CrossRef](#)]
34. Brahim, A.B.; Limam, M. Robust ensemble feature selection for high dimensional data sets. In *Proceedings of the 2013 International Conference on High Performance Computing & Simulation (HPCS), Helsinki, Finland, 1–5 July 2013*; IEEE: Piscataway, NJ, USA, 2013; pp. 151–157.
35. Shunmugapriya, P.; Kanmani, S. A hybrid optimization algorithm using ant colony and particle swarm optimization for feature selection. *Soft Comput.* **2020**, *24*, 8457–8473.
36. Aladeemy, M.; Tutun, S.; Khasawneh, M.T. A new hybrid approach for feature selection and support vector machine model selection based on self-adaptive cohort intelligence. *Expert Syst. Appl.* **2017**, *88*, 118–131. [[CrossRef](#)]
37. Abd Elaziz, M.; Ewees, A.A.; Oliva, D. Binary genetic algorithm assisted by a modified artificial bee colony for feature selection. *Neural Comput. Appl.* **2017**, *29*, 261–276.
38. Abdel-Basset, M.; El-Shahat, D.; El-Henawy, I.; De Albuquerque, V.H.C.; Mirjalili, S. A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst. Appl.* **2020**, *139*, 112824. [[CrossRef](#)]
39. Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A survey on evolutionary computation approaches to feature selection. *IEEE Trans. Evol. Comput.* **2015**, *20*, 606–626. [[CrossRef](#)]
40. Alshammari, H.H.; Taloba, A.I.; Shahin, O.R. Identification of olive leaf disease through optimized deep learning approach. *Alex. Eng. J.* **2023**, *72*, 213–224. [[CrossRef](#)]
41. Tomar, V.; Bansal, M.; Singh, P. Metaheuristic Algorithms for Optimization: A Brief Review. *Eng. Proc.* **2023**, *59*, 238. [[CrossRef](#)]
42. Diker, A.; Elen, A.; Közkurt, C.; Kılıçarslan, S.; Dönmez, E.; Arslan, K.; Kuran, E.C. An effective feature extraction method for olive peacock eye leaf disease classification. *Eur. Food Res. Technol.* **2024**, *250*, 287–299. [[CrossRef](#)]
43. Zhang, Q. A novel ResNet101 model based on dense dilated convolution for image classification. *SN Appl. Sci.* **2022**, *4*, 9. [[CrossRef](#)]
44. Nan, Y.; Ju, J.; Hua, Q.; Zhang, H.; Wang, B. A-MobileNet: An approach of facial expression recognition. *Alex. Eng. J.* **2022**, *61*, 4435–4444. [[CrossRef](#)]
45. Gunduz, H. An efficient dimensionality reduction method using filter-based feature selection and variational autoencoders on Parkinson’s disease classification. *Biomed. Signal Process. Control* **2021**, *66*, 102452. [[CrossRef](#)]
46. Holland, J. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975; Volume 7, pp. 390–401.
47. Golberg, D.E. Genetic algorithms in search, optimization, and machine learning. *Addion Wesley* **1989**, *1989*, 36.
48. Cherrington, M.; Thabtah, F.; Lu, J.; Xu, Q. Feature selection: Filter methods performance challenges. In *Proceedings of the 2019 International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 3–4 April 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–4.

49. Gunduz, H.; Yaslan, Y.; Cataltepe, Z. Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowl.-Based Syst.* **2017**, *137*, 138–148. [[CrossRef](#)]
50. Bahassine, S.; Madani, A.; Al-Sarem, M.; Kissi, M. Feature selection using an improved Chi-square for Arabic text classification. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 225–231. [[CrossRef](#)]
51. Gunduz, H.; Cataltepe, Z. Borsa Istanbul (BIST) daily prediction using financial news and balanced feature selection. *Expert Syst. Appl.* **2015**, *42*, 9001–9011. [[CrossRef](#)]
52. Zhou, H.; Wang, X.; Zhu, R. Feature selection based on mutual information with correlation coefficient. *Appl. Intell.* **2022**, *52*, 5457–5474. [[CrossRef](#)]
53. Qiu, P.; Niu, Z. TCIC_FS: Total correlation information coefficient-based feature selection method for high-dimensional data. *Knowl. Based Syst.* **2021**, *231*, 107418. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.