

Optimizing Deep Learning for Cotton Leaf Disease Detection Using Meta-Heuristic Feature Selection Algorithms

Naeem Ullah*, Francisco Martínez-Álvarez[§], Ivanoe De Falco[‡] and Giovanna Sannino[‡]

*Department of Electrical Engineering and Information Technology - University of Naples “Federico II”, Naples, Italy
Email: naeem.ullah@unina.it

[§]Data Science & Big Data Lab, Pablo de Olavide University, Seville, Spain
Email: fmaralv@upo.es

[‡]Institute for High-Performance Computing and Networking (ICAR) - National Research Council (CNR), Naples, Italy
Email: {ivanoe.defalco, giovanna.sannino}@icar.cnr.it

Abstract—Effective and efficient disease detection is crucial, particularly for economically important crops like cotton. In this paper, we move from the initial development of a deep-learning model for cotton leaf disease detection, called Deep-CCNet, to a more comprehensive comparison of different feature selection algorithms, such as RainWater Algorithm, Particle Swarm Optimization, Bee Evolutionary Algorithm, Genetic Algorithm, and Binary Dragonfly Algorithm. Although Deep-CCNet achieved satisfactory classification performance, the goal of this study is to improve the classification performance and efficiency of deep learning models with meta-heuristic feature selection techniques. This study aims to determine which feature selection method achieves the best balance between performance and computational efficiency. We used the Kaggle “cotton leaf disease dataset”, which has 1,711 images from four classes (namely curl virus, bacterial blight, fusarium wilt, and healthy leaf images), to compare these techniques systematically. Our research attempts to find the most effective method that maximizes model performance while minimizing computing resources, in addition to benchmarking the computational and performance parameters of each approach. The results of this study provide a new approach for the choice of feature selection methods in plant pathology, leading to better early disease diagnosis and increased crop resilience via efficient farming practices.

I. INTRODUCTION

COTTON, also referred to as the “silver fiber”, is among the most significant world agricultural products, the cornerstone of the textile industry, and a means of livelihood for millions of farmers, exporters, processors, and producers, particularly in developing nations [1], [2]. Cotton production, however, is facing threats from numerous pests and diseases, which considerably reduce the quantity and quality of the cotton crop, resulting in significant economic losses [3], [4]. Some of these issues are cotton leaf diseases, which are directly harmful to the health of a plant and ultimately its fiber production. Some familiar infections include fusarium wilt, bacterial blight, and cotton leaf curl virus. These diseases reduce crop yield and weaken plants, increasing vulnerability to pests and secondary infections. [3].

Studies show that pests and diseases cause a loss of approximately 42% of world agricultural production each year

[4]. Traditional disease diagnosis methods, which are mainly based on manual inspection and chemical treatment, are labor-intensive, time-consuming, and often prone to errors. For rural areas where agricultural experts are not available, a precise manual diagnosis is even harder. Therefore, there is a critical need for improved, more efficient, accurate, and timely disease diagnosis techniques [5], [6].

To address the above limitations, the farm sector has witnessed a strong shift towards applying computer vision (CV) and artificial intelligence (AI) technology to automatic, early detection of plant disease [7]–[9]. Leaves have been discovered to be most suitable for such analysis due to their ease of access and visibility compared to other plant components, including stems and roots [10]. Modern CV pipelines typically involve processes such as dataset collection, image pre-processing, feature extraction, feature selection, and classification [11], [12].

Deep convolutional neural networks (DCNNs) have been found effective for automatic feature extraction and classification [13]. In particular, the fully connected layers in DCNNs play an important role by integrating spatial features, such as edges, textures, and color intensities, from the preceding layers to predict the correct class labels [14]. Nevertheless, deep models tend to generate an enormous number of redundant or irrelevant features, thereby leading to overfitting and inefficiencies.

To enhance model resilience and performance, optimization-based feature selection algorithms have been gaining momentum [15]. Techniques such as Particle Swarm Optimization (PSO) [16], Binary Dragonfly Algorithm (BDA) [17], and various meta-heuristic techniques are well suited to select the most salient features with the elimination of noisy or uninformative features. This process of feature selection not only enhances classification accuracy but also reduces the computational cost and results in models that are efficient and lightweight.

In the context of these advancements, this study proposes an integrated approach in which a novel Deep-CCNet deep

learning (DL) model is coupled with meta-heuristic feature selection methods to better improve cotton leaf disease classification performance. Through optimization of feature subsets derived using DeepCCNetFS, we aim to develop a robust and computationally capable model with real-time usage potential in precision farming.

A. Motivation

The key aim of this work is to fulfill the essential requirement of proper early diagnosis of cotton leaf diseases with the least dependency on chemical treatment and facilitate farmers in taking wise decisions using real-time tools. Being highly susceptible to diseases, cotton crops need precise and timely diagnosis for sustainable agriculture, enhanced crop management, and improved yield.

But the use of DL models in such tasks is generally hindered by high-dimensional feature spaces that not only incur computational expense but also introduce redundant information, leading to overfitting and reduced generalization capability. Thus, efficient feature selection is very important in avoiding these constraints.

Lastly, this research seeks to facilitate sustainable agriculture by reducing the application of unnecessary pesticides through efficient early disease detection while making such AI tools accessible and feasible for in-field implementation in cotton farming.

B. Feature Selection Strategies and Proposed Framework

Feature selection is a key component of the modern machine learning pipeline, particularly when dealing with high-dimensional learned feature spaces from deep neural networks. It is crucial to select a subset of the most critical features in building lightweight, efficient, but very accurate models [18]. Without effective feature selection, models can overfit, become computationally more expensive, and be less generalizable, characteristics particularly undesirable in real-world applications such as smart agriculture.

