Journal of Information Systems Engineering and Management
2025, 10(22s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Hybrid Model for Plant Disease Diagnosis: Transfer Learning with Dynamic Feature Selection

Rasha Dawood Salman[1*], Yossra Hussain Ali[2]
[1]*Informatics Institute for postgraduate studies, Information Technology & Communication University, Baghdad, Iraq.*
[2]*Department of Computer Sciences, University of Technology, Baghdad, Iraq.*
[*]*Corresponding Author: ms202320750@iips.edu.iq*

ARTICLE INFO

Received: 21 Dec 2024

Revised: 04 Feb 2025

Accepted: 18 Feb 2025

ABSTRACT

Plant disease classification is critical in sustainable agriculture, essential for ensuring food security and reducing environmental harm. Traditional disease detection methods are often inefficient, time-intensive, and costly, highlighting the need for advanced computational solutions. This work presents a hybrid intelligent model that integrates a framework of transfer learning with bio-inspired optimization in the identification of early and accurate plant diseases. The pre-trained VGG-19 is re-trained parietally for the feature extraction process by applying the transfer learning concept. The Grey Wolf Optimization (GWO) is utilized to select relevant features from the extracted features. This work assesses and evaluates the performance of the proposed model using the Plant Village dataset (39 classes) of 14 plants (Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato).From the experiment result, it is evident that the proposed model has an accuracy of 99.44%, which is higher compared to the traditional machine learning algorithm and state-of-the-art approaches. This enhancement has further shown the potential of the model in evolving how plant diseases are managed due to timely and precise interventions.

Keywords: Plant Disease Classification; Transfer Learning; Grey Wolf Optimization (GWO); Feature Selection; Deep Neural Networks (DNN)

INTRODUCTION:

Currently, the earth is inhabited by people; therefore, it is necessary to maintain its stability and resource availability, including for the growth of resource-rich environments that will support humankind and preserve sustainable agriculture [1, 2]. This involves agroecological and economic sustainability because the practice of agriculture highly impacts ecosystems, leading to erosion of biodiversity and transforming natural landscapes. The traditional approaches to plant disease identification relied on visual examination or identification, laboratory diagnosis, and consultation from an expert. This method might not be efficient, effective, or fast when used, as it might also be costly to undertake [3]. Furthermore, several attributes are to be found notable, such as external scale, a complicated interdependent problem-solving nature, and diverse stakeholder needs. That makes it important for performing high-level computational analyses for diagnosing plant diseases and emerging with the right interventions [4]. Thus, it becomes imperative to rely on intelligent models to either eliminate or minimize the constraints in the conventional approaches of plant disease identification [5].

Computer vision (CV) is an improved, intelligent method of analyzing, enhancing, and classifying images. It could be applied in plant disease classification and achieve much more than a decrease of the obstacles of conventional approaches [6]. CV technique relies on image scanning of colors present in the plant leaves to diagnose diseases. The CV consists of machine learning and deep learning algorithms to tackle the process of detection and classification. The integration of two or more techniques in CV could lead to the emergence of the following undesirable consequences: computational difficulty, extensive data needs, risk of overfitting, and delays in response times. As a result, transfer learning algorithms and bio-inspired optimization have been used as possibly prominent techniques in plant health research to improve agricultural practices [7].

Transfer learning is a machine learning technique whereby information learned from a pre-trained task on a source task is used to improve target task performance when a large, labeled dataset may not be available [8]. This then makes use of new deep learning models such as CNNs that have been pre-trained on large databases to learn, for example, texture, shape, and color. Transfer learning also cuts down the amount of training data required while achieving enhanced detection rates without much computation overhead. In this research, these models are further fine-tuned for the task of plant disease classification [9]. The approach improves generalization by allowing the better representation of features and accurate classification with a small, high-quality dataset, thereby making it a suitable solution for tasks that are characterized by a small data set and limited resources [10].

This work contributes to proposing a new approach to plant disease classification using transfer learning algorithms and bio-inspired optimization, as follows:

1- Its goal is to capture plant diseases at early stages to enable interventions, thereby minimizing crop loss.

2- It starts with preprocessing a set of data composed of various images of plant leaves infected with diseases.

3- The proposed model employed transfer VGG-19 for extracting features. After that, a swarm-based optimization algorithm (Grey Wolf optimization (GWO)) is incorporated into selecting the appropriate features that have been extracted earlier in the proposed model. For the classification process, the Deep Neural Network (DNN) is employed.

