

Article

Integrating Feature Selection and Deep Learning: A Hybrid Approach for Smart Agriculture Applications

Ali Roman¹ , Md Mostafizer Rahman², Sajjad Ali Haider¹, Tallha Akram¹  and Syed Rameez Naqvi^{1,2,*} 

¹ Department of Computer Engineering, COMSATS University Islamabad, Wah Campus, Wah Cantonment 47040, Pakistan; aliroman@ciitwah.edu.pk (A.R.); sajjadali@ciitwah.edu.pk (S.A.H.); tallha@ciitwah.edu.pk (T.A.)

² Department of Computer Science, Tulane University, New Orleans, LA 70118, USA; mrahman9@tulane.edu

* Correspondence: snaqvi@tulane.edu or rameeznaqvi@ciitwah.edu.pk

Abstract: This research tackles the critical challenge of achieving precise and efficient feature selection in machine learning-based classification, particularly for smart agriculture, where existing methods often fail to balance exploration and exploitation in complex, high-dimensional datasets. While current approaches, such as standalone nature-inspired optimization algorithms, leverage biological behaviors for feature selection, they are limited by their inability to synergize diverse strategies, resulting in suboptimal performance and scalability. To address this, we introduce the Hybrid Predator Algorithm for Classification (HPA-C), a novel hybrid feature selection algorithm that uniquely integrates the framework of a nature-inspired feature selection technique with position update equations from other algorithms, harnessing diverse biological behaviors like echolocation, foraging, and collaborative hunting. Coupled with a custom convolutional neural network (CNN), HPA-C achieves superior classification accuracy (98.6–99.8%) on agricultural datasets (Plant Leaf Diseases, Weed Detection, Fruits-360, and Fresh n Rotten) and demonstrates exceptional adaptability across diverse imagery applications.

Keywords: smart agriculture; feature extraction; feature selection; classification; deep neural networks



Academic Editor: Hugo Wai Leung Mak

Received: 6 March 2025

Revised: 7 April 2025

Accepted: 10 April 2025

Published: 12 April 2025

Citation: Roman, A.; Rahman, M.M.; Haider, S.A.; Akram, T.; Naqvi, S.R. Integrating Feature Selection and Deep Learning: A Hybrid Approach for Smart Agriculture Applications. *Algorithms* **2025**, *18*, 222. <https://doi.org/10.3390/a18040222>

Copyright: © 2025 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) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Rapid developments in smart agriculture are redefining how modern farming addresses the challenges of resource optimization and sustainability [1,2]. With the growing global demand for food, precision agriculture has become visible as a key solution, leveraging advanced technologies to improve productivity and efficiency [3,4]. Imagery data analysis is essential in this domain, as it allows one to perform activities such as weed identification, disease detection, and fruit quality evaluation [5,6]. Imagery data are also used to address agricultural challenges of land use and land cover change [7]. These analytical capabilities give farmers detailed information that helps them make informed decisions. This can help in improving crop yields and economic outcomes [8].

Smart agriculture now relies heavily on machine learning (ML), which provides strong capabilities for the processing of vast amounts of complex agricultural data [2]. Deep learning models have shown impressive effectiveness in processing visual data, identifying patterns, and producing classification results that are quite precise [9]. These advances enable diverse applications, from pest and disease detection [10] to resource management [11]. Although ML models have great potential, their performance is highly depen-

dent on the quality of input features, particularly when working with high-dimensional imagery datasets.

For machine learning implementations, feature selection is an essential step that provides the most relevant and important characteristics from data that can be used for model training [12]. It improves the robustness and efficiency of classification models by reducing the overfitting risk and decreasing computational complexity [13]. Feature selection is essential for producing consistent and comprehensible findings in agricultural applications, as datasets frequently show diversity in class distributions and resolution.

By presenting a unique hybrid feature selection algorithm, the Hybrid Predator Algorithm for Classification (HPA-C), and a customized Convolutional Neural Network (CNN), this study addresses the issues of smart agricultural classification. HPA-C combines a basic framework with position update equations modeled after a variety of natural activities, including cooperative hunting, foraging, and echolocation, integrating different optimization strategies. This special hybridization allows for better feature selection by improving the algorithm's exploration and exploitation capabilities. By applying these techniques to smart agricultural datasets (dataset details in Section 3.1), the study demonstrates significant improvements in classification accuracy within the 98.6% to 99.8% range. The outcomes directly benefit farmers by providing actionable insights for sustainable and efficient farming practices.

The primary contributions of this work are described as follows:

1. A hybrid algorithm named HPA-C, combining GWO, ABC, and BAT, is presented to provide important features.
2. A custom CNN architecture is developed for classification tasks in agricultural imagery datasets.
3. The proposed methods are empirically validated and demonstrate superior performance compared to benchmark approaches.

The remainder of this paper is organized as follows: Background and related work are covered in Section 2. Section 3 details the datasets and algorithms used in this study. The proposed framework is described in Section 4. Simulation results and statistical analyses are provided in Section 5, and the paper concludes with insights and future directions in Section 6.

2. Related Work

Smart agriculture has undergone some remarkable developments in recent years, driven by the integration of machine learning (ML) and deep learning (DL). A summary of some of the latest work is presented in Table 1. These technologies are being used to address critical challenges such as disease detection, crop classification, and fruit recognition, with a strong emphasis on improving accuracy and operational efficiency. The authors of [14] conducted a comparative study of seven ML techniques. The main objective was to detect diseases in economically important crops such as apples, grapes, maize, bell peppers, and tomatoes. Their work showed the superiority of Random Forest (RF), with an accuracy of 91.80%. A similar type of the work was carried out in [15] but with soybean. The authors performed soybean leaf disease classification with a combination of SWIN transformers and CNNs, achieving impressive accuracies ranging from 76% to 98% in multiple datasets, including PlantVillage and Embrapa. These studies demonstrate how well transformer-based models and ensemble approaches handle the complexities of agricultural data.

In smart agriculture, fruit and crop classification has also gained the attention of researchers and farmers. The authors of [16] investigated the impact of transfer learning on fruit detection. They used the Fruits-360 dataset and achieved accuracies ranging from 65% to 99%. The ML models used in their research work were VGG16, ResNet50, and

EfficientNet-B0. The authors of [17,18] used CNNs to classify fruit images. Their classification accuracies were 98.17% and 97.15%, respectively. Their research work shows the significance of machine learning methods for fruit classification. The detection of crop leaf disease has also received significant attention, with studies such as [19,20] suggesting the use of a hybrid formed by merging Artificial Bee Colony (ABC) and Gray Wolf Optimization (GWO). The authors enhanced the feature selection process and improved classification accuracies. Their implementations achieved the classification accuracies of 98.53% and 95.69%, respectively. Their works highlight the potential of these hybrid algorithms in improving agricultural disease detection. Similarly, the research work carried out by Ravichandran et al. [21], merged Random Forest Optimization (RFO) with SVM and achieved an accuracy of 97.70% for leaf disease classification. Weed classification was performed by Ajayi and Ashi [22] using a Region-Based CNN (RCNN), achieving a classification accuracy of 99.40%. This study emphasizes the importance of object detection methods in precision agriculture, especially when it comes to differentiating weeds from crops.

Feature selection methods have become a key part of improving the performance of ML models. These methods help to increase model performance by reducing data dimensionality and removing unnecessary features. The research work carried out by the authors of [23] introduced a hybrid approach to intrusion detection by merging the Crow Search Algorithm and the Flower Pollination Algorithm. Their work resulted in the achievement of accuracies of 99.14% and 97.98% with AdaBoost. Another work carried out in [24] developed a hybrid of the Grasshopper Optimization Algorithm (GOA) and the Genetic Algorithm (GA) for the detection of cloud intrusion. The authors achieved classification accuracies of 98%, 99%, and 92% using different datasets. These studies show the role of hybrid optimization methods in the selection of features for security applications. At the same time, nature-inspired feature selection algorithms have gained popularity in diverse fields. An interesting work presented in [25] classified emotions. The authors utilized Gray Wolf Optimization (GWO) for EEG signal-based emotion classification and achieved accuracies of 89.63% and 84.54% with XGBoost. Leukemia classification was carried out in [26]. The authors employed entropy-controlled firefly feature selection and achieved classification accuracy above 96% with a multi-class SVM. These diverse cases highlight the adaptability of nature-inspired algorithms, expanding their use beyond agriculture to include healthcare and other domains.

Hybrid optimization techniques are increasingly being adopted to improve accuracy for classification and disease detection. The authors of [20,27] used hybrid optimization methods such as Coyote–Gray Wolf optimization and GWO for the detection of mango and grape leaf disease. They achieved accuracies of 96.71% and 95.69%, respectively. The integration of feature selection techniques with advanced ML models has proven instrumental in achieving high accuracy across various tasks, from disease detection to fruit classification.

In summary, the literature highlights notable advancements in feature selection algorithms and smart agriculture, which are powered by the application of state-of-the-art ML and DL methods. Researchers are increasingly using hybrid optimization strategies to solve challenging issues with remarkable accuracy and effectiveness. To improve these models' interpretability and ease their application in actual agricultural systems, future research could investigate the use of explainable AI (XAI) methodologies. As the sector develops further, these developments have the potential to completely transform agriculture and associated fields, opening the door to more intelligent and sustainable solutions.

Table 1. Summary of research on classification and detection techniques.