In this context, the deep features extracted from the DeepCCNet model need to be passed from the feature selection step. Since the extracted feature vectors have a high dimensionality, an optimal feature optimization method is essential to enhance the model's performance with computational efficiency.

To address this, we comprehensively evaluate and compare five meta-heuristic feature selection algorithms with different approaches to dynamic exploration and exploitation of the vast and complex feature space: Rainwater Algorithm (RWA) [19], Genetic Algorithm (GA) [20], Particle Swarm Optimization (PSO) [21], Binary Dragonfly Algorithm (BDA) [22], and Bee Evolutionary Algorithm (BEA) [23].

All of these methods are implemented within the DeepCCNet pipeline to optimize feature subsets derived from DeepCCNet for the highest classification performance and the lowest model complexity. Comparative evaluation of these feature selection algorithms is performed based on two key criteria:

- 1) **Classification Performance:** Assessment of how well the selected features enable accurate disease diagnosis over the cotton leaf disease dataset.

- 2) **Computational Efficiency:** Measuring the overall execution time, feature selection time, and classification time, which is important for effective deployment in resource-constrained agricultural settings.

Furthermore, we used the grid search to carefully optimize the hyperparameters, guaranteeing the best possible model efficiency and performance. For empirical validation, we employ the publicly available Cotton Leaf Disease dataset with 1710 images corresponding to four classes of diseases.

II. RELATED WORKS

Recently, the accurate and early detection of cotton leaf diseases using computational methods has been the subject of much research because of its potential effects on agricultural production and sustainability. Conventional ML methods have been the mainstay of previous research, but deep learning (DL) and evolutionary algorithms (nature-inspired feature selection) techniques have lately gained traction and provided notable gains in automation and accuracy.

A. Traditional Methods

Image processing methods and traditional ML algorithms were the main tools used in the early stages of plant disease identification efforts. For example, Sarangdhar et al., [24] proposed a method for the classification and control of diseases on cotton leaf along with soil quality monitoring. The study suggested a regression approach based on support vector machines (SVM) for the diagnosis of five diseases that affect cotton leaves: Alternaria, Bacterial Blight, Cereospra, Fusarium wilt, and Gray Mildew. Farmers were given access to the identification of disease along with its cures through an Android app upon illness detection. Along with the water level in a tank, the Android app was also utilized to display the values of soil factors, including temperature, moisture content, and humidity. Shakel et al., [25] divided the images of Cercospora cotton leaves into clusters using the K-means clustering technique. The hybrid approach for extracting texture and color characteristics was used to extract features. Ultimately, Cercospora cotton leaves were classified using SVM. Similarly, a Multi-SVM-based categorization system for cotton leaf diseases was proposed by Jenifa et al. [26]. The suggested algorithm processed the crops automatically at every stage, identifying defects at an early stage and preventing losses to the farmer in terms of products and money. When provided with the necessary data, the aforementioned Multi-SVM algorithm models, which are either human or robot-supervised, enable the segmentation and classification of various disease kinds. Despite being innovative, these methods have several difficulties. The quality and diversity of the input photos had a significant impact on the accuracy of disease identification, and the manual feature extraction procedure was labor-intensive and prone to human error, which made it challenging to scale and adapt to various plant diseases.

B. Deep Learning Advances

To address the shortcomings of manual feature extraction, the shift towards DL, particularly DCNNs, has significantly

enhanced the robustness and accuracy of plant disease detection systems. These models automate the feature extraction process, capturing complex patterns more effectively than traditional methods. Compared to conventional techniques, these models capture complicated patterns more successfully by automating the feature extraction process [27]. For example, Rai et al. [28] concentrated on developing an enhanced DCNN-based model to address the issue of disease detection and classification in cotton plants. Three distinct experimental setups were examined to examine the effects of varying data split ratios, pooling layer selections (max-pooling vs. average-pooling), and epoch length. A collection including 2293 photos of plants and cotton leaves was used to train the models. Four different leaf classes, plant disease combinations, and the corresponding categories were included in the data. The paper in [29] introduces a dynamic feature selection methodology applied to deep neural networks for univariate and multivariate time series forecasting, based on the Coronavirus Optimization Algorithm [30]. The proposed method, evaluated against random forest models, demonstrated superior performance in terms of mean squared error, highlighting the efficiency and effectiveness of dynamic feature selection. The network that Rajasekar et al. [31] proposed outperformed previous state-of-the-art methods by combining the advantages of the Xception component with the ResNet pre-trained on ImageNet. Kotian et al. [32] used the ResNet50 and KNN ML model for the classification of cotton leaf diseases. Numerous studies have lately used DL methods for the categorization of cotton leaf diseases, as Table 1 illustrates. However, for real-world agricultural applications, their high computing complexity and opaque decision-making procedures often make them more difficult to trust and less accessible. In addition to the studies discussed above, several other DL-based approaches are summarized in Table I for completeness and broader context.

C. Limitations

Our research intends to solve the following important shortcomings of current approaches, despite significant advances in agricultural diagnostics, especially in the diagnosis of cotton leaf disease:

1) *Limited Integration of Feature Selection with DL:* DL models are commonly used in plant pathology research to detect diseases, although adaptive and dynamic-feature selection methods are not fully integrated. DL models are extremely strong, but their efficacy is sometimes limited by static feature sets that don't change in response to fresh information or intricate disease patterns. Moreover, evolutionary algorithms are rarely used in research, despite their potential to improve model accuracy and computing efficiency by constantly optimizing and refining feature selection.