The remainder of this paper is structured as follows: Section 2 presents related work, highlighting recent advancements in plant disease classification. Section 3 details the proposed methodology, while Section 4 discusses experimental setup and evaluation. Finally, Section 5 concludes the study with findings and future directions.

RELATED WORK

There are several models intended to create plant disease taxonomy. Some of these models are:

In crop leaves, in 2022 [11] Paymode and Malode proposed a challenging issue of plant disease detection, particularly on tomato and grape leaves. They proposed a CNN on VGG architecture recommended for multi-crop disease identification for leaves. Furthermore, the authors correctly classified and predicted diseased and healthy tissues using a dataset of diseased and healthy leaf images. The proposed method provided a recognition accuracy of 98.40% for grape leaves and 95.71% for tomato leaves.

An approach using transfer learning in deep learning to diagnose diseases in the paddy crop leaves which are easily affected by disease-causing agents including fungi and bacteria had been developed by Gautam et al. in 2022 [12]. VGG16, ResNet, InceptionV3, SqueezeNet as well as VGG19 were used by the authors in classifying the images of a leaf. The suggested approach was prescreening through semantic segmentation for area extraction with adjustment of transfer learning models for classification. This made the model impressive with a general accuracy of 96.4%.

Elaraby et al. in 2022 [13] directed their works on improving deep learning model to distinguish disease in many crops to enhance the performance. To classify diseases the authors used a deep convolutional neural network called AlexNet made more suitable by Particle Swarm Optimization (PSO) algorithm and they used five crops containing 25 classes of disease. With their method, they recorded a great sensitivity estimate of 98.83%, specificity of 98.56% and sensitivity of 98.78%.

Tabbakh and Barpanda in 2023 [14] proposed a novel, transfer learning and Vision Transformers (ViT) based model called TLMViT for plant disease classification. The proposed method refines feature extraction by initially using pre-trained CNN models like VGG19 or ResNet50 and then using a ViT at a deeper level. We evaluated our proposed TLMViT model using the PlantVillage and Wheat dataset to obtain validation accuracy scores equivalent to 98.81% and 99.86%, respectively.

Nayak et al. in 2023 [15] focused on rice disease and nutrient deficiency detection; this study demonstrates the effectiveness of CNN-based transfer learning models for smartphone image processing. The MobileNetV2 model achieved superior results with a validation accuracy of 97.56% and was integrated into an Android application for real-time diagnostics. This research provides a low-cost, accessible solution for farmers, enabling on-field disease detection without reliance on internet connectivity.

Ahad et al.in 2023 [16] compared six CNN architectures, including DenseNet121 and ResNet152V, and introduced a DEX ensemble model for rice disease classification. Using a dataset of nine epidemic rice diseases, the ensemble model achieved accuracy of 98%.

Sharma et al. in 2024 [17] addressed a very current need for plant disease detection systems for accurate classification of diseases given the shortage of it due to the minimal availability of large datasets.[1] They concentrated their study on optimizing and enhancing the state-of-the-art Deep Learning models: MobileNetV2, ResNet18, AlexNet, and VGG16 to recognize multiple types of plant diseases. The best result of 94.4% was achieved using MobileNetV2 after fine-tuning these architectures on a dataset containing 39 classes of healthy and diseased leaves from 14 species of plants.

Shafik et al. in 2024.[18] focused on the challenge of accurately detecting plant diseases and overfitting and fine feature extraction in training deep learning models. The authors proposed two innovative plant disease detection models: PDDNet-AE and PDDNet-LVE, which integrate nine pre-trained CNN architectures, including DenseNet201 and ResNet50, combined with ensemble learning techniques. Using the Plant Village dataset containing 54,305 images of 38 categories, the proposed models achieved accuracies of 96.74% and 97.79%, respectively.

Naseer et al. in 2024 [19] provided an insight into the type of growth stages of pomegranates while providing a CRnet transfer learning-based approach that extracts spatial features, best matched to a random forest classifier to enhance the precision. On examining a collection of 5,857 images distributed over five growth stages, the proposed model was found to attain a classification accuracy of 98%. This study contributes significantly to improving pomegranate production and market quality in terms of yield and quality by mobilizing resources efficiently and avoiding pests.