Reference	Year	Research Idea	Datasets	Feature Selection	Implementation	Accuracy
[14]	2025	Disease detection in economically important Crops	PlantVillage (maize, apple, grapes, tomato, and bell pepper)	NA	7 ML techniques (best results with RF)	91.80%
[15]	2025	Classification of soybean leaf disease diagnosis	PlantVillage, AI2018, PlantDoc, and Embrapa	NA	SWIN transformer + CNN	98.00%, 97.00%, 76.00%, and 92.00%
[16]	2025	Impact of transfer learning on fruit detection	Fruits-360	NA	CNN with VGG16, ResNet50, MobileNet-V2, and EfficientNet-B0	65% to 99%
[23]	2025	Intrusion detection	UNSW-NB15 and CIC-IDS2017 datasets	Hybrid of crow search algorithm and flower pollination algorithm	AdaBoost	99.14% and 97.98%
[25]	2025	EEG signal-based emotion classification	DEAP and MAHNOB-HCI	Gray Wolf Optimization	XGBoost	89.63% and 84.54%
[26]	2025	Leukemia classification	ALLID_B1, ALLID_B2, C_NMC 2019, and ASH	Entropy-controlled firefly feature selection	Multi-class SVM	99.64%, 98.96%, 96.67%, and 98.89%
[18]	2024	Fruit image classification	Fruits-360	NA	CNN with EfficientNet-B0, MobileNetV2, and ResNet50V2	97.15%
[17]	2024	A CNN-based implementation of fruit recognition	Fruits-360	NA	CNN	98.17%
[28]	2024	UC MERCED land use dataset classification	UC MERCED LandUse	NA	VGG19, VGG16	95.07% and 91%
[27]	2024	Mango leaf disease classification using a hybrid-tuned neural network model	PlantVillage	Hybrid Coyote–Gray Wolf Optimization	NN model	96.71%
[20]	2024	Grape leaf disease detection	PlantVillage (Leaf Blight, Black measles, Black rot, and healthy)	Gray Wolf Optimization	Ensemble classifier	95.69%
[19]	2024	Plant leaf disease detection	PlantVillage	Hybrid ABC	HYgmABC	98.53%
[21]	2024	Plant leaf detection framework	Plant leaf diseases (apple scab, apple black rot, apple cedar rust, and healthy apple)	RFO	SVM	97.70%
[24]	2024	Cloud-IDS by NI-FS techniques	UNSW-NB15, CIC-DDoS2019, and CIC Bell DNS EXF 2021	Hybrid of Grasshopper Optimization Algorithm (GOA) and Genetic Algorithm (GA)	Random Forest	98%, 99%, and 92%
[29]	2024	COVID-19 detection using medical images	Custom dataset	Hybrid of cuckoo search optimization and learning-based optimization	XGBoost	95.99%
[22]	2023	Weed classification	Weed Detection	NA	RCNN	99.40%
[30]	2022	Plant leaf disease detection through CNN	Plant leaf diseases	NA	CNN	98.41%

3. Methods

3.1. Dataset Description

With a particular focus on smart agriculture, this study integrates nine different RGB image datasets from various fields. These datasets address a range of classification difficulties with differences in class count, image distribution, resolution, and scope. Class annotations are applied to all datasets, and the effects of class balance are not taken into consideration for the current study. The datasets are summarized in Table 2.

Table 2. Summary of datasets used in research.

Dataset	# Classes	Images per Class	Total Images	Resolution	Balanced/ Imbalanced	Scope	Version
Plant Leaf Diseases [31]	39	~1000	54,306	Variable (256 × 256 common)	Imbalanced	Plant disease detection	2019
Weed Detection in Soybean crops [32]	12	~(1191 to 7356)	15,336	Variable	Imbalanced	Weed classification	2017
Fruits-360 [33]	141	~490	94,110	100 × 100	Balanced	Fruit recognition	2024
Fresh n Rotten [34]	16	~740	12,335	1020 × 1020	Balanced	Food freshness	2018
UC MERCED [35]	21	100	2100	256 × 256	Balanced	Scene classification	2010
AID [36]	30	~(220 to 400)	10,000	Variable (600 × 600 common)	Imbalanced	Aerial imagery	2017
NWPU-RESISC45 [37]	45	~700	31,500	Variable (256 × 256 common)	Balanced	Remote sensing	2023
SIRI-WHU [38]	12	200	2400	200 × 200	Balanced	Remote sensing	2016
RSSCN7 [39]	7	400	2800	224 × 224	Balanced	Scene classification	2015

The datasets illustrate the development of image categorization research from 2010 to 2024. The purpose of this work is to gain insights into smart agricultural issues by utilizing various labeled datasets, although some datasets are naturally balanced and others are imbalanced. Incorporating additional resources for broader insight while concentrating on datasets directly related to agriculture, this study highlights the potential of data-driven solutions to improve agricultural production and decision making.

Dataset Diversity and Generalization

The datasets utilized in this study were carefully selected to cover a broad spectrum of challenges in smart agriculture. This selection ensures a diverse and representative evaluation of the proposed HPA-C method.

In terms of dataset characteristics, our selection includes variations in resolution, image quality, class distributions, and environmental conditions. These make the evaluation more robust. The plant disease and weed detection datasets use field conditions with real-world variations in lighting, background, and plant health status. The fruit classification datasets further introduce controlled yet challenging classification tasks due to similarities between different fruit categories. The remote sensing datasets contribute large-scale imagery variations, demonstrating the applicability of the method beyond agricultural tasks.

The results indicate strong generalization across these diverse datasets, but we also acknowledge that additional validation on other agricultural datasets, such as multi-season imagery, hyperspectral data, or multi-modal datasets, could further strengthen the conclusions.

3.2. Hybrid Feature Selection

Feature selection is a process of identifying and preserving the important features from a dataset. It helps in enhancing model efficiency and predictive accuracy [40]. This procedure enhances interpretability, lessens computational complexity, and mitigates overfitting. Models can more successfully generalize to new data by concentrating on important features and removing redundancy, which results in more reliable decision making [41].

Nature-inspired optimization algorithms may traverse large search spaces while striking a balance between local exploitation and global exploration. They have shown considerable efficacy in solving feature selection problems. To find the best feature subsets, these algorithms use heuristic-based search methods that draw inspiration from biological, physical, or social events. Consider Genetic Algorithms (GAs) [42], which simulate evolutionary selection. Particle Swarm Optimization (PSO) [43] is focused on swarm intelligence and has been widely adopted for feature selection tasks. Gray Wolf Optimization (GWO) [44] replicates the cooperative hunting strategy of wolves and has demonstrated strong performance in refining search efficiency. Artificial Bee Colony (ABC) [45] focuses on the forging patterns of honeybees. The Bat Algorithm (BA) [46] uses the echolocation behavior of bats for adaptive search. These methods are very beneficial for feature selection in cases where data have high dimensionality.

To improve search effectiveness, we can combine different nature-inspired algorithms to create a hybrid model. These hybrid approaches improve search effectiveness, solution quality, and convergence speed by utilizing the merits of the algorithms used for hybridization [29]. Premature convergence can be avoided and classification performance can be improved by balancing diversification and intensification of the used approaches. The benefits of hybridization have been highlighted in some recent studies conducted in diverse application domains. Consider the example of spam classification. A hybrid bat algorithm with momentum acceleration and Gray Wolf Optimization (GWO) was used to improve spam classification performance [47]. Similarly, a modified bat algorithm was designed to address real-world optimization issues, leading to improved robustness [48]. Another hybrid framework was introduced for telecommunication applications. It introduces the merger of Ant Colony Optimization (ACO) and Artificial Bee Colony (ABC) for optimization of resource allocation in cognitive radio networks. The setup ensured superior management of communication resources [49].

The impact of the hybrids can be observed in other diverse domains like navigation, improving energy efficiency and increasing productivity. The RFO-GWO hybrid was used for path planning in autonomous robots. The sole purpose was to improve navigation efficiency [50]. For high energy efficiency, a GWO-enhanced PSO hybrid was implemented for photovoltaic system optimization [51]. Another GA-GWO hybrid was designed to optimize scheduling in coal mines. The purpose was to increase the operational productivity of coal mines [52]. These accomplishments highlight the use of hybrid approaches in feature selection in cases where dealing with high-dimensional datasets might be difficult.

These developments motivated us to present a new hybrid approach that combines the Bat Algorithm (BA), Artificial Bee Colony (ABC), and Gray Wolf Optimization (GWO) for feature selection in the smart agricultural application domain. The approach suggested in this work combines the ABC's adaptive foraging mechanism, the BA's dynamic search capability, and the leadership-based exploration and exploitation capabilities of GWO.

3.3. Benchmark Feature Selection Techniques

In order to assess the proposed method's efficacy, the performance of the suggested HPA-C algorithm is compared with some prominent established optimization frameworks. We used ReliefF [53], Particle Swarm Optimization (PSO), the Genetic Algorithm (GA), and Gray Wolf Optimization (GWO) for this study because to their efficacy in feature selection and optimization.

These benchmark approaches were chosen because of their complimentary feature selection strengths. PSO and GWO maintain a balance between local and global search, whereas ReliefF provides a quick baseline for assessing feature relevance. GA introduces adaptive evolutionary mechanisms that enhance search efficiency in complex optimization tasks. By comparing HPA-C against these techniques, its advantages can be effectively demonstrated.

While a formal ablation study is not included, we acknowledge the potential ambiguity in isolating the exact contribution of each component within HPA-C. However, as discussed in this section, the hybridization of GWO, ABC, and BAT is carefully designed to enhance the balance between exploration and exploitation. The prime reason for doing this is to prevent premature convergence. The statistical analysis across multiple datasets also demonstrates the overall effectiveness of HPA-C. The analysis indirectly validates the impact of each component. This logic underpins our design choice and supports the robustness of the proposed approach.

3.4. Working with CNNs

CNNs are good at extracting hierarchical features, representing a class of deep learning models that are widely utilized for image and video analysis and classification. They enable automated feature learning through back propagation. They can process temporal and spatial relationships using learnable filters. A typical CNN is a deep neural network with several layers, each of which has a specific purpose.

1. **Convolutional Layers:** These layers use convolution operations with trainable filters to extract spatial characteristics. As they pass over the input, the filters pick up patterns like edges and textures. The convolution operation is mathematically expressed as follows:

$$\phi_{\alpha,\beta,\kappa} = \sum_{\mu=1}^M \sum_{\nu=1}^N \xi_{\alpha+\mu-1,\beta+\nu-1,\lambda} \cdot \omega_{\mu,\nu,\lambda,\kappa} + \theta_{\kappa} \quad (1)$$

where

- ξ : input feature map
 - ϕ : output feature map
 - ω : convolutional filter
 - θ : bias term
 - α, β : spatial indices
 - λ : index of input channel
 - κ : index of output channel
 - M, N : filter dimensions
2. **Normalization and Activation Layers:** After convolution, batch normalization is applied to stabilize training and accelerate convergence. The transformation is defined as follows:

$$\hat{\chi}^{(i)} = \frac{\chi^{(i)} - \mu^{(i)}}{\sqrt{\sigma^{(i)2} + \epsilon}}, \quad \psi^{(i)} = \gamma^{(i)} \hat{\chi}^{(i)} + \beta^{(i)} \quad (2)$$

where

- $\mu^{(i)}, \sigma^{(i)2}$: batch mean and variance, respectively

- $\gamma^{(i)}, \beta^{(i)}$: trainable scale and shift parameters
- ε : small constant for numerical stability
- χ, ψ : input and output activations

To enhance learning capacities, the Rectified Linear Unit (ReLU) activation function adds non-linearity:

$$\psi = \max(0, \chi) \quad (3)$$

where

- χ : input activation
 - ψ : output after ReLU transformation
3. Pooling layers and Dropout Layers: In order to minimize computational complexity while maintaining essential properties, pooling layers carry out spatial downsampling. This is how the max-pooling operation is defined:

$$\phi_{\alpha, \beta, \kappa} = \max_{\mu, \nu} \{\zeta_{\alpha + \mu, \beta + \nu, \kappa}\} \quad (4)$$

where

- ζ : input feature map
- ϕ : pooled output
- α, β : spatial indices
- κ : channel index
- μ, ν : pooling window indices

Dropout is used to reduce overfitting and improve model generalization by randomly deactivating neurons. The dropout operation is expressed as follows:

$$\psi = \frac{\delta \cdot \chi}{1 - \rho} \quad (5)$$

where:

- χ : input activation
 - δ : binary mask, where $\delta \sim \text{Bernoulli}(1 - \rho)$
 - ρ : dropout probability
 - ψ : output activation
4. Fully Connected and Output Layers: Decision-making methods incorporate high-level feature representations through the fully connected layers. These layers produce classification outputs by transforming the retrieved feature maps using weighted transformations. Regression tasks mostly use a linear function in the final output layer, while multi-class classification mostly uses a softmax activation function.