2) *Computational Efficiency and Practical Deployment:* The computing needs of training large-scale models are often overlooked by DL techniques, which restricts their application in resource-constrained situations common to many agricultural contexts. This disparity emphasizes the requirement for optimal

models that strike a compromise between operational viability and performance.

D. Solutions

Presently, our research methodology rigorously examines and assesses several feature selection algorithms inside the Deep-CCNet approach, including RWA, PSO, BDA, GA, and BEA. The goal of this comparison study is to evaluate how each strategy affects the model's accuracy, computing efficiency, flexibility, and understandability. By doing this, we close important knowledge gaps regarding feature selection efficiencies in agricultural applications and establish new standards for the creation of systems that are scalable, trustworthy, and understandable for the diagnosis of cotton leaf disease. This effort opens the door for more sophisticated AI-driven agriculture systems to be used on a global scale.

III. METHODOLOGY

We performed two types of experiments. Firstly, we conducted end-to-end training, validation, and testing of our Deep-CCNet model. Secondly, to improve the performance and reduce the computational cost of our approach, we extracted deep features from the last fully connected layer of the Deep-CCNet model and subjected them to various feature selection algorithms to determine their effectiveness in reducing feature dimensionality and improving classifier performance. We used SVM, KNN, Decision Tree, and Naive Bayes classifiers for the classification of the selection features. The basic workflow of the proposed method is given in Fig. 1.

A. Image Resizing

To maintain uniformity in input size throughout the dataset, all input images are resized to 224 by 224 pixels.

B. Dataset Partitioning

The Deep-CCNet model is evaluated with a twofold approach: 10% of the images are used as a hold-out test set for testing, and the rest, 90% are used in a five-fold cross-validation setup, where in each iteration a different fold (data) is used as a validation set. This approach ensures strong testing against overfitting and encourages generalization.

C. Deep-CCNet Architecture details

The architecture of the Deep-CCNet model is based on the ResNet18's architectural principles; however, it also includes notable modifications specifically designed to maximize the classification performance and reduce computational costs (Table II). The proposed Deep-CCNet architecture is a lightweight convolutional neural network that is optimized for efficiency and fast convergence, ideal for small or medium-sized datasets such as cotton leaf disease classification. Deep-CCNet possesses approximately 5.4 million learnable parameters, significantly lower than ResNet18's 11.18 million parameters. Unlike ResNet18, which uses default ReLU activations, Deep-CCNet uses Leaky ReLU activation units throughout the network to correct the dying neuron problem and facilitate gradient flow.

TABLE I
RECENT WORK ON AUTOMATIC COTTON LEAF DISEASE DETECTION USING DL AND ML TECHNIQUES

Reference	Number of classes	Method	Images in the dataset	Advantages	Limitation	Year
Lakshmi et al. [33]	Seven	Pre-trained ResNet50 and region based segmentation	2,384	Satisfactory results	No Explainability (NE) and computationally expensive approach	2024
Bhujade et al. [34]	Ten	Hybrid approach (CNN with Multi-Resolution Feature Optimization)	990	Good generalization ability and efficient	No Unseen Samples Testing (NUST) and NE	2024
Mohammad et al. [35]	Six	Pre-trained models and custom CNN	1,200	Proposed a new dataset with high-quality images, satisfactory results on unseen samples	NE	2024
Irfan et al. [36]	Eight	Deep CNN with modified dragonfly optimization	4,107	Good generalization ability and satisfactory performance on four datasets	NE and manual hyper-parameter selection	2,024
Manurkar et al. [37]	Six	CNNs	3,600	Balanced dataset used and efficient approach	NUST, low-performance results and NE	2024
Rai et al. [28]	Four	Improved Alexnet	2,293	Satisfactory results, testing on unseen samples, less complex and efficient	Low generalization ability	2023
Arathi et al. [38]	Four	Pre-trained DenseNet-121 model	781	Simple approach	Computationally expensive and NE	2023
Zafar et al. [39]	Four	Feature extraction using EfficientNet-b0 and Inception-v3, feature selection using Emperor Penguin Optimizer, and classification using Quadratic Discriminant Analysis	1,786	Satisfactory performance and good generalization ability	NE	2023
Naeem et al. [40]	Four	VGG16	700	Custom dataset and satisfactory results with a small dataset	NE, low generalization ability, and NUST	2023
Govindasamy et al. [41]	Five	Filited convolutional feature pyramid network	26,100	Efficient and reliable approach	Low classification accuracy	2023
Pandey et al. [42]	Four	Hybrid model (CNN plus SVM)	1,710	Accurate and effective approach of classification	NE, computationally expensive, and low generalization ability	2023

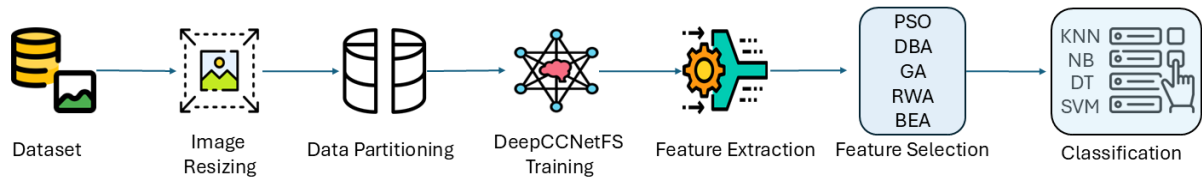


Fig. 1. Overview of the proposed methodology for cotton leaf disease classification.