Ibarra-Pérez et al. in 2024 [20] worked on classifying the phenological stages of beans by evaluating four CNN architectures: AlexNet, VGG19, SqueezeNet, and GoogleNet. From a dataset produced by RGB cameras, GoogleNet offered the best results, achieving an accuracy of 96.71% and specificity of 98.73%.

Rezaei et al. in 2024 [21] focused on the problem associated with the use of limited data for plant disease identification and have thus proposed the use of few-shot learning (FSL) with Feature Attention (FA) Model. Their PMF+FA method resulted in 90.12% accuracy in the PlantDoc benchmark in few as five images per class even with such complex field settings. This work shows that FSL techniques can be applied to low-data environments thus making it easy to scale up for real-time digital farming systems.

MATERIAL AND METHODS

This section introduces the essential material and method used in the proposed model.

Dataset

The dataset for this study was obtained from the PlantVillage database, containing 54,305 images of 14 plants (Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato) species across 39 classes, with each class labeling leaves as healthy or diseased. There is also a category with 1,143 background images without leaves. The dataset underwent several augmentation processes to increase training data diversity and reduce overfittings, such as image flipping, gamma correction, noise injection, color augmentation via PCA, rotation, and scaling. This resulted in an expanded dataset of 61,486 images. Figure 1 shows a sample of the dataset.

Figure 1: sample of the dataset

Grey Wolf Optimization (GWO)

The Gray Wolf Optimization Algorithm (GWO) is inspired by grey wolves and their social ranking behavior in the pack. GWO has shown promising outcomes in several optimization domains compared to other intelligent algorithms[22]. GWO involves four operational mechanisms. The first mechanism is the encircling alpha operation, where wolves surround the alpha wolf for prey. Other wolves surround the prey. The second mechanism involves the encircling beta operation, where wolves run to the beta wolf while closing in on the prey. The third mechanism is the encircling delta operation, where the wolves locate the delta wolf while closing in on it. The fourth operational mechanism relates to the encircling omega operation, where wolves approach the prey from different directions[23]. In addition to the four mechanisms, a mathematical representation depicts the search strategies of the wolves. In standard GWO, wolves of different ranked members search for prey, with the alpha wolf having the top-ranked position, the beta wolf having the second top-ranked position, and the delta wolf ranked third among the wolves. The rest of the wolves scatter at random to search for the prey. The position of the alpha wolf, the beta wolf, and the delta wolf in the search space represents the movement of the wolves to minimize the objective function. The number of iterations or the termination condition for GWO is also defined.

Visual Geometry Group -19 (VGG-19)

Convolutional Neural Networks (CNNs) have achieved remarkable performance in various visual perception tasks, such as image recognition, object detection, and semantic segmentation. Among the CNN architectures, the VGG16/19 is the state-of-the-art method involving deeper two-dimensional filters that exploit spatial hierarchies. The VGG uses small convolution filters of size 3 x 3 with huge depth and 2 x 2 max pooling. Figure 2 shows the main architecture of VGG-19.

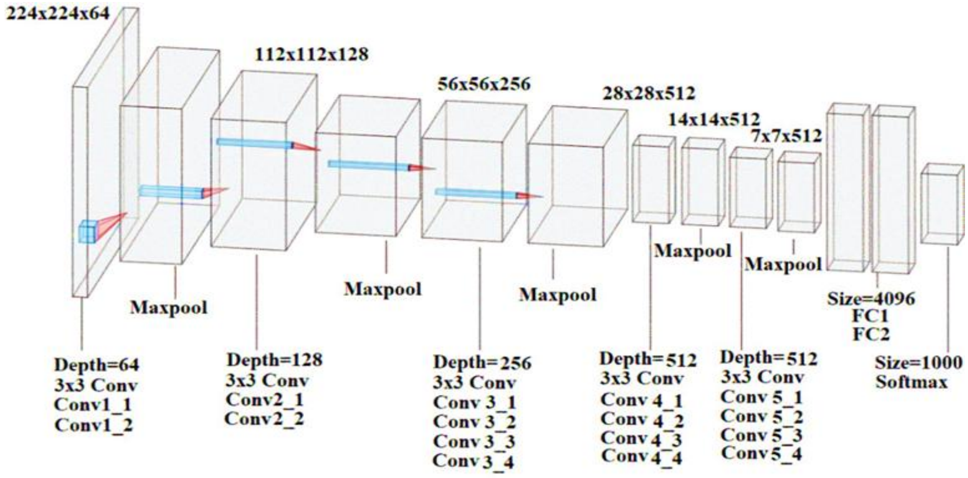

Figure 2: VGG19 Architecture

Deep Neural Networks (DNNs)