3.5. Performance Evaluation

In multi-class classification, accuracy, macro precision, macro recall, and the macro F1 score are key evaluation metrics.

The **accuracy** of the model is computed as the ratio of correctly predicted instances to the total number of instances:

$$\text{Accuracy} = \frac{|\{i \mid y_i = \hat{y}_i\}|}{|\{i \mid y_i = \hat{y}_i\}| + |\{i \mid y_i \neq \hat{y}_i\}|} \quad (6)$$

where

- y_i : true label for the i -th instance
- $\hat{y}_i$: predicted label for the i -th instance

- $|\cdot|$: cardinality (number of elements in the set).

Macro precision (P_{macro}) is the average of class-wise precision values.

The macro precision (P_{macro}) is computed as the arithmetic mean of precision values for all classes:

$$P_{macro} = \frac{1}{N} \sum_{i=1}^N \frac{|\{j \mid y_j = i \wedge \hat{y}_j = i\}|}{|\{j \mid \hat{y}_j = i\}|} \quad (7)$$

where

- y_j : true label for the j -th instance
- $\hat{y}_j$: predicted label for the j -th instance
- N : total number of classes
- $|\cdot|$: cardinality (number of elements in the set).

Macro recall (R_{macro}) is computed similarly.

The macro recall (R_{macro}) is computed as the arithmetic mean of recall values for all classes:

$$R_{macro} = \frac{1}{N} \sum_{i=1}^N \frac{|\{j \mid y_j = i \wedge \hat{y}_j = i\}|}{|\{j \mid y_j = i\}|} \quad (8)$$

where

- y_j : true label for the j -th instance
- $\hat{y}_j$: predicted label for the j -th instance
- N : total number of classes
- $|\cdot|$: Cardinality (number of elements in the set).

The **macro F1 score** is then calculated as the harmonic mean of macro precision and macro recall:

$$F1_{macro} = \frac{2 \times P_{macro} \times R_{macro}}{P_{macro} + R_{macro}} \quad (9)$$

Since this research focuses on **diverse multi-class datasets**, the **macro precision**, **recall**, and **F1 score** provide a more balanced evaluation than accuracy or micro metrics.

By using macro metrics, a **fair comparison across all classes** is ensured, making the hybrid feature selection method (HPA-C) more interpretable and applicable to **smart agriculture applications**, where minority-class performance is crucial for reliable decision making.

4. Framework

4.1. HPA-C Framework

The process begins by initializing a random population of wolves according to Equation (10):

$$\text{wolves}_{i,j} \sim \text{Bernoulli}(0.5), \quad \forall i \in \{1, 2, \dots, \text{num_wolves}\}, \forall j \in \{1, 2, \dots, \text{num_features}\} \quad (10)$$

The initial population goes through a fitness evaluation test. Equation (11) presents the fitness test in terms of classification accuracy:

$$\text{Accuracy}(\text{wolves}_i) = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \times 100 \quad (11)$$

Following the fitness evaluation, the solutions are sorted based on $\text{Accuracy}(\text{wolves}_i)$ in descending order, where α_{wolf} is termed the best, β_{wolf} the second most appropriate, and δ_{wolf} is the least appropriate. They are then subjected to an iterative position update and fitness evaluation process. The technique uses the main optimization loop and computes three new position variables: X_1 , X_2 , and X_3 . X_1 uses the main GWO position

calculation mechanism, X_2 uses the ABC mechanism, and X_3 follows the BAT procedure, as expressed by Equation (12).

$$\begin{aligned} X_1 &= \alpha_{\text{wolf},j} - A1 \cdot |C1 \cdot \alpha_{\text{wolf},j} - \text{wolves}_{i,j}| \\ X_2 &= \text{wolves}_{i,j} + \phi_1 \cdot (\text{wolves}_{i,j} - \beta_{\text{wolf},j}) \\ X_3 &= \text{wolves}_{i,j} + [\phi_3 \cdot (\delta_{\text{wolf},j} - \text{wolves}_{i,j}) + \text{velocities}_{i,j}] \end{aligned} \quad (12)$$

where

- X_1 : variable updated using the Gray Wolf Optimization (GWO) update equation
- X_2 : variable updated using the Artificial Bee Colony (ABC) update equation
- X_3 : variable updated using the bat update equation (BAT)
- $\alpha_{\text{wolf},j}$: position of the alpha wolf in dimension j
- $\beta_{\text{wolf},j}$: position of the beta wolf in dimension j
- $\delta_{\text{wolf},j}$: position of the delta wolf in dimension j
- $A1, C1$: random coefficients for GWO
- ϕ_1 : Random coefficient specific to ABC
- ϕ_3 : Random coefficient specific to BAT.

For exploitation, a single position variable (P) is computed by taking the maximum of the three update mechanisms:

$$P = \max(X_1, X_2, X_3) \quad (13)$$

The motivation behind choosing the maximum function is that it retains the most significant feature importance score among three incorporated mechanisms in HPA-C (Algorithms 1–3). This ensures that highly discriminative features emphasized by at least one technique are not suppressed. Unlike averaging, which can dilute strong feature contributions, taking the minimum may underestimate feature significance. The maximum function ensures that the best-ranked features from any algorithm are preserved. This enhances the robustness of the selected feature subset and improves classification accuracy.

The wolves' positions are bounded using Equation (14), and binary quantization (thresholding at 0.5) is applied as expressed by Equation (15):

$$\text{wolves}_{i,j} = \max(\min(P, \text{ub}_j), \text{lb}_j) \quad (14)$$

$$\text{wolves}_{i,j} = \begin{cases} 1 & \text{if } \text{wolves}_{i,j} > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

The choice of 0.5 as a threshold ensures a balanced decision boundary. The feature selection scores are normalized between 0 and 1. Our choice makes retention and discarding equally probable. This approach aligns with widely adopted binary discretization techniques in metaheuristic-based feature selection to ensure methodological consistency. Other threshold choices (0.4 and 0.6) showed no significant performance gains, with 0.5 providing stable results across datasets.

A flow chart of the proposed HPA-C is shown in Figure 1.

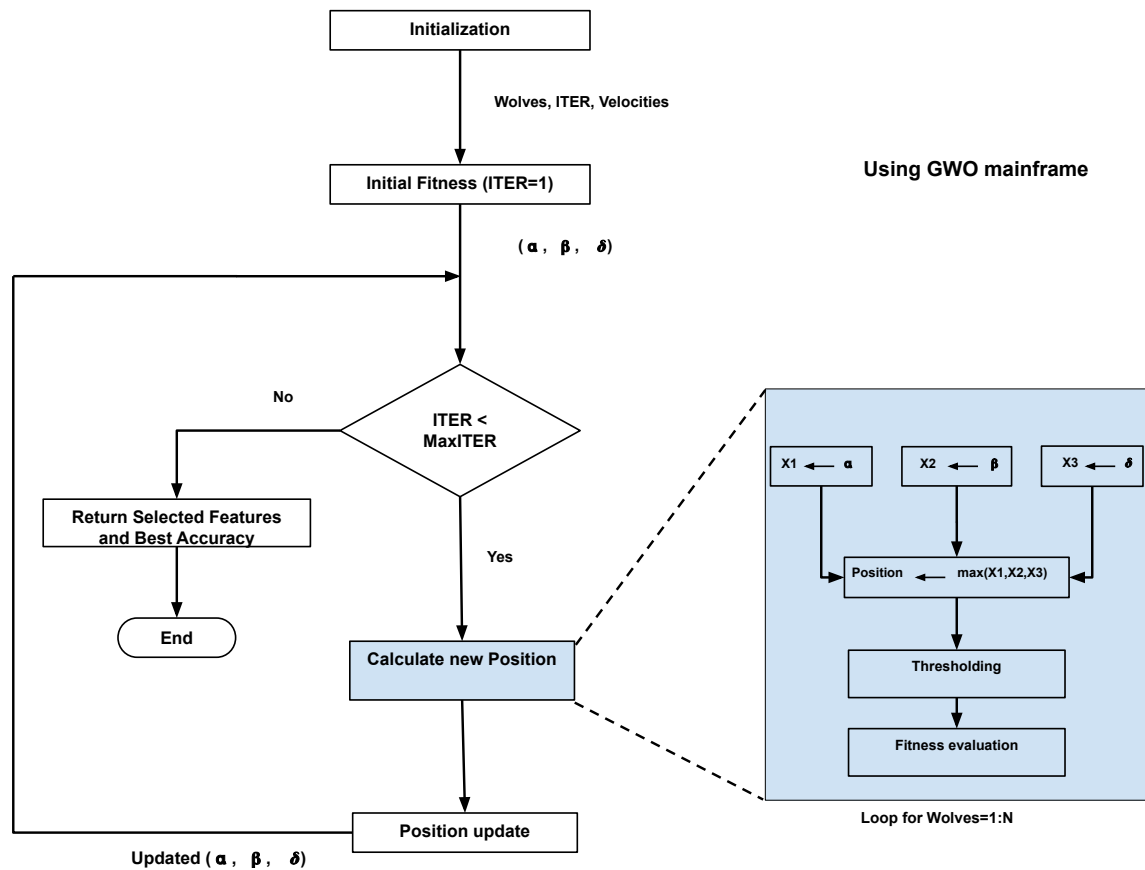


Figure 1. HPA-C flow diagram.