Furthermore, Deep-CCNet reduces the residual block architecture by eliminating every other convolutional block, creating a shallower but effective model without over-sacrificing feature learning. Deep-CCNet introduces an additional fully connected layer of 1096 neurons preceded by a dropout layer (probability = 0.5) to mitigate overfitting, which is not included in ResNet18.

Moreover, although both networks employ strided convolutions to downsample, Deep-CCNet is specifically architected to predict for 4 classes (i.e., three types of cotton leaf diseases and a healthy class), as opposed to ResNet18's native 1000-

class ImageNet configuration. This change in structure yields a computationally more lightweight model that is simpler to train and better suited to domain-specific classification tasks.

D. Feature Extraction

Following preprocessing, images are sent to the DeepCCNetFS, where feature extraction is handled by many layers. Images are subjected to several filter applications as they go through the convolutional layers, such as conv1 of the first residual unit, which produces feature maps that capture different

TABLE II
KEY DIFFERENCES BETWEEN RESNET18 AND DEEPCCNFS.

S. No	Property	DeepCCNetFS	ResNet18
1	Number of Learnable Layers	12	18
2	Number of Fully Connected (FC) Layers	2	1
3	Number of Convolutional Layers	12	17
4	Number of Dropout Layers	1	0
5	Activation Function (AF)	Leaky ReLU (LReLU)	ReLU
6	Focus	Optimized for specific patterns in cotton leaf diseases	General-purpose image recognition
7	Computational Efficiency	Enhanced by reducing convolutional blocks and simplifying architecture	Standard efficiency with more layers
8	Layer Configuration	Customized layers and fewer residual blocks for specific disease feature extraction	Standardized residual blocks for diverse feature extraction
9	Total learnable parameters	5.4 Million	11.18 Million
10	Residual Blocks	4	8

aspects of the images, such as edges and textures, and can be represented as:

$$f_{ij}^k = \sum ((c^k \times x) + b_k) \quad (1)$$

Here, x represents the local region of the image being convolved, k is used to represent the k th layer, f denotes the feature's value, (i, j) represents pixel coordinates, c^k represents the convolution kernel of current layer, and b_k denotes the bias. Batch normalization layers are then used to normalize these feature maps, which reduces internal covariate shifts and allows for faster and more stable training.

Max pooling layers, such as pool1, assist in lowering the computational burden and model complexity by controlling overfitting and summarizing the convolutional layer outputs into more abstract representations. This reduces the spatial dimension. The network can learn identification functions from residual connections, like res2a, to make sure that no data is lost between layers. Because these connections give gradients direct paths during backpropagation, they are essential for enabling the training of very deep networks. The function $f(x)$ yields the ideal feature map.

$$f(x) = \max(x_1, x_2, x_3, \dots, x_n) \quad (2)$$

where $x_1, x_2, x_3, \dots, x_n$ are the activation values within a local region (e.g., 3×3 window) of the feature map. This operation retains the most prominent features while discarding less significant ones, making the representation more robust and compact. Higher-level reasoning in the network takes place in the FC layers, where fc7 and fc8 are essential building blocks for condensing the distilled information into a format appropriate for categorization. The most important characteristics in Deep-CCNet are effectively highlighted by a global average pooling layer (pool5) that first reduces the feature maps to a single value per map.

The output from the last fully connected layer (fc8) serves as the extracted feature set for downstream tasks. Finally, the softmax layer computes class probabilities, enabling the model to classify cotton leaf images into specific disease categories.

E. Feature Selection Algorithms

Our study's feature selection procedure uses sophisticated algorithms that combine local and global search techniques to extract the most pertinent characteristics from high-dimensional data. By removing redundant and unnecessary features, these techniques decrease the dimensionality of the data and improve the classification models' performance.

1) *Binary Dragonfly Algorithm (BDA)*: The BDA is a nature-inspired optimization method based on the foraging behavior of dragonflies. This algorithm uses a swarm of candidate solutions as binary vectors. It iterates in steps where each solution is updated according to local best (exploitation) and global best (exploration) strategies. The algorithm uses a combination of velocity updates and binary transformations (with sigmoid-like updates) to search the feature space for an optimal feature subset. The fitness function evaluates the feature subsets using a classification model, with the best features selected based on minimum classification error.

2) *Bee Evolutionary Algorithm (BEA)*: The BEA is inspired by the foraging process of bees, i.e., employed, onlooker, and scout stages. The algorithm starts by initializing a population of binary solutions (representations of feature subsets) and evaluating their fitness based on a classification model (KNN). The employed bees attempt to mutate solutions to reach better fitness, and onlooker bees select solutions probabilistically based on fitness. Scout bees search randomly in the solution space when they cannot find better solutions. These steps are repeated by the algorithm iteratively, tracking convergence and improving the population for better feature selection.

3) *Genetic Algorithm (GA)*: The GA is a metaheuristic algorithm inspired by natural selection that is used to solve search and optimization problems. In this algorithm, every member of the population is a binary vector where every bit corresponds to a feature being included or not. The algorithm determines the fitness of every member by training a classifier (for example, KNN) using the selected features. In each generation, the population is refreshed using selection, crossover, and mutation. Selection serves as a promise of

carrying forward superior individuals to the next generation. Crossover exchanges traits among individuals, and mutation introduces randomness using bit flipping. The iterations are iterated over a defined number of generations, and the optimal individual (set of features) is employed in model construction.

4) *Particle Swarm Optimization (PSO)*: The PSO algorithm is an optimization method that is based on the social behavior of bird flocking. The PSO uses a population of particles, and each particle is a possible feature subset. Each particle updates its position according to its own best position and the global best position discovered by the swarm. The algorithm balances exploration (moving towards new areas of the solution space) and exploitation (refining current good solutions) through velocity updates. The particles converge to an optimal subset of features after iterating around the solution space. Fitness is evaluated in terms of classification accuracy on a selected set of features.