Deep Neural Networks (DNNs) can be considered a more intricate version of the human brain's communication system. They are built from layers of interconnected units, commonly known as neurons, each performing a relatively simple calculation. Stacking these layers one after another allows the network to learn more complex patterns gradually. It's similar to how children learn: first recognizing simple shapes, then putting them together into objects, and eventually understanding entire scenes. Figure 3 illustrates the DNN architecture [24].

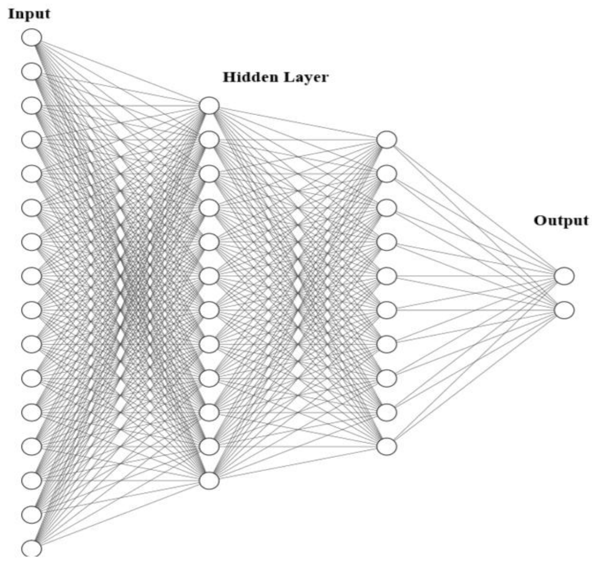

Figure 3: Basic DNN Architecture

Evaluation metrics

Evaluation metrics are essential tools for assessing the performance of classification models, particularly in applications like plant disease detection. They provide

insights into how well a model can classify data, identify errors, and balance trade-offs between outcomes. The evaluation metrics include accuracy, precision, recall, and F1-score.

Accuracy evaluates the proportion of correctly classified samples out of the total samples. Equation 1 calculates the accuracy of the model:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad \ldots 1$$

Where: TP is the number of instances correctly predicted as belonging to the positive class, TN is the number of instances correctly predicted as belonging to the negative class, FP is the number of instances incorrectly predicted as belonging to the positive class, and FN is the number of instances incorrectly predicted as belonging to the negative class.

Recall (or Sensitivity) indicates the proportion of actual positives correctly identified by the model. Equation 2 calculates the Recall of the model:

$$Recall = \frac{TP}{TP+FN} \quad \ldots 2$$

Precision measures the proportion of true positive predictions among all positive predictions. Equation 3 calculates the Precision of the model:

$$Precision = \frac{TP}{TP+Fp} \quad \ldots .3$$

F1-Score combines precision and recall into a single metric to balance their trade-offs. Equation 4 calculates the F1-Score of the model:

$$F1_{Score} = 2\,\frac{Preceision\times Recall}{Preceision+ Recall} \quad \ldots 4$$

The Proposed Model

The proposed mode of plant disease detection is based on transfer learning and swarm algorithms. It consists of several stages, starting from preprocess data to classify the plant disease. The proposed model consists of four major stages: pre-processing, feature extraction, feature selection, and evaluation of the proposed model. Each of these stage plays a crucial role in the overall effectiveness and efficiency of the disease detection system. Figure 4 illustrates the essential stage of the proposed model design.