Algorithm 1 The Proposed NI Optimization Algorithm (HPA-C)**Require:** objective_function: Function to be maximized

- 1: num_samples: Number of features
- 2: lb: Search space lower bounds
- 3: ub: search space upper bounds
- 4: num_wolves: Number of wolves
- 5: max_iterations: Maximum number of iterations

Ensure: selected_samples: Best solution found

6: accuracy: Fitness value of the best solution

7: $wolves \leftarrow \text{InitializeWolves}(\text{num_wolves}, \text{num_samples})$ 8: $\text{UpdateWolves}(wolves, \text{objective_function}, lb, ub)$

▷ Initial evaluation

9: $\text{SortWolves}(wolves, \text{objective_function})$ 10: **for** iteration = 1 **to** max_iterations **do**11: $a \leftarrow 2 - \text{iteration} \cdot (2 / \text{max_iterations})$ 12: **for** $i = 1$ **to** num_wolves **do**13: $\text{UpdatePosition}(wolves[i], \alpha_wolf, \beta_wolf, \delta_wolf, a, lb, ub)$ 14: **end for**15: $\text{UpdateWolves}(wolves, \text{objective_function}, lb, ub)$ 16: $\text{SortWolves}(wolves, \text{objective_function})$ 17: **end for**18: **Return** α_wolf as selected_samples and its fitness as accuracy

Algorithm 2 UpdatePosition (Using X1, X2, X3 and Max Selection)

```

1: procedure UPDATEPOSITION(wolf, alpha_wolf, beta_wolf, delta_wolf, a, lb, ub)
2:   for j = 1 to num_samples do
3:     % GWO Component
4:      $X1 \leftarrow \alpha\_wolf[j] - A1 \cdot |C1 \cdot \alpha\_wolf[j] - wolf[j]|$ 
5:     % ABC Component
6:      $X2 \leftarrow wolf[j] + \phi \cdot (beta\_wolf[j] - wolf[j])$ 
7:     % BAT Component
8:      $velocities[i][j] \leftarrow velocities[i][j] + (delta\_wolf[j] - wolf[j]) \cdot rand()$ 
9:      $X3 \leftarrow wolf[j] + velocities[i][j]$ 
10:    % Compute the final position using maximum selection
11:     $P1 \leftarrow \max(X1, X2, X3)$ 
12:    % Apply boundary constraints
13:     $wolf[j] \leftarrow \max(\min(P1, ub[j]), lb[j])$ 
14:    % Binary discretization (Thresholding at 0.5)
15:     $wolf[j] \leftarrow 1$  if  $wolf[j] > 0.5$ , 0 otherwise
16:  end for
17:  % Evaluate fitness of the updated wolf
18:   $fitness \leftarrow objective\_function(wolf)$ 
19:  return fitness
20: end procedure

```

Algorithm 3 UpdateWolves

```

1: procedure UPDATEWOLVES(wolves, objective_function, lb, ub)
2:   for i = 1 to num_wolves do
3:     % Update position of each wolf
4:      $fitness\_values[i] \leftarrow UpdatePosition(wolves[i], \alpha\_wolf, \beta\_wolf,$ 
5:        $\delta\_wolf, a, lb, ub)$ 
6:   end for
7: end procedure

```

4.2. Convergence Analysis of HPA-C

It is generally accepted that performing convergence analysis of an NI process is arduous and often needless. There may be multiple reasons behind this conjecture: (1) NI processes are often stochastic in nature, making their behavior impossible to predict, even under identical initial conditions [54]. (2) They usually involve complex interactions among multiple components, thereby creating nonlinear and unpredictable dynamics [55].

However, the convergence criteria can be defined—in this case, based on the following conditions:

1. Decreasing Step Size:

a must decrease linearly from 2 to 0 over iterations.

2. Diversity Maintenance:

- Random components r_1 , r_2 , and ϕ ensure that the population maintains diversity.
- Velocity update in the BAT component prevents premature convergence.
- Binary thresholding ensures that selected features remain in a valid discrete space.

3. Boundary Handling:

$$X_i = \max(lb, \min(X_i, ub)), \quad (16)$$

that is, the positions must be bounded within the lower and upper bounds $[lb, ub]$.

4. Stability and Convergence:

- The algorithm should converge if the classification accuracy stabilizes and does not show significant changes over iterations.
- Mathematically, the convergence condition is expressed as follows:

$$\lim_{k \rightarrow \infty} f(a^k) = f(\mathbf{X}^*) \quad (17)$$

where k is the iteration number, a^k is the position of the alpha wolf at iteration k , and $\mathbf{X}^*$ represents the global optimum solution maximizing classification accuracy.

4.3. Complexity Analysis of HPA-C

The time complexity analysis of HPA-C is discussed in this section. It provides a detailed analysis of the time complexity of the proposed HPA-C and a comparison with baseline algorithms, including GWO, PSO, GA, and ReliefF.

HPA-C is primarily based on GWO while incorporating enhancements inspired by the ABC and BA. These modifications improve the exploration and exploitation phases but do not introduce additional iterations, keeping the overall complexity comparable to that of GWO.

Initialization Phase:

$$O(P \times D) + O(P \times f(D)),$$

where P is the population size, D is the feature number, and $f(D)$ represents complexity of the fitness evaluation.

Iterative Optimization Phase: The computational steps per iteration are outlined as follows:

- Position updates (based on GWO structure with ABC and BAT modifications): $O(P \times D)$
- Binary discretization (constant-time adjustments): $O(P \times D)$
- Fitness evaluation and sorting: $O(P \times f(D)) + O(P \log P)$.

The total complexity per iteration remains as follows:

$$O(P \times (D + f(D)) + P \log P).$$

Since the core framework is GWO, the additional computations from ABC and BAT do not introduce new loops or higher-order terms, preserving the asymptotic complexity. Thus, in theoretical terms, the complexity of HPA-C remains identical to that of GWO.

For `max_iterations` iterations, the overall complexity is expressed as follows:

$$O(\text{max_iterations} \times P \times (D + f(D) + \log P)).$$

Table 3 showcases a comparative analysis of the time complexities of HPA-C and the baseline algorithms.

Table 3. Comparison of time complexities of HPA-C and baseline algorithms.

Algorithm	Per-Iteration Complexity	Overall Complexity
GWO	$O(P \times (D + f(D)) + P \log P)$	$O(\text{max_iterations} \times P \times (D + f(D) + \log P))$
PSO	$O(P \times D + P \times f(D))$	$O(\text{max_iterations} \times P \times (D + f(D)))$
GA	$O(P \times D + P \times f(D) + P \log P)$	$O(\text{max_iterations} \times P \times (D + f(D) + \log P))$
ReliefF	$O(f(D))$	$O(f(D))$
HPA-C	$O(P \times (D + f(D)) + P \log P)$	$O(\text{max_iterations} \times P \times (D + f(D) + \log P))$

The computational complexity of HPA-C remains comparable to that of GWO. The additional computational overhead from ABC and BAT is minimal, since no extra loops or high-order dependencies are introduced, ensuring computational efficiency.

4.4. Proposed CNN Framework

The proposed 17-layer CNN is specifically designed for feature-based classification. It strikes a balance between efficiency and accuracy; the architecture is shown in Figure 2. Compared with traditional CNNs that process raw images, our model has structured feature representations ($1 \times 1 \times \text{numFeatures}$) as input, reducing computational overhead. The architecture primarily has three convolutional layers, each followed by batch normalization and ReLU activation. Their combinations are represented as LC1, LC2, and LC3. The first and third convolution layers use 3×3 kernels to capture local feature patterns, while the middle 1×1 convolution layer in LC2 acts as a bottleneck to reduce dimensionality and improve computational efficiency. A max-pooling layer (stride = 2) is used to refine feature representations. It is followed by a dropout layer (50%) to enhance generalization. The model transitions into a 128-neuron dense layer with ReLU activation before reaching the final softmax and classification layer.

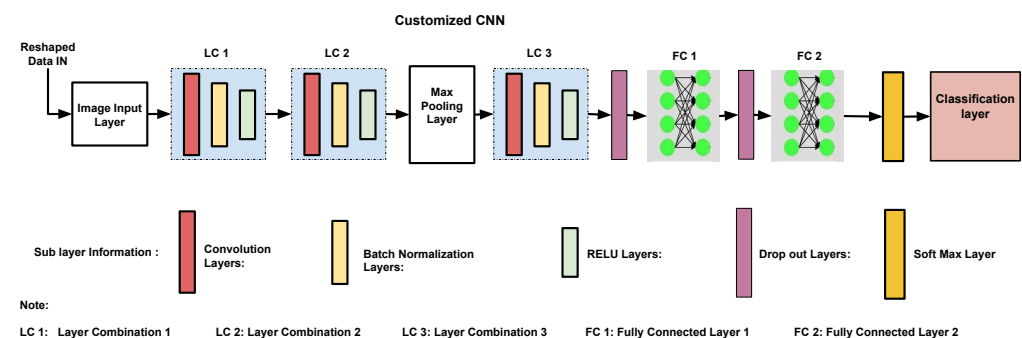


Figure 2. Customized CNN architecture used with HPA-C.

Different alternative architectures and activation functions were considered during the design phase. Deeper networks with additional convolutional layers or larger dense layers (e.g., 256 or 512 neurons) were tested but did not provide significant performance improvements while increasing design complexity. Conventional CNNs designed for raw image processing were unsuitable due to redundant feature learning. Activation functions such as sigmoid and tanh were explored but led to vanishing gradient issues. This slowed down convergence. Leaky ReLU was also tested but did not offer any advantage in classification accuracy.

The combination of 3×3 and 1×1 convolutions enhances feature extraction while keeping the parameter count low. Batch normalization stabilized training, and dropout prevented overfitting. The Adam optimizer with a learning rate of 0.001 was selected for fast convergence. Softmax activation in the final layer enables probabilistic multi-class classification. This design ensures that the custom CNN effectively complements HPA-C by focusing on the most relevant features without excessive computational cost.

4.5. Complete Pipeline

The complete pipeline is shown in Figure 3. The performance of HPA-C depends on key hyperparameters. These are the number of wolves (N), the maximum number of iterations (T), step size (a), and learning rate (η). To ensure optimal performance, we tried different ranges of hyperparameters and conducted sensitivity analysis.

Table 4 shows the tested ranges for the used hyperparameters. The step-size parameter (a) is particularly important because it is linearly decreased from an initial value of 2 to 0

over the iterations. This helps in controlling the exploration–exploitation trade-off. The selected hyperparameters are presented in Table 5.