5) *RainWater Algorithm (RWA)*: RWA is inspired by rain-drop movement, where raindrops “move down” to lower levels, representing a natural minimization search for the location of the minimum in the feature space. It simulates raindrop behavior through iterative solution movement by fitness in trying to achieve the best set of features, reducing classification error. The solutions (sets of features as subsets) are refined through multiple iterations. The algorithm initializes with a random population of feature subsets, computes their fitness from their sum of feature values, and shifts them to a new position by adjusting velocities toward the optimal solution. A dynamic threshold guarantees that the correct features are selected, even when the initial threshold does not yield any selection.

IV. RESULTS AND DISCUSSION

A. Dataset

In this work, a publicly available Kaggle data set named the “Cotton leaf disease data set” is used [43]. The dataset consists of 1,711 images representing four classes: Curl Virus (418 images), Bacterial Blight (448 images), Fusarium Wilt (419 images), and Healthy (426 images). The images were captured under real-world conditions in the fields of the Southern Punjab region of Pakistan, a major cotton-producing area. Some of the images were also downloaded from the internet. This dataset aims to assist in the identification and classification of cotton leaf diseases, which are critical for the agriculture industry, particularly for cotton cultivation and export to Europe and the US. The representative samples of the dataset are given in Figure 2.

B. Performance of Deep-CCNet without Feature Selection

The Deep-CCNet model is first evaluated without feature selection as a baseline performance. The five-fold cross-validation results are shown in Table III, and the test results on unseen data are shown in Table IV.

Deep-CCNet achieved a good mean accuracy of 98.06% under cross-validation, with precision, recall, and F1 score all well aligned around 98.0% (Table III). The model is very stable

in its performance, indicated by the low standard deviations (below 0.7 for all metrics) and low variance values. The best accuracy achieved is 99.03%, indicating the model generalized well when trained.

On unseen test samples (Table IV), the model performed consistently with a mean accuracy of 96.13%, precision of 96.09%, recall of 96.26%, and F1-score of 96.15%. The results show minimal performance loss between the validation and testing phases, indicating the generalizability of the model. Furthermore, the values of standard deviation on the test set remained low, pointing towards the stability of the model across different data splits.

Overall, the Deep-CCNet model alone without feature selection already exhibits a very good classification capacity for the detection of cotton leaf disease and offers a high baseline for achievable improvement through feature selection techniques.

TABLE III
FIVE-FOLD CROSS-VALIDATION RESULTS FOR THE COTTON LEAF DISEASE DATASET (VALIDATION).

Metric	Accuracy	Precision	Recall	F1-score
Average	98.06	97.97	98.02	97.99
Median	97.73	97.70	97.80	97.75
Maximum	99.03	98.72	98.72	98.71
Minimum	97.09	97.07	97.08	97.08
Standard deviation	0.63	0.69	0.22	0.62
Variance	0.50	0.40	0.48	0.39

TABLE IV
TESTING ON UNSEEN SAMPLES FOR THE COTTON LEAF DISEASE DATASET.

Metric	Accuracy	Precision	Recall	F1-score
Average	96.13	96.09	96.26	96.15
Median	96.43	96.43	96.43	96.43
Maximum	96.51	96.47	96.59	96.53
Minimum	95.35	95.26	95.55	95.37
Standard deviation	0.48	0.37	0.37	0.44
Variance	0.20	0.23	0.14	0.20

C. Performance Evaluation of Meta-Heuristic Feature Selection Methods

The objective of this experiment is to evaluate systematically the performance of various meta-heuristic feature selection algorithms, including RWA, PSO, DBA, GA, and BEA, in improving the performance of classification models of cotton leaf disease. The classification using four different classifiers—SVM, Naïve Bayes, KNN, and Decision Tree is shown in Table V.

Among the tested methods, the BEA algorithm combined with Naïve Bayes yielded the best overall performance, achieving the highest accuracy (97.66%), precision (97.62%), recall (97.65%), and F1-score (97.63%). The next best results are obtained using BEA with KNN (accuracy: 97.08%, F1-score: 97.07%) and SVM (accuracy: 95.32%, F1-score: 95.30%)



Fig. 2. Sample images from the Cotton Leaf Disease Dataset.

classifiers, respectively. The DBA method also demonstrated strong performance, particularly with KNN, which achieved a precision of 97.22% and an F1-score of 97.56%. Similarly, the GA algorithm combined with Naïve Bayes produced competitive results, with an accuracy of 97.14%, precision of 97.22%, recall of 97.22%, and F1-score of 97.22%. These results suggest that while BEA leads in overall classification performance, both DBA and GA also offer robust alternatives when paired with appropriate classifiers.

On the other hand, RWA generated the poorest classification performances of all the classifiers, with accuracies between 58.00% and 72.00%, which means that its feature selection quality is inferior for this task. PSO and GA showed comparable performance, particularly when applied with Naïve Bayes and Decision Tree classifiers, which means that they offer good trade-offs between feature selection quality and model generalization.

In general, BEA and DBA proved to be the best feature selection methods for improving model performance, with RWA being less ideal in this situation.

Furthermore, as shown in Table V, the number of selected features varies across methods, with RWA selecting only 1 feature, PSO selecting 9, and other algorithms like DBA, GA, and BEA consistently selecting 4 features, demonstrating the impact of feature subset size on classification performance.

D. Computational Cost Analysis of Meta-Heuristic Feature Selection Methods

The objective of this experiment is to compare the computational efficiency of all algorithms used for feature selection in terms of feature selection time, classifier training and testing time, and total running time, as illustrated in Table VI.