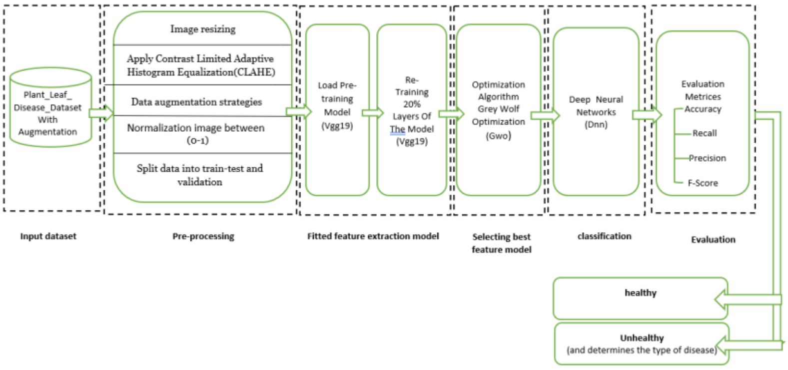

Figure 4: Proposed model design

Data Pre-processing Techniques

This deals with data pre-processing before feeding them to the machine learning models. During the preprocessing stage, data transformations and cleaning steps are employed to improve the data acquired from different sources so that the results are reliable. This step includes Image Resizing and Normalization, Image enhancement using histogram normalization and Data Augmentation Strategies. By carefully handling the data in this stage, any inconsistencies or errors are minimized, thus leading to a more accurate and reliable disease classification process [19].

Image Resizing

Image resizing is the process of changing the dimensions of an image to fit the requirements of a particular model or application. Most machine learning models, especially convolutional neural networks (CNNs), require input images to be of a consistent size [25]. Therefore, in this uniform model, the size of the images (in both train and test models) is uniform to make the deep learning work properly. This work used the nearest-neighbor interpolation method to resize and uniform the image. It works by selecting the value of the nearest pixel to determine the value of a new pixel when enlarging or reducing an image. This technique does not consider other surrounding pixel values, making it computationally efficient. It can lead to jagged edges or a blocky appearance, especially when enlarging images [25].

Image Contrast Enhancement

Enhance Image Contrast refers to technologies highlighting details and altering the contrast between dark and light. This work uses Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve image contrast. It is designed to improve the contrast of images by working on small regions (tiles) rather than the entire image at once, ensuring that the enhancement is adaptive to local variations in intensity. This approach prevents noise over-amplification and reduces the risk of introducing unwanted artifacts. Figure 5 shows an example with the main steps of Contrast Limited Adaptive Histogram Equalization (CLAHE):

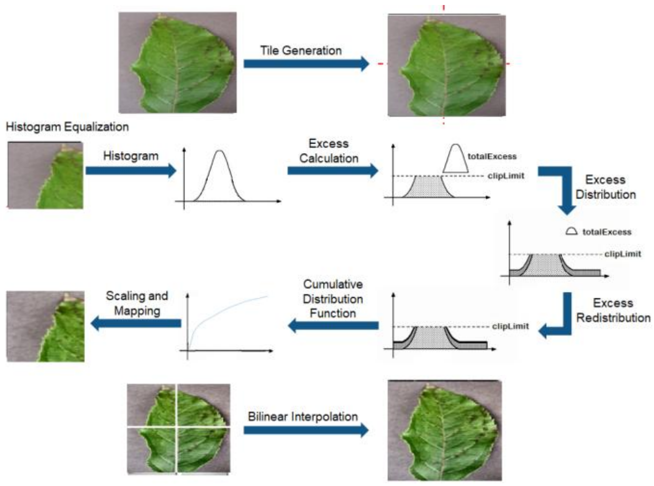

Figure 5: Example and main steps of CLAHE

Data Augmentation Strategies

Image augmentation techniques have become very common in data pre-processing. They range from rotation, flipping, scaling, and image shifting. Random rotations increase hyperparameter tuning but help increase the training data diversity [26]. Flipping images, either horizontally or vertically, helps increase the image diversity representation when there is reflection symmetry; images must not be flipped. Moreover, images should only be flipped vertically if their context involves a human head or facial and perspective objects close to us. Scaling invariance of training data augmentations may be of interest, especially in the case of object detection. Object scaling is irrelevant for object localization, which does not learn the object's identity. Until recently, data distortion techniques like elastic distortion were introduced [19].

Image Normalization

Image normalization is a fundamental pre-processing technique in image processing and computer vision. It involves adjusting the range of pixel intensity values in an image to a standard scale or distribution. This work used Min-Max Normalization (Feature Scaling). This method scales the pixel values to a specific range, typically between 0 and 1 or 0 and 255. Equation 5 calculates the pixel normalization (I_{norm}):

$$I_{\text{norm}} = \left(\frac{I - I_{\min}}{I_{\max} - I_{\min}}\right) \quad \cdots 5$$

Where: I represent original pixel intensity, $I_{\min}, I_{\max}$: represent minimum and maximum pixel intensities in the original image.