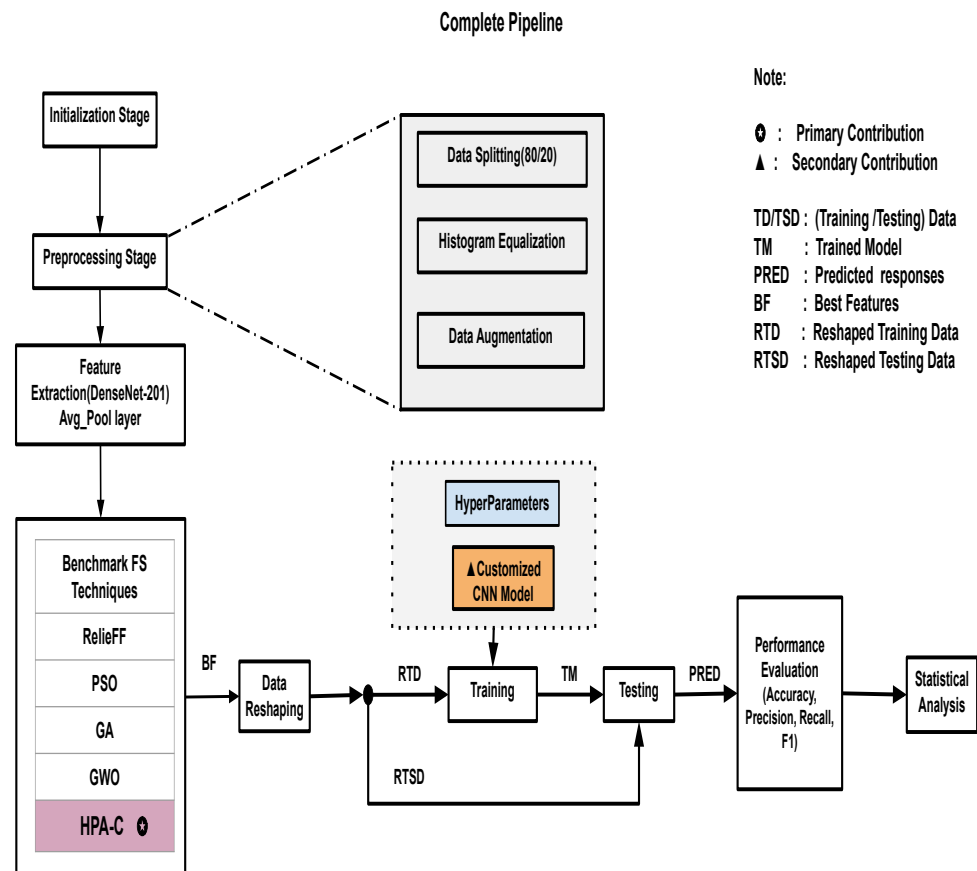


Figure 3. Complete pipeline of the research work.

Table 4. Explored ranges of hyperparameters.

Hyperparameter	Explored Range
Number of wolves (N)	5 to 30
Maximum iterations (T)	10 to 150
Step size (a)	Decreasing from 2 to 0
Learning rate (η)	0.0005 to 0.005
Optimizers	Adam, SGDM, RMSProp
Execution environment	CPU, GPU

Based on sensitivity analysis, we selected hyperparameters that balanced accuracy and computational efficiency. The chosen values are summarized in Table 5.

Table 5. Final selected hyperparameters.

Hyperparameter	Selected Value
Number of wolves (N)	20
Maximum iterations (T)	100
Step size (a)	Decreases from 2 to 0
Learning rate (η)	0.001
Optimizer	Adam
Execution environment	GPU

Hyperparameter selection was based on empirical results. Sensitivity analysis showed that increasing N beyond 20 or T beyond 100 resulted in negligible accuracy gains but significantly increased computational cost. The step size (α) follows a decreasing pattern from 2 to 0, ensuring a smooth transition from exploration to exploitation. The Adam optimizer demonstrated superior convergence properties and improved accuracy. The GPU-based implementation provided faster execution compared to the CPU.

MATLAB was used to access and preprocess the datasets utilized in this study in order to guarantee the best possible performance on classification tasks. Histogram equalization, data augmentation, and data separation are the three main processes in the preprocessing pipeline. The datasets used in this work were divided into training (80%) and testing (20%) subsets, a widely adopted practice to maintain a balance between training robustness and unbiased model evaluation. Histogram equalization was used to improve image contrast, making it easier to identify feature-rich areas. This is especially useful in agricultural datasets, where lighting conditions often vary. To further enhance dataset diversity, reduce the chance of overfitting, and improve the model's generalization capacity, data augmentation techniques including rotation, flipping, and scaling were used. In our proposed pipeline, data augmentation was applied consistently across all datasets, regardless of their class distribution. The used augmentation techniques, such as rotation, flipping, and scaling, were selected to enhance model generalization and robustness rather than to address class imbalance.

While certain datasets may have different properties, our strategy was designed to maintain a uniform approach that preserves the quality of images while introducing variations that are beneficial for training. We ensured that augmentation operations were applied in a controlled manner, preventing excessive transformations that could distort critical features.

An essential first step in utilizing pre-trained architectures based on deep learning for effective representation learning is feature extraction. DenseNet-201 was used in this investigation, with its average pooling layer used to extract features. The densely connected design of DenseNet-201, which encourages feature reuse and addresses the vanishing gradient issue, justifies its adoption as a very effective method for image classification problems. As high-level representations of the input images, the recovered feature vectors capture complex texture-related and spatial properties that help with precise categorization.

Following feature extraction, five feature selection algorithms (ReliefF, PSO, GA, GWO, and the suggested HPA-C) are applied to the extracted features. The most pertinent features that have a major impact on classification performance are chosen by each algorithm. The purpose of HPA-C is to increase selection efficiency by utilizing the advantages of ABC, GWO, and BA. The classification model then receives the selected best features (BFs).

We performed feature selection analysis across agricultural datasets and found three dominant categories: texture features, color-based features, and shape-based features. Texture features like GLCM contrast and entropy capture surface irregularities. These are crucial for plant disease and soil condition assessment. Color features like mean RGB and HSV histograms help evaluate fruit ripeness and plant health by detecting pigmentation changes. Shape features like aspect ratio and compactness are necessary for distinguishing weeds from crops and classifying plant species. The selected features align with key agricultural indicators, emphasizing their biological relevance and consistency across datasets.

To validate feature importance, we analyzed selection frequency across datasets. Table 6 summarizes the most frequently chosen features.

Table 6. Most Frequently Selected Features Across Agricultural Datasets

Feature Type	Selected Features
Texture	Contrast, entropy, and homogeneity
Color	Mean RGB, mean HSV, and standard deviation of L* (Lab)
Shape	Aspect Ratio, compactness, and convexity

Note: L* represents the lightness component in the CIE Lab color space. The asterisk (*) denotes that value is perceptually uniform.

These results confirm that HPA-C consistently selects biologically meaningful features.

After feature selection, a customized convolutional neural network is designed and uses pre-selected high-dimensional features instead of raw images. This reduces computational complexity without sacrificing classification effectiveness. With different convolutional layers, batch normalization, activation functions, and fully connected layers, the CNN architecture is designed to maximize learning and classification accuracy. The efficiency of the model is increased overall because the integration of the chosen features guarantees that only the most discriminative data are used in the classification process.

Several performance indicators, like accuracy, macro precision, macro recall, and macro F1 score, are calculated to assess the model's efficacy.

Additionally, accuracy is used as a critical criterion in statistical analysis to confirm the superiority of the suggested HPA-C feature selection method. The resilience of HPA-C in maximizing classification performance is confirmed by comparative analysis against alternative feature selection strategies, reaffirming its appropriateness for intelligent agricultural applications.

5. Results

The research work was carried out on a PC with an Intel Core i7 processor, 32 GB RAM, and NVIDIA Tesla K40 GPU, and Microsoft Windows 10 Operating System. The framework was implemented using MATLAB 2022A. We calculated performance metrics, specifically accuracy, macro precision, macro recall, and macro F1 score, in order to assess the framework.

The results are presented in Table 7.

These findings demonstrate that HPA-C, combined with the custom CNN, achieves remarkable improvements in feature selection, even in scenarios where existing techniques have already reached extremely high performance levels (beyond 99% accuracy). In smart agricultural datasets such as *Plant Leaf Diseases*, *Weed Detection*, *Fruits-360*, and *Fresh n Rotten*, methods like GWO and GA already perform at near-optimal levels, yet HPA-C still manages to push the boundaries further, achieving up to a 0.5% to 1.5% increase in accuracy and a 1% to 3% boost in macro F1 score. While these numerical gains may seem small, at such high-performance levels, even a fraction of a percent can represent a significant improvement in real-world applications, where misclassification can lead to costly agricultural losses.

The Fresh n Rotten dataset stands out, where HPA-C reaches 99.8% accuracy, surpassing GWO by 0.5% and significantly outperforming RelieFF by 12.8%, demonstrating its superior feature selection capability. Similarly, in the Plant Leaf Diseases dataset, while GWO achieves 98.3% accuracy, HPA-C improves it to 98.6%, further enhancing class-wise performance as evidenced by a 1.5% boost in macro F1 score. These refinements highlight the ability of HPA-C to extract highly discriminative features, even when existing techniques seem to have saturated performance levels. The accuracy plot for the different datasets and feature selection techniques is also shown in Figure 4.

Table 7. Performance metrics of feature selection methods across datasets using DenseNet-201(FE) and a custom CNN.

Dataset	Feature Selection	Accuracy	Macro Precision	Macro Recall	Macro F1 Score
Plant Leaf Diseases	RelieFF	0.826	0.805	0.790	0.790
	PSO	0.978	0.974	0.974	0.974
	GA	0.973	0.968	0.967	0.967
	GWO	0.983	0.979	0.979	0.979
	HPA-C	0.986	0.983	0.983	0.983
Weed Detection	RelieFF	0.990	0.993	0.977	0.984
	PSO	0.994	0.990	0.992	0.991
	GA	0.996	0.994	0.995	0.995
	GWO	0.997	0.996	0.996	0.996
	HPA-C	0.998	0.998	0.997	0.997
Fruits-360	RelieFF	0.987	0.987	0.987	0.987
	PSO	0.991	0.990	0.990	0.990
	GA	0.981	0.980	0.981	0.981
	GWO	0.982	0.981	0.981	0.981
	HPA-C	0.986	0.986	0.986	0.985
Fresh n Rotten	RelieFF	0.885	0.813	0.793	0.794
	PSO	0.991	0.981	0.979	0.980
	GA	0.983	0.967	0.964	0.964
	GWO	0.993	0.984	0.984	0.983
	HPA-C	0.998	0.994	0.994	0.993
UC-MERCEd	RelieFF	0.640	0.642	0.640	0.640
	PSO	0.940	0.943	0.940	0.940
	GA	0.947	0.948	0.947	0.947
	GWO	0.956	0.956	0.954	0.955
	HPA-C	0.962	0.964	0.962	0.962
AID	RelieFF	0.584	0.584	0.577	0.566
	PSO	0.914	0.915	0.912	0.913
	GA	0.889	0.886	0.884	0.885
	GWO	0.908	0.907	0.908	0.906
	HPA-C	0.927	0.926	0.926	0.925
NWPU-RESISC45	RelieFF	0.547	0.547	0.547	0.541
	PSO	0.853	0.855	0.854	0.854
	GA	0.834	0.835	0.834	0.833
	GWO	0.854	0.857	0.854	0.854
	HPA-C	0.882	0.883	0.882	0.882
RSSCN7	RelieFF	0.744	0.760	0.744	0.741
	PSO	0.896	0.898	0.896	0.896
	GA	0.876	0.880	0.876	0.877
	GWO	0.914	0.914	0.914	0.914
	HPA-C	0.926	0.926	0.925	0.925
SIRI-WHU	RelieFF	0.666	0.664	0.666	0.661
	PSO	0.927	0.930	0.927	0.927
	GA	0.920	0.920	0.920	0.920
	GWO	0.927	0.927	0.927	0.927
	HPA-C	0.929	0.929	0.929	0.928

Beyond smart agriculture, HPA-C proves its adaptability to remote sensing datasets, where class diversity and imbalance present additional challenges. Although benchmark techniques like GWO perform well, HPA-C consistently delivers improvements of up to 6.5% in macro F1 score and 5% in accuracy in complex datasets such as NWPU-RESISC45 and AID. Even in scenarios where GA and PSO struggle with minority-class recognition, HPA-C ensures a 30% to 35% improvement in macro recall, making it more robust for diverse class distributions.