RWA recorded the fastest total execution time (7.55 seconds) because of its relatively simple optimization process, but poor classification accuracy limits its practical applicability. PSO and DBA recorded moderate execution times (16.65 and 14.71

seconds, respectively) but excellent classification accuracy, making them good options where both efficiency and accuracy matter.

GA, although ranking well, took significantly more overall execution time (60.15 seconds) than PSO and DBA due mainly to the iterative nature of genetic operations. BEA, although showing the best classification performance, took the highest computational cost (451.39 seconds), which highlights a trade-off between predictive performance and computational resource usage. This substantial computational overhead is largely attributed to the ensemble-based nature and nested search operations within BEA, which, while improving feature selection quality, lead to increased time complexity.

From a practical standpoint, the high execution time of BEA raises concerns regarding scalability and real-time deployment feasibility, especially for edge devices or resource-constrained environments. For instance, in large-scale time series applications or streaming scenarios, such computational demand may hinder timely decision-making or exhaust device resources. In contrast, PSO and DBA strike a more favorable balance by offering competitive accuracy at a fraction of the computational cost, making them more viable for real-world deployment.

Therefore, while BEA may be suitable for offline analysis or batch processing tasks where accuracy is prioritized over speed, PSO and DBA are better suited for operational settings requiring rapid inference and low-latency processing. This analysis underscores the need to carefully match algorithm selection with deployment requirements and hardware constraints.

Briefly, although BEA offers higher classification accuracy, its deployment may be impractical in latency-sensitive environments. PSO and DBA are more balanced choices for scenarios where computational expense is a key consideration.

V. CONCLUSION

In this work, we have introduced an end-to-end deep learning architecture, Deep-CCNet, for detecting cotton leaf disease and

TABLE V
PERFORMANCE COMPARISON OF FEATURE SELECTION METHODS AND CLASSIFICATION ALGORITHMS

Feature Selection Algorithm	Classifier	# Selected Features	Accuracy	Precision	Recall	F1-Score
RWA	SVM	1	72.55	72.66	73.30	72.57
RWA	Naïve Bayes	1	72.55	72.80		72.58
RWA	KNN	1	58.82	58.99	57.73	58.03
RWA	Decision Tree	1	58.82	59.15	58.33	58.69
PSO	SVM	9	91.43	91.32	91.87	91.63
PSO	Naïve Bayes	9	85.71	86.11	88.19	86.65
PSO	KNN	9	82.35	84.99	85.78	84.47
PSO	Decision Tree	9	88.24	88.19	88.54	88.06
DBA	SVM	4	94.12	94.95	94.95	94.95
DBA	Naïve Bayes	4	91.81	92.67	92.22	92.34
DBA	KNN	4	97.06	97.22	97.91	97.56
DBA	Decision Tree	4	94.12	94.95	94.95	94.95
GA	SVM	4	94.12	93.75	95.00	94.21
GA	Naïve Bayes	4	97.14	97.22	97.22	97.22
GA	KNN	4	91.18	90.50	91.87	91.27
GA	Decision Tree	4	91.18	90.62	93.18	91.00
BEA	SVM	4	95.32	95.28	95.33	95.30
BEA	Naïve Bayes	4	97.66	97.62	97.65	97.63
BEA	KNN	4	97.08	97.06	97.16	97.07
BEA	Decision Tree	4	94.15	94.13	94.21	94.16

TABLE VI
COMPUTATIONAL COST ANALYSIS OF FEATURE SELECTION METHODS.

Feature Selection Algorithm	Best Accuracy	Feature Selection Time (s)	Classifier Training and Testing Time (s)	Total Execution Time (s)
RWA	72.55	0.24	6.24	7.55
PSO	91.43	15.73	0.44	16.65
DBA	97.06	12.99	1.01	14.71
GA	97.14	58.92	1.02	60.15
BEA	97.66	448.65	2.61	451.39

have extensively analyzed the impact of various meta-heuristic feature selection techniques, including RWA, PSO, DBA, GA, and BEA, on classification performance and computational expense. The Deep-CCNet model has achieved high baseline accuracy (96.13% on unseen samples without feature selection), which is proof of its effectiveness for this task.

Through comprehensive experimentation, we found that the BEA achieved the highest classification performance among different classifiers, followed by DBA and PSO, providing a better balance between classification accuracy and computational cost. However, RWA, though having low computational cost, reported significantly lower predictive performance and was therefore less suitable for this application.

Aside from these promising results, there are several limitations to this work. First, the study is conducted on a single cotton leaf disease dataset; hence, the generalizability of the Deep-CCNet model and feature selection techniques to other crops and general plant pathology tasks has not been established. Second, while the study achieved high classification performance, it did not incorporate explainability techniques to interpret model predictions or feature selection outcomes, limiting the transparency and trustworthiness of the system for

end users such as farmers and agronomists. Finally, the current approach uses traditional classifiers after feature selection; more advanced classifiers or ensemble techniques could further improve results.

Future work can extend this research in many directions. One direction would be to apply the Deep-CCNet model and feature selection steps to larger, multi-crop datasets for their scalability and robustness. Exploring light-weight models optimized for deployment on mobile and edge devices would also enhance the real-world usability of the system in actual agricultural settings. Last but not least, developing explainable AI (XAI) modules for model prediction and feature importances could provide actionable insights to farmers and agronomists and bring the study closer to the overall goals of interpretable and trustworthy AI in agriculture.

CODE AVAILABILITY

The source code used in this study is publicly available¹. This facilitates reproducibility, enables validation of the results, and encourages further research and development in the field.