Split data into train-test and validation

When building machine learning models, evaluating their performance accurately is crucial. To achieve this, data is typically split into three distinct sets: training, validation, and test datasets. Each of these

datasets serves a specific purpose in the model development process [27].

Transfer learning for feature extraction

Transfer learning is a powerful technique in deep learning where a pre-trained model, such as VGG19, is repurposed for a new task, leveraging its learned knowledge from a large dataset like ImageNet. This approach significantly reduces the computational cost and training time required to build a model from scratch, especially when the target dataset is small or domain-specific [3].

Feature extraction in VGG19 can be done by retaining only certain layers of the model, normally the convolutional layers that contain high-hierarchical features like edges, textures and patterns [12]. For example, retaining 10% of the original VGG19 architecture combines concentration on establishing the bottom-level features of vision and elimination of deeper layers for classification. Such a truncated approach is useful because it allows the model to be used as a feature extractor to create useful, interpretable and usable representations of the input data for other models or classifiers to work with. Such applications are especially useful where features extracted from the images are used in enhancing the results such as image quality, detected objects, or in compressing fine-grained image analysis. In the proposed model, VGG19 was used due to its effectiveness in performing hierarchical feature extraction as observed in other studies [11]. The pre-train model has been gained from extensive pre-training on large-scale image datasets. Rather than building a network from scratch, we rely on VGG19's strong, pre-learned representation of visual patterns—ranging from low-level edges and textures to high-level shapes and objects—and then adjust only a fraction of its parameters to fit our specific task. This selective fine-tuning helps adapt the model to our domain with minimal data and computational demands, ensuring that we benefit from its rich foundational features while avoiding the pitfalls of full-scale retraining on a limited dataset [16]. In the proposed work, the pre-trained VGG19 model is imported into the ImageNet database. Then, all of the model's layers are frozen to preserve their original weights. Only the top layers are frozen, while the lower 20% are frozen for retraining using new data. Figure 6 shows the main Illustration of TL method.

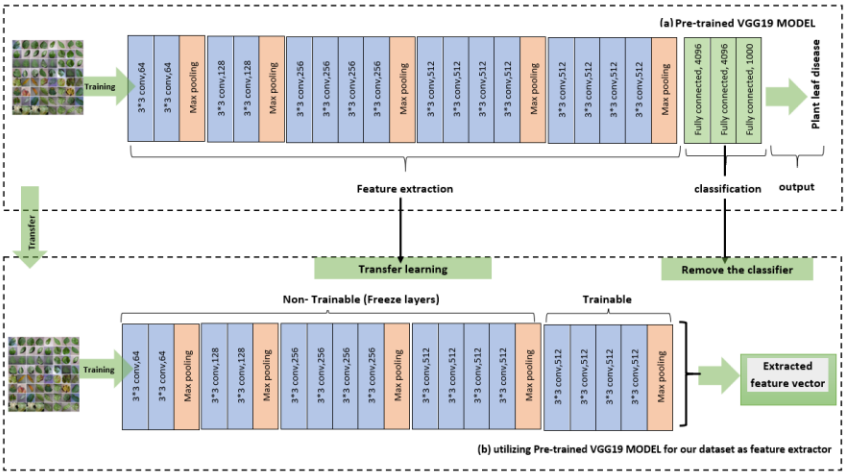

Figure 6: Illustration of TL method

Metaheuristic for feature selection

Metaheuristics are high-level strategies that aim to find, generate, or select a heuristic that provides a sufficiently good solution to an optimization problem, especially in incomplete or imperfect information. One such application is feature selection, which aims to identify the most relevant features for a given data set to improve the performance of machine learning models [21]. The GWO is used as an optimizer to select features suitable for the proposed model's classification step.

- **Feature selection strategy**

The proposed feature selection strategy employs a real-number encoding mechanism and a threshold-based decision rule to identify the most relevant features from a dataset. This approach leverages the flexibility and exploration capabilities of metaheuristic optimization methods while addressing the limitations of traditional binary encoding.