The key takeaway is that, while traditional feature selection techniques appear to have reached their peak, HPA-C successfully extracts more meaningful feature subsets, leading to noticeable refinements, even at the highest performance levels. This reinforces its reliability for real-world agricultural and remote sensing applications, where even the smallest increase in classification accuracy can have substantial economic and operational benefits.

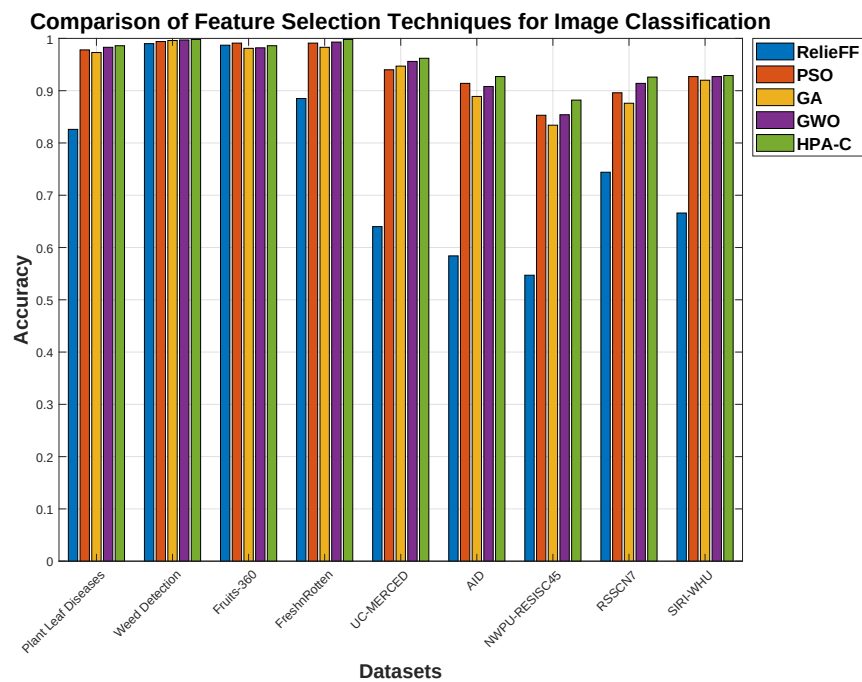


Figure 4. Accuracy plots for 9 datasets and 5 FS techniques.

5.1. Statistical Analysis

The effectiveness of the proposed HPA-C feature selection technique was evaluated by conducting a comparative analysis against four benchmark methods: ReliefF, PSO, GA, and GWO. The classification accuracy was measured across nine diverse datasets related to smart agriculture and remote sensing. To ensure a statistically rigorous comparison, we performed a one-way ANOVA, followed by Tukey's HSD post hoc test and the Wilcoxon signed-rank test.

The box plot obtained from the ANOVA is presented in Figure 5. The results of the one-way ANOVA presented in Table 8 indicate a statistically significant difference among the feature selection techniques ($p = 8.776 \times 10^{-5}$, $F = 7.88$). This suggests that at least one of the methods performs significantly better than the others.

Table 8. ANOVA results for accuracy.

Source	SS	df	MS	F	<i>p</i> -Value
Columns	0.23797	4	0.05949	7.88	8.776×10^{-5}
Error	0.30189	40	0.00755	—	—
Total	0.53986	44	—	—	—

The statistical analysis revealed that PSO, GA, and GWO do not exhibit significant differences in performance. This observation suggests the convergence to similar optima in the given datasets. However, their inclusion is still meaningful because it demonstrates that HPA-C outperforms not just one but multiple representative optimization-based approaches. The lack of significant differences among PSO, GA, and GWO highlights that conventional population-based methods may not always offer substantial distinctions in feature selection performance, emphasizing the need for more advanced hybrid approaches like HPA-C.

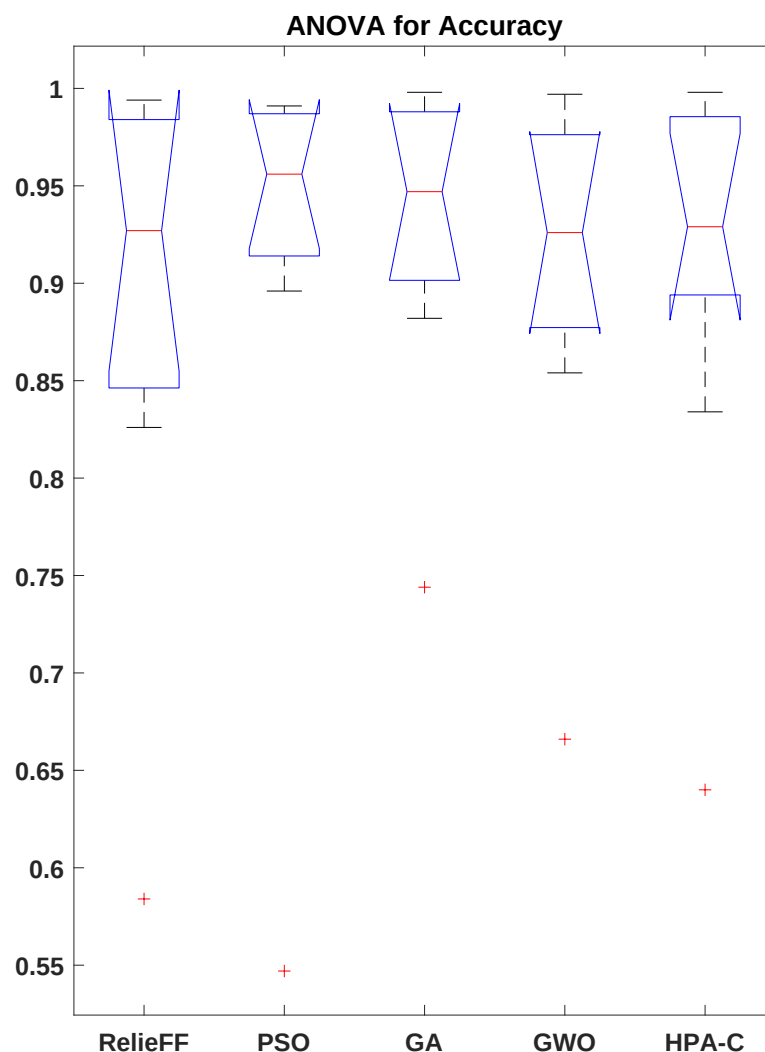


Figure 5. ANOVA box plots.

To provide a comprehensive evaluation, we retained all three in the comparative analysis. The analysis showed that HPA-C is not only superior to a statistical method (RelieFF) but also to three widely used metaheuristic techniques. While redundancy in performance exists, excluding any of these methods would weaken the generalizability of the findings.

In addition to the ANOVA, a Wilcoxon signed-rank test was also performed to validate the superiority of HPA-C. The test results, summarized in Table 9, prove that HPA-C outperforms all other used benchmark feature selection techniques, with $p < 0.05$ in all cases. This value provides strong statistical evidence for the robustness of HPA-C across diverse datasets.

Table 9. Wilcoxon signed-rank test results.

Comparison	<i>p</i> -Value	Significance
HPA-C against RelieFF	0.0078	Significant
HPA-C against PSO	0.0195	Significant
HPA-C against GA	0.0039	Highly Significant
HPA-C against GWO	0.0039	Highly Significant

The results clearly prove that HPA-C significantly improves classification accuracy when compared to ReliefF, PSO, GA, and GWO. HPA-C consistently performs better than the benchmark approaches.

Statistical analysis indicates that HPA-C is a robust and effective feature selection technique for smart agricultural applications and beyond. It demonstrated strong generalization across diverse datasets.

5.2. Feature Selection Consistency Across Datasets

An analysis of feature selection patterns across datasets revealed that while selected features varied based on dataset characteristics, certain features were consistently prioritized, indicating that HPA-C effectively identifies highly discriminative features that enhance classification performance. The texture-based and color histogram features were frequently selected in plant disease and fruit classification tasks, while spectral and shape-based features were dominant in weed detection and remote sensing datasets. This consistency suggests that HPA-C captures domain-relevant attributes rather than selecting features arbitrarily.

5.3. Convergence and Wall-Clock (Elapsed) Time

The convergence plot for HPA-C and GWO using the Plant Leaf Diseases dataset is shown in Figure 6.

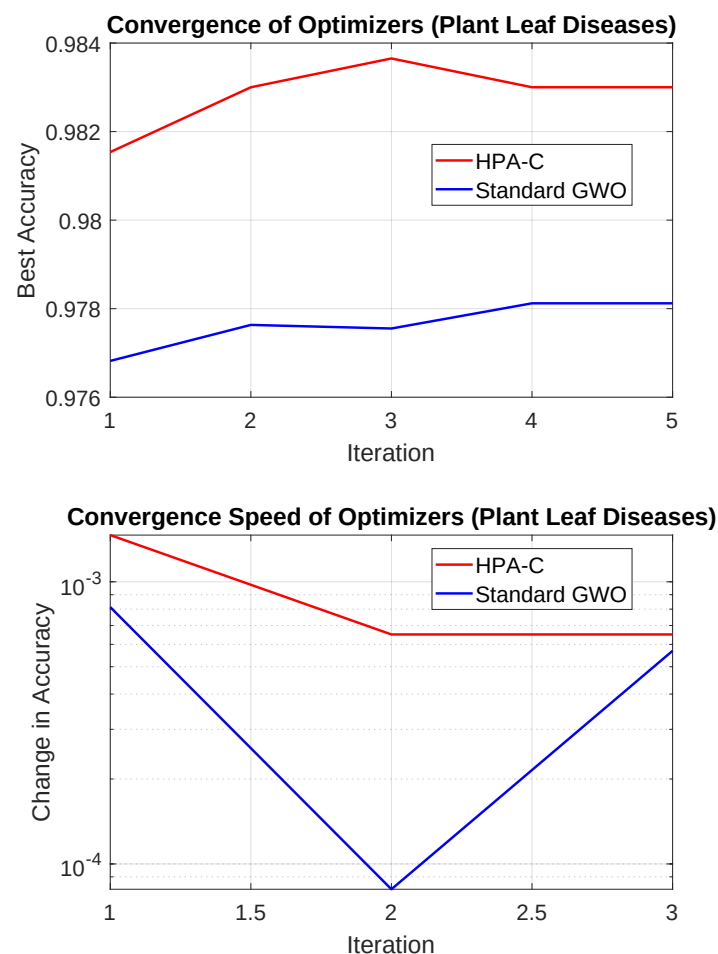


Figure 6. Convergence plot using Plant Leaf Diseases dataset.