¹<https://github.com/Engr-Naeem-Ullah/Optimizing-Deep-Learning-with-Meta-Heuristic-Feature-Selection-Algorithms>

REFERENCES

- [1] S. Radhakrishnan, "Sustainable cotton production," in *Sustainable fibres and textiles*, pp. 21–67, Elsevier, 2017.
- [2] S. P. Patil and R. S. Zambre, "Classification of cotton leaf spot disease using support vector machine," *International Journal of Engineering Research*, vol. 3, no. 4, pp. 1511–1514, 2014.
- [3] A. Farooq, J. Farooq, A. Mahmood, A. Shakeel, K. A. Rehman, A. Batool, M. Riaz, M. T. H. Shahid, and S. Mehboob, "An overview of cotton leaf curl virus disease (clcvd) a serious threat to cotton productivity," *Australian Journal of Crop Science*, vol. 5, no. 13, pp. 1823–1831, 2011.
- [4] A. Jenifa, R. Ramalakshmi, and V. Ramachandran, "Cotton leaf disease classification using deep convolution neural network for sustainable cotton production," in *2019 IEEE international conference on clean energy and energy efficient electronics circuit for sustainable development (INCCES)*, pp. 1–3, IEEE, 2019.
- [5] H. Qayyum, S. T. H. Rizvi, M. Naeem, U. b. Khalid, M. Abbas, and A. Coronato, "Enhancing diagnostic accuracy for skin cancer and covid-19 detection: A comparative study using a stacked ensemble method," *Technologies*, vol. 12, no. 9, p. 142, 2024.
- [6] A. Ismail, M. Naeem, U. B. Khalid, and M. Abbas, "Improving adherence to medication in an intelligent environment using reinforcement learning," *Journal of Reliable Intelligent Environments*, vol. 11, no. 1, p. 3, 2025.
- [7] N. Ullah, J. A. Khan, S. Almakdi, M. S. Alshehri, M. A. Qathrad, E. A. Aldakheel, and D. S. Khafaga, "A lightweight deep learning-based model for tomato leaf disease classification," *Computers, Materials & Continua*, 2023.
- [8] P. Timalisina, S. Bhattarai, S. Paudel, and S. Jha, "Enhancing plant disease detection through image analysis using ssdmobilenetv2 and resnet50," in *Proceedings of the Ninth International Conference on Research in Intelligent Computing in Engineering* (V. K. Solanki, T. D. Tan, P. Kumar, and M. Cardona, eds.), vol. 42 of *Annals of Computer Science and Information Systems*, p. 79–87, PTI, 2024.
- [9] M. H. Calp and V. K. Solanki, "Classification of plant species with iris dataset using ann, knn and k-means algorithms," in *Proceedings of the Eighth International Conference on Research in Intelligent Computing in Engineering* (P. Kumar, M. Cardona, V. K. Solanki, T. D. Tan, and A. Wahid, eds.), vol. 38 of *Annals of Computer Science and Information Systems*, p. 7–10, PTI, 2023.
- [10] N. Ullah, J. A. Khan, S. Almakdi, M. S. Alshehri, M. Al Qathrad, N. El-Rashidy, S. El-Sappagh, and F. Ali, "An effective approach for plant leaf diseases classification based on a novel deepplantnet deep learning model," *Frontiers in Plant Science*, vol. 14, p. 1212747, 2023.
- [11] A. Ismail, M. Naeem, M. H. Syed, M. Abbas, and A. Coronato, "Advancing patient care with an intelligent and personalized medication engagement system," *Information*, vol. 15, no. 10, p. 609, 2024.
- [12] N. Ullah, F. Guzmán-Aroca, F. Martínez-Álvarez, I. De Falco, and G. Sannino, "A novel explainable ai framework for medical image classification integrating statistical, visual, and rule-based methods," *Medical Image Analysis*, p. 103665, 2025.
- [13] N. Ullah, I. De Falco, and G. Sannino, "A novel deep learning approach for colon and lung cancer classification using histopathological images," in *2023 IEEE 19th International Conference on e-Science (e-Science)*, pp. 1–10, IEEE, 2023.
- [14] X. Zhang, X. Zhang, and W. Wang, "Convolutional neural network," in *Intelligent Information Processing with Matlab*, pp. 39–71, Springer, 2023.
- [15] D. Gutiérrez-Avilés, M. J. Jiménez-Navarro, J. F. Torres, and F. Martínez-Álvarez, "MetaGen: A framework for metaheuristic development and hyperparameter optimization in machine and deep learning," *Neurocomputing*, vol. 637, p. 130046, 2025.
- [16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.
- [17] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, "Binary dragonfly algorithm for feature selection," in *2017 International conference on new trends in computing sciences (ICTCS)*, pp. 12–17, IEEE, 2017.
- [18] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends*, vol. 1, no. 1, pp. 56–70, 2020.
- [19] T. Biyanto, "Rain water algorithm." <https://www.mathworks.com/matlabcentral/fileexchange/65617-rain-water-algorithm>, 2025. MATLAB Central File Exchange, retrieved May 14, 2025.
- [20] M. Farzaneh, "Feature selection in classification using genetic algorithm." <https://www.mathworks.com/matlabcentral/fileexchange/74105-feature-selection-in-classification-using-genetic-algorithm>, 2025. MATLAB Central File Exchange, retrieved May 14, 2025.
- [21] J. Too, A. R. Abdullah, and N. Mohd Saad, "A new co-evolution binary particle swarm optimization with multiple inertia weight strategy for feature selection," *Informatics*, vol. 6, no. 2, p. 21, 2019.
- [22] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural computing and applications*, vol. 27, pp. 1053–1073, 2016.
- [23] S. M. H. Mousavi, "Bees cnn evolutionary algorithm." <https://github.com/SeyedMuhammadHosseinMousavi/Bees-CNN-Algorithm>, 2025. GitHub repository, retrieved May 14, 2025.
- [24] A. A. Sarangdhar and V. Pawar, "Machine learning regression technique for cotton leaf disease detection and controlling using iot," in *2017 international conference of electronics, communication and aerospace technology (ICECA)*, vol. 2, pp. 449–454, IEEE, 2017.
- [25] W. Shakeel, M. Ahmad, and N. Mahmood, "Early detection of cercospora cotton leaf disease by using machine learning technique," in *2020 30th International Conference on Computer Theory and Applications (ICCTA)*, pp. 44–48, IEEE, 2020.
- [26] A. Jenifa, R. Ramalakshmi, and V. Ramachandran, "Classification of cotton leaf disease using multi-support vector machine," in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, pp. 1–4, IEEE, 2019.
- [27] N. Ullah, J. A. Khan, L. A. Alharbi, A. Raza, W. Khan, and I. Ahmad, "An efficient approach for crops pests recognition and classification based on novel deeppestnet deep learning model," *IEEE Access*, vol. 10, pp. 73019–73032, 2022.
- [28] C. K. Rai and R. Pahuja, "Classification of diseased cotton leaves and plants using improved deep convolutional neural network," *Multimedia Tools and Applications*, vol. 82, no. 16, pp. 25307–25325, 2023.
- [29] M. J. Jiménez-Navarro, C. Restrepo-Estrada, L. Melgar-García, and D. Gutiérrez-Avilés, "Feature Selection Guided by CVOA Metaheuristic for Deep Neural Networks: Application to Multivariate Time Series Forecasting," *Lecture Notes in Networks and Systems*, vol. 749, p. 209–218, 2023.
- [30] F. Martínez-Álvarez, G. Asencio-Cortés, J. F. Torres, D. Gutiérrez-Avilés, L. Melgar-García, R. Pérez-Chacón, C. Rubio-Escudero, A. Troncoso, and J. C. Riquelme, "Coronavirus Optimization Algorithm: A bioinspired metaheuristic based on the COVID-19 propagation model," *Big Data*, vol. 8, no. 4, pp. 308–322, 2020.
- [31] V. Rajasekar, K. Venu, S. R. Jena, R. J. Varthini, and S. Ishwarya, "Detection of cotton plant diseases using deep transfer learning," *J. Mobile Multimedia*, vol. 18, no. 2, pp. 307–324, 2022.
- [32] S. Kotian, P. Ettam, S. Khariche, K. Saravanan, and K. Ashokkumar, "Cotton leaf disease detection using machine learning," *Proceedings of the Advancement in Electronics & Communication Engineering*, 2022.
- [33] M. D. Lakshmi, B. K. Sahoo, and R. K. Ganiya, "Analysis of cotton leaf curl diseases using advanced learning model," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 2s, pp. 188–194, 2024.
- [34] V. G. Bhujade and V. Sambhe, "Multi-disease classification and severity estimation of cotton and soybean leaves using convolutional neural network," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 11s, pp. 584–594, 2024.
- [35] S. Mohmmad, G. Varshitha, E. Yadlaxmi, B. S. Nikitha, D. Anurag, C. Shivakumar, and D. M. S. Isan, "Detection and classification of various diseases in cotton crops using advanced neural network approaches," in *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pp. 1–6, IEEE, 2024.
- [36] I. Haider, M. A. Khan, M. Nazir, A. Hamza, O. Alqahtani, M. T.-H. Alouane, and A. Masood, "Crops leaf disease recognition from digital and rs imaging using fusion of multi self-attention rbnet deep architectures and modified dragonfly optimization," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024.
- [37] V. Manurkar, S. Kulkarni, S. Rokade, and R. Mirajkar, "Cotton plant disease prediction and remedy recommendation system," in *2024 ASU*