- **Representation of Features**

In the dataset, features are given in terms of real values normally from [0,1]. This encoding provides a probabilistic solution to the feature-selection process of arriving at the final subset. The designated real-number values represent the probability of selecting the feature mentioned in the corresponding subset into the final subset. The solution vector represents a candidate feature subset as a real-valued vector of dimension n of the total feature in the dataset. For instance, a one solution vector [0.2,0.8,0.6,0.3] indicates potential inclusion probabilities of four features, (n=4). The solution vectors are created interactively during the optimization routine based on the chosen metaheuristic algorithm which can be PSO, GA or GWO. This work used GWO as a metaheuristic algorithm to generate a solution factor.

By encoding features as real numbers and applying a threshold for inclusion, this strategy balances exploration of the solution space and fine-tuning of feature subsets. The initial random generation ensures diverse exploration, while iterative updates optimize the feature selection process to identify subsets that maximize the objective function.

- **Threshold Mechanism**

A predefined threshold T is applied to the real-number encoded vector to determine feature inclusion. Equation 6 represents how the selected feature is based on the threshold value (T):

$$f(x) = \begin{cases} x \geq T, & selected\ featuer \\ otherwise, & remove\ veatuer \end{cases} \qquad \textbf{... 6}$$

If the value of a feature $x_i \geq T$, the feature is selected; otherwise, it is excluded. For instance, with $T = 0.5$, the vector [0.2,0.8,0.6,0.3] results in the selected subset [remove, selected , selected , remove], indicating the inclusion of the second and third features.

For example, if there is a dataset with 10 features and 5 samples, and T=0.5, the selected feature would be:

optimization vector =

0.4	0.7	0.9	0.3	0.8	0.2	0.5	0.6	0.1	0.7
R	S	S	R	S	R	R	S	R	S

Dataset =

1.2	0.8	3.1	2.4	4.1	3.7	0.9	1.8	2.5	1.1
1.5	0.7	2.9	2.2	4.0	3.5	0.8	1.7	2.4	1.0
1.3	0.9	3.0	2.5	4.2	3.6	1.0	1.9	2.6	1.2
1.4	0.8	2.8	2.3	4.1	3.4	0.9	1.8	2.5	1.1
1.6	0.7	3.2	2.6	4.3	3.8	1.1	2.0	2.7	1.3

In the above scenario, the optimization model selects 5 features over 10. The Threshold T can be fixed or dynamically adjusted during the optimization process. A dynamic threshold strategy begins with a higher threshold to select fewer, highly significant features and gradually reduces the threshold to refine the feature subset as the algorithm converges.

Using a static threshold (T) in real-number-based feature selection strategies has several limitations. Here are the key drawbacks:

- Lack of Adaptability to Dataset Characteristics;
- The performance of the selection process heavily depends on the initial choice of T;
- Reduced Flexibility for Metaheuristic Algorithms;
- Possible to make constant Feature selections across Iterations.

The dynamic thresholds provide flexibility and improve the adaptability of the feature selection process, enabling better alignment with the dataset's characteristics and the metaheuristic's optimization behavior:

$$T = (T_{min} + (T_{max} - T_{min}) * \frac{i}{N}) * r \qquad \text{... 7}$$

Where T_{min},T_{max} represent a maximum and minimum threshold in value (0,0.5) (, i current iteration, N maximum iteration r random value in interval [0,1].

In the proposed approach, the dynamic threshold starts at a minimum value (T_{min}) during the initial iterations of the algorithm. As the search progresses, the threshold gradually increases based on the iteration index and other factors, ensuring that the threshold adapts to the search progress. This gradient-based adjustment allows the algorithm to initially focus on selecting only strongly relevant features and later include features with marginal relevance, optimizing the feature subset selection process. With an additional condition that triggers a modification in T when stagnation is identified. This approach makes the dynamic threshold adaptive to the iteration progress and the algorithm's performance, balancing exploration and exploitation throughout the optimization process. Algorithm 1 illustrates the proposed feature selection algorithm.

Algorithm 1: Proposed dynamic features selection

Input:

- Dataset D, Maximum Iterations N, Stagnation Threshold $N_{stagnation}$,

Objective Function F, T_{min}, T_{max}

Output:

- Optimal Feature Subset S*

Begin

1. **Initialize**

population S with random real-number vectors Set initial best solution S*

T = T_{max} *Initial* iteration i = 1 and stagnation counter c = 0

2. While i ≤ N do
For each solution s_j in S do
B_j ← Select features based on T (Eq.6) and optimization vector (s_j)
Evaluate fitness F(B_j)
If F(B_j) > F(S*): //
Update S* = B_j
Reset stagnation counter c = 0
Else stagnation counter c = +1
If c > $N_{stagnation}$
Update T = (T_{min} + (T_{max}- T_{min}) * i / N) * r
End For
End While
3. Return Optimal Feature Subset (S*)
4. Return Optimal Feature Subset (S*)
End

Figure 7 shows the distribution of T over the iterations of the optimization process.