In terms of computational time, HPA-C required 5631.75 s, while GWO completed the feature selection process in 5224.16 s. This marginal increase of approximately 10%

in elapsed time for HPA-C is justified by its enhanced search capabilities and improved feature selection. Moreover, the theoretical complexity of the two algorithms remains similar, as discussed in Section 4.3, ensuring that hybridization does not impose significant computational overhead.

5.4. Challenges and Future Work

While this research demonstrates the effectiveness of the proposed hybrid feature selection technique and custom CNN for smart agricultural datasets, several challenges highlight areas for further exploration and improvement.

One of the key challenges encountered during implementation was the limitation of computational resources, particularly GPU memory. State-of-the-art architectures, such as Vision Transformers (ViTs) and large-scale convolutional networks, require substantial memory and processing power, making their deployment infeasible on constrained hardware. This limitation restricted the scope of model exploration and prevented the utilization of potentially more powerful architectures. Future work can address this challenge by leveraging model compression techniques such as quantization, pruning, and knowledge distillation to optimize memory efficiency. Additionally, distributed training using frameworks like TensorFlow and PyTorch 2.0 across multiple GPUs or cloud-based solutions, such as Google Colab, AWS, or TPUs, could enable the training of high-capacity models without hardware bottlenecks.

The datasets used in this study vary in terms of image quality, resolution, and class distribution, reflecting real-world challenges in smart agricultural applications. All datasets contain RGB images, but their resolutions differ. This difference could potentially introduce inconsistencies in feature representation. The preprocessing pipeline, including image resizing and normalization, was designed to mitigate such variations and ensure a consistent input format for the CNN. Our study does not explicitly focus on image quality variations such as noise, blur, or lighting conditions, which could influence model performance. Future work could explore image enhancement techniques to improve robustness against such factors.

Class imbalance is a problem in datasets where certain diseases or conditions are under-represented and could impact classification performance. This issue is especially concerning in minority classes. Although the proposed model demonstrated strong performance across diverse datasets, class-balancing techniques such as oversampling (e.g., SMOTE and ADASYN) or cost-sensitive learning could further enhance generalization. We could use advanced synthetic data generation approaches, including Generative Adversarial Networks (GANs) and diffusion models. These could also help address imbalance-related challenges by generating realistic samples for under-represented categories.

Despite these dataset limitations, the model consistently outperformed benchmark methods. This demonstrates its robustness in handling diverse agricultural imagery. Future research should systematically analyze the impact of resolution variations, noise levels, and class distribution on classification accuracy to provide deeper insights into how dataset quality affects feature selection and CNN performance.

The prospects of using Python-based tools (v3.13.2) can also be explored. Tools like PyTorch's weighted loss functions for cost-sensitive learning and imbalanced learning for oversampling can be explored. To ensure that these balancing procedures are effective in a variety of settings, future research should assess how they affect classification performance.

The primary focus of this research was tilted towards smart agricultural datasets. There is still room to expand the concept to other fields, such as medical imaging. However, direct application of medical imaging datasets is difficult because there is extreme class imbalance. These datasets also require specialized preprocessing. Future studies could investigate the

creation of sophisticated preprocessing pipelines that fulfill image processing needs by utilizing Python libraries like OpenCV and MONAI.

This study compares HPA-C with well-known optimization-based approaches in order to assess its efficacy as a feature selection strategy. The chosen benchmarks guarantee an equitable evaluation of HPA-C's contributions. We illustrate that the observed advantages are due to our hybrid strategy and not due to incidental factors in comparison with separate heuristic techniques.

We see the potential for more insights from comparisons with end-to-end deep learning models and hybrid metaheuristic techniques. The primary driving force behind our effort is to improve feature selection for applications where computing efficiency and interpretability are crucial. Direct comparisons with deep learning models that do not use feature selection are not straightforward because they are based on a different paradigm. Specifically in medical or specialized domains, combining transfer learning with pre-trained models like SWIN transformers may be a promising path toward further improvement.

This work can be expanded in the future by comparing HPA-C against cutting-edge deep learning models and hybrid metaheuristic-based feature selection methods. Transparency is crucial in applications like medical diagnosis, where using visualization techniques like Grad-CAM could offer deeper insights into model decision making. In addition to meeting the changing requirements of intelligent classification systems, these directions would improve HPA-C's wider application.

For future work, we can use model parallelism and distributed training with frameworks like DeepSpeed or PyTorch Distributed to scale training across numerous nodes. We can maximize computational efficiency by using NVIDIA's Apex for low-precision training, such as mixed precision (FP16). This might greatly speed up calculations without sacrificing accuracy. We can Use Numba and CuPy to implement GPU-accelerated computations for feature selection and preprocessing. This can result in improved execution speed and make handling large datasets easier.

6. Conclusions

HPA-C is a novel feature selection algorithm for classification using agricultural datasets. It leverages a variety of biological activities, including cooperative hunting, foraging, and echolocation, to accomplish superior exploration and exploitation capabilities.

When used along with a custom convolutional neural network (CNN), HPA-C delivered exceptional classification performance by achieving accuracy levels of up to 99.8% on key agricultural datasets, including Plant Leaf Diseases, Weed Detection, Fruits-360, and Fresh n Rotten. These results not only surpass those of methods like GWO and GA but also demonstrate the algorithm's ability to extract highly discriminative features, even in scenarios where performance gains are thought to be saturated. These improvements, though seemingly incremental, are transformative in agricultural applications where even fractional gains in accuracy can reduce misclassifications and enhance decision making.

HPA-C outperformed benchmark techniques in both accuracy and macro F1 score, particularly in challenging datasets like Fresh n Rotten, where it achieved a 12.8% improvement over traditional methods like ReliefF. Beyond agriculture, the competitive performance of HPA-C on supplementary datasets shows its potential for broader applicability, including in domains like medical imaging and environmental monitoring.

This study also focused attention on critical challenges to pave the way for future innovation. Computational resource limitations, particularly GPU memory constraints, restricted the exploration of more advanced architectures like Vision Transformers (ViTs). Addressing these challenges through model compression, distributed training, and low-precision computing could unlock even greater performance gains. Additionally, the lack

of explicit handling of dataset imbalance presents an opportunity to integrate advanced balancing techniques, such as SMOTE, ADASYN, or generative approaches like GANs, to further enhance generalization. Extending the framework to high-stakes domains like medical imaging, where explainability and domain adaptation are paramount, represents another exciting avenue for future research.

In conclusion, this work not only establishes HPA-C as a state-of-the-art feature selection algorithm but also sets a new benchmark for hybrid optimization techniques in machine learning. By addressing the identified challenges and exploring the proposed future directions, this framework has the potential to revolutionize classification tasks across a wide range of applications. Its impact extends beyond smart agriculture, offering a scalable, efficient, and highly accurate solution for real-world problems. We believe this research opens the door to a new era of hybrid optimization-driven machine learning, with far-reaching implications for both academia and industry. The findings presented here are a testament to the transformative power of interdisciplinary innovation, and we are confident that this work will inspire further advancements in the field.

Author Contributions: A.R.—investigation, software, and manuscript draft preparation; M.M.R.—formal analysis and manuscript draft review; S.A.H.—methodology and resources; T.A.—data curation and project administration; S.R.N.—conceptualization, supervision, and manuscript draft preparation. All authors have read and agreed to the published version of the manuscript.

Funding: The authors did not receive support from any organization for the submitted work.

Informed Consent Statement: This research work did not involve any human and/or animal participants.

Data Availability Statement: The datasets used in this study are publicly available in Mendeley and Kaggle and are referenced in the manuscript as [31–39]. The code used/developed in this study will be made available upon receiving a reasonable request.

Conflicts of Interest: The authors declare no competing interests related to the submitted work.

Abbreviations

The following abbreviations are used in this manuscript:

HPA-C	Hybrid Predator Algorithm for Classification
CNN	Convolutional Neural Networks

References

1. Vellingiri, A.; Kokila, R.; Nisha, P.; Kumar, M.; Chinnusamy, S.; Boopathi, S. Harnessing GPS, Sensors, and Drones to Minimize Environmental Impact: Precision Agriculture. In *Designing Sustainable Internet of Things Solutions for Smart Industries*; IGI Global: Hershey, PA, USA, 2025; pp. 77–108.
2. Padhiary, M.; Kumar, R. Enhancing Agriculture Through AI Vision and Machine Learning: The Evolution of Smart Farming. In *Advancements in Intelligent Process Automation*; IGI Global: Hershey, PA, USA, 2025; pp. 295–324.
3. Pandeya, S.; Gyawali, B.R.; Upadhaya, S. Factors Influencing Precision Agriculture Technology Adoption Among Small-Scale Farmers in Kentucky and Their Implications for Policy and Practice. *Agriculture* **2025**, *15*, 177. [\[CrossRef\]](#)
4. Rajak, P.; Ganguly, A.; Adhikary, S.; Bhattacharya, S. Internet of Things and smart sensors in agriculture: Scopes and challenges. *J. Agric. Food Res.* **2023**, *14*, 100776. [\[CrossRef\]](#)
5. Ghazal, S.; Munir, A.; Qureshi, W.S. Computer vision in smart agriculture and precision farming: Techniques and applications. *Artif. Intell. Agric.* **2024**, *13*, 64–83. [\[CrossRef\]](#)
6. Chen, J.; Yu, T.; Cherney, J.H.; Zhang, Z. Optimal integration of optical and SAR data for improving alfalfa yield and quality traits prediction: New insights into satellite-based forage crop monitoring. *Remote Sens.* **2024**, *16*, 734. [\[CrossRef\]](#)
7. Ul Din, S.; Mak, H.W.L. Retrieval of land-use/land cover change (LUCC) maps and urban expansion dynamics of Hyderabad, Pakistan via Landsat datasets and support vector machine framework. *Remote Sens.* **2021**, *13*, 3337. [\[CrossRef\]](#)
8. Kumar, D.; Pradhan, A.K.; Jain, R.; Kumar, V.; Murmu, S.; Samal, I.; Chaurasia, H.S. Remote Sensing in Precision Agriculture: Current Status and Applications. In *Artificial Intelligence and Smart Agriculture: Technology and Applications*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 23–41.