- International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSYS)*, pp. 1616–1620, IEEE, 2024.
- [38] B. Arathi and U. N. Dulhare, “Classification of cotton leaf diseases using transfer learning-densenet-121,” in *Proceedings of third international conference on advances in computer engineering and communication systems: ICACECS 2022*, pp. 393–405, Springer, 2023.
- [39] M. Zafar, J. Amin, M. Sharif, M. A. Anjum, S. Kadry, and J. Kim, “Cnn based features extraction and selection using epo optimizer for cotton leaf diseases classification,” *Computers, Materials and Continua*, vol. 76, no. 3, pp. 2779–2793, 2023.
- [40] A. B. Naeem, B. Senapati, A. S. Chauhan, S. Kumar, J. C. O. Gavilan, and W. M. Abdel-Rehim, “Deep learning models for cotton leaf disease detection with vgg-16,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 2, pp. 550–556, 2023.
- [41] S. Govindasamy and D. Jayaraj, “Collaborative ant colony optimization-assisted support vector machine for accurate cotton leaf disease classification and yield prediction,” *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 15, 2023.
- [42] B. N. Pandey, R. P. Singh, M. S. Pandey, and S. Jain, “Cotton leaf disease classification using deep learning based novel approach,” in *2023 International Conference on Disruptive Technologies (ICDT)*, pp. 559–561, IEEE, 2023.
- [43] S. K. Noon, M. Amjad, M. Ali Qureshi, and A. Mannan, “Computationally light deep learning framework to recognize cotton leaf diseases,” *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 6, pp. 12383–12398, 2021.