9. Zhou, Y.; Wang, Y.; Yan, N.; Feng, L.; Chen, Y.; Wu, T.; Gao, J.; Zhang, X.; Zhu, W. Contrastive-Learning-Based Time-Series Feature Representation for Parcel-Based Crop Mapping Using Incomplete Sentinel-2 Image Sequences. *Remote Sens.* **2023**, *15*, 5009. [CrossRef]
10. Dolatabadian, A.; Neik, T.X.; Danilevicz, M.F.; Upadhyaya, S.R.; Batley, J.; Edwards, D. Image-based crop disease detection using machine learning. *Plant Pathol.* **2025**, *74*, 18–38. [CrossRef]
11. El Sakka, M.; Ivanovici, M.; Chaari, L.; Mothe, J. A Review of CNN Applications in Smart Agriculture Using Multimodal Data. *Sensors* **2025**, *25*, 472. [CrossRef]
12. Rajput, V.; Mulay, P.; Mahajan, C.M. Bio-inspired algorithms for feature engineering: Analysis, applications and future research directions. *Inf. Discov. Deliv.* **2025**, *53*, 56–71. [CrossRef]
13. Sutojo, T.; Ignatius Moses Setiadi, D.R.; Rustad, S.; Akrom, M.; Herowati, W. Hybrid Quantum-Deep Learning Approach: Optimizing Land Cover Classification with GMM Outlier and Fusion Key Feature Selection. *Int. J. Intell. Eng. Syst.* **2025**, *18*, 638.
14. Goyal, S.R.; Kulkarni, V.S.; Choudhary, R.; Jain, R. A comparative analysis of efficacy of machine learning techniques for disease detection in some economically important crops. *Crop. Prot.* **2025**, *190*, 107093. [CrossRef]
15. Sharma, V.; Tripathi, A.K.; Mittal, H.; Nkenyereye, L. SoyaTrans: A novel transformer model for fine-grained visual classification of soybean leaf disease diagnosis. *Expert Syst. Appl.* **2025**, *260*, 125385. [CrossRef]
16. Salem, D.A.; Hassan, N.A.; Hamdy, R.M. Impact of transfer learning compared to convolutional neural networks on fruit detection. *J. Intell. Fuzzy Syst.* **2024**, *46*, 7791–7803. [CrossRef]
17. Huang, Z. A CNN-based implementation of fruit recognition. *Highlights Sci. Eng. Technol.* **2024**, *85*, 1162–1166. [CrossRef]
18. Sinha, B.B.; Dhanalakshmi, R. A multi-fused convolutional neural network model for fruit image classification. *Int. J. Cogn. Comput. Eng.* **2024**, *5*, 416–424. [CrossRef]
19. Pavithra, P.; Aishwarya, P. Plant leaf disease detection using hybrid grasshopper optimization with modified artificial bee colony algorithm. *Multimed. Tools Appl.* **2024**, *83*, 22521–22543. [CrossRef]
20. Kaur, N.; Devendran, V. A novel framework for semi-automated system for grape leaf disease detection. *Multimed. Tools Appl.* **2024**, *83*, 50733–50755. [CrossRef]
21. Ravichandran, K.; Nandhini, S.A.; Sasilatha, T. Investigation of automated plant disease detection framework using machine learning classifier with novel segmentation and feature extraction strategy. In *Hyperautomation in Precision Agriculture*; Elsevier: Amsterdam, The Netherlands, 2025; pp. 313–321.
22. Ajayi, O.G.; Ashi, J. Effect of varying training epochs of a faster region-based convolutional neural network on the accuracy of an automatic weed classification scheme. *Smart Agric. Technol.* **2023**, *3*, 100128. [CrossRef]
23. Zhang, F.; Huang, L.; Shi, K.; Zhai, S.; Lan, Y.; Li, Q. Intrusion detection based on hybrid metaheuristic feature selection. *Comput. J.* **2025**, *68*, 13–22. [CrossRef]
24. Bakro, M.; Kumar, R.R.; Husain, M.; Ashraf, Z.; Ali, A.; Yaqoob, S.I.; Ahmed, M.N.; Parveen, N. Building a cloud-IDS by hybrid bio-inspired feature selection algorithms along with random forest model. *IEEE Access* **2024**, *12*, 8846–8874. [CrossRef]
25. Asemi, H.; Farajzadeh, N. Improving EEG signal-based emotion recognition using a hybrid GWO-XGBoost feature selection method. *Biomed. Signal Process. Control* **2025**, *99*, 106795. [CrossRef]
26. Maqsood, S.; Damaševičius, R.; Maskeliūnas, R.; Forkert, N.D.; Haider, S.; Latif, S. Csec-net: A novel deep features fusion and entropy-controlled firefly feature selection framework for leukemia classification. *Health Inf. Sci. Syst.* **2025**, *13*, 1–26. [CrossRef]
27. Seetha, J.; Ramanathan, R.; Goyal, V.; Tholkapiyan, M.; Karthikeyan, C.; Kumar, R. Mango leaf disease classification using hybrid Coyote-Grey Wolf optimization tuned neural network model. *Multimed. Tools Appl.* **2024**, *83*, 17699–17725. [CrossRef]
28. Maurya, A.; Akashdeep; Kumar, R. Classification of University of California (UC), Merced Land-Use Dataset Remote Sensing Images Using Pre-Trained Deep Learning Models. In *Deep Learning Techniques for Automation and Industrial Applications*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2024; pp. 45–67.
29. Singh, L.K.; Khanna, M.; Monga, H.; Pandey, G. Nature-inspired algorithms-based optimal features selection strategy for COVID-19 detection using medical images. *New Gener. Comput.* **2024**, *42*, 761–824. [CrossRef]
30. Pandian, J.A.; Kanchanadevi, K.; Kumar, V.D.; Jasińska, E.; Goño, R.; Leonowicz, Z.; Jasiński, M. A five convolutional layer deep convolutional neural network for plant leaf disease detection. *Electronics* **2022**, *11*, 1266. [CrossRef]
31. Geetharamani, G.; Pandian, A. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.* **2019**, *76*, 323–338.
32. dos Santos Ferreira, A.; Freitas, D.M.; da Silva, G.G.; Pistori, H.; Folhes, M.T. Weed detection in soybean crops using ConvNets. *Comput. Electron. Agric.* **2017**, *143*, 314–324. [CrossRef]
33. Moldovan, I.A. Fruits-360 Dataset. 2018. Available online: <https://www.kaggle.com/datasets/moltean/fruits> (accessed on 22 January 2025).
34. Sultana, N.; Jahan, M.; Uddin, M.S. An extensive dataset for successful recognition of fresh and rotten fruits. *Data Brief* **2022**, *44*, 108552. [CrossRef]

35. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th Sigspatial International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
36. Chengala, J. AID Scene Classification Datasets. 2017. Available online: <https://www.kaggle.com/datasets/jiayuanchengala/aid-scene-classification-datasets> (accessed on 22 January 2025).
37. Pirzada, A.R. NWPU-RESISC45 Dataset. 2023. Available online: <https://www.kaggle.com/datasets/aqibrehmanpirzada/nwpuresisc45> (accessed on 22 January 2025).
38. Zhao, B.; Zhong, Y.; Xia, G.S.; Zhang, L. Dirichlet-derived multiple topic scene classification model for high spatial resolution remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *54*, 2108–2123. [CrossRef]
39. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325. [CrossRef]
40. Theng, D.; Bhoyar, K.K. Feature selection techniques for machine learning: A survey of more than two decades of research. *Knowl. Inf. Syst.* **2024**, *66*, 1575–1637. [CrossRef]
41. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [CrossRef]
42. Mahmud, T.; Datta, N.; Chakma, R.; Das, U.K.; Aziz, M.T.; Islam, M.; Salimullah, A.H.M.; Hossain, M.S.; Andersson, K. An approach for crop prediction in agriculture: Integrating genetic algorithms and machine learning. *IEEE Access* **2024**, *12*, 173583–173598. [CrossRef]
43. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [CrossRef]
44. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
45. Abu-Mouti, F.S.; El-Hawary, M.E. Overview of Artificial Bee Colony (ABC) algorithm and its applications. In Proceedings of the 2012 IEEE International Systems Conference SysCon 2012, Vancouver, BC, Canada, 19–22 March 2012; pp. 1–6.
46. Yang, X.S.; Hossein Gandomi, A. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput.* **2012**, *29*, 464–483. [CrossRef]
47. Dhal, P.; Azad, C. Hybrid momentum accelerated bat algorithm with GWO based optimization approach for spam classification. *Multimed. Tools Appl.* **2024**, *83*, 26929–26969. [CrossRef]
48. Umar, S.U.; Rashid, T.A.; Ahmed, A.M.; Hassan, B.A.; Baker, M.R. Modified Bat Algorithm: A newly proposed approach for solving complex and real-world problems. *Soft Comput.* **2024**, *28*, 7983–7998. [CrossRef]
49. Lakshmiramana, P.; Al-Farouni, M.; Senthil, P.; Shashidhara, K.; Mutalik, P. Hybrid of Ant Colony and Artificial Bee Colony Optimization for Resource Allocation in Cognitive Radio Networks. In Proceedings of the 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT), Bengaluru, India, 15–16 March 2024; pp. 1–5.
50. Ketafa, F.H.; Al-Darraj, S. Path Planning for Autonomous Mobile Robots Using the RFO-GWO Optimization Algorithm. *Iraqi J. Sci.* **2024**, *65*, 1070–1088. [CrossRef]
51. Águila-León, J.; Vargas-Salgado, C.; Díaz-Bell, D.; Montalvá, C.M. Optimizing Photovoltaic Systems: A Meta-Optimization Approach with GWO-Enhanced PSO Algorithm for Improving MPPT Controllers. *Renew. Energy* **2024**, *230*, 120892. [CrossRef]
52. Hu, B.; Xiong, Z.; Sun, A.; Yuan, Y. Scheduling of Container Transportation Vehicles in Surface Coal Mines Based on the GA-GWO Hybrid Algorithm. *Appl. Sci.* **2024**, *14*, 3986. [CrossRef]
53. Urbanowicz, R.J.; Meeker, M.; La Cava, W.; Olson, R.S.; Moore, J.H. Relief-based feature selection: Introduction and review. *J. Biomed. Inform.* **2018**, *85*, 189–203. [CrossRef]
54. Green, D.; Aleti, A.; Garcia, J. The nature of nature: Why nature-inspired algorithms work. In *Nature-Inspired Computing and Optimization: Theory and Applications*; Springer Nature: Berlin/Heidelberg, Germany, 2017; pp. 1–27.
55. Shandilya, S.K.; Datta, A.; Nagar, A.K. Nature-inspired Algorithms. In *A Nature-Inspired Approach to Cryptology*; Springer Nature: Singapore, 2023; pp. 3–36. [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.