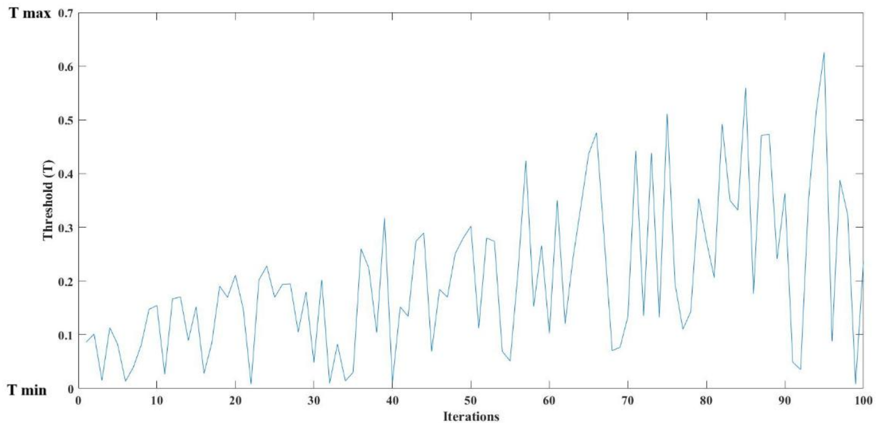

Figure 7: Distribution of the dynamic value of T

- **Integration with Metaheuristics**

The real-number encoding and threshold mechanism are seamlessly integrated into the metaheuristic algorithm. This work used the GWO optimization algorithm as a metaheuristic algorithm to optimize real-number vectors in the continuous search space.

RESULT DISCUSSION

This section discusses the effectiveness of the proposed methods, focusing on their performance improvements compared to traditional approaches. The findings validate the models' ability to address specific challenges, highlighting their potential for practical applications and further advancements. In this section, we will discuss the results in three ways: analysis of the training phase, comparison of the proposed model with machine learning algorithms, and finally, comparison of the proposed model with some plant disease classification methods.

Training result

Figure 8 illustrates the training and validation accuracy of the proposed model, demonstrating a clear trend of increasing accuracy over the training epochs. This indicates that the model is effectively learning from the training data, as evidenced by the gap between the training and validation accuracy becoming smaller, suggesting good generalization.

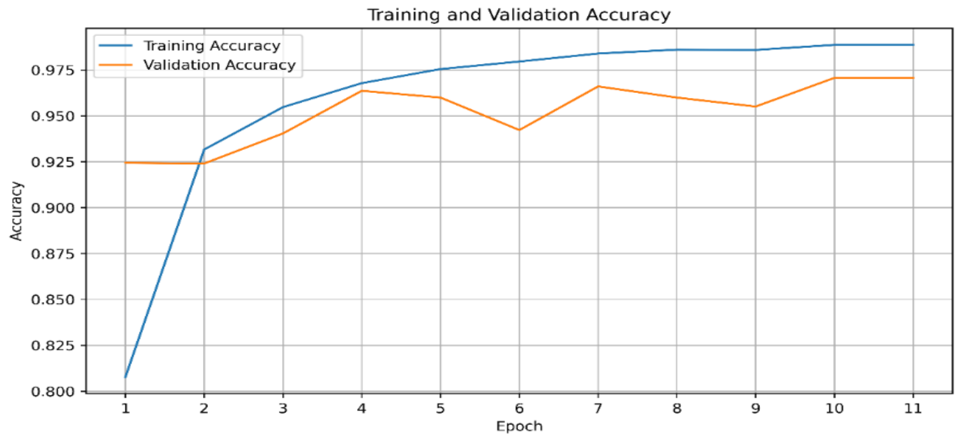

Figure 8: Training and validation accuracy of the proposed model

Figure 9 presents the Grey Wolf Optimization (GWO) convergence accuracy used for feature selection. This figure likely showcases how the feature selection process improves the model's performance by honing in on the most relevant features, contributing to more efficient training and potentially higher accuracy in classification tasks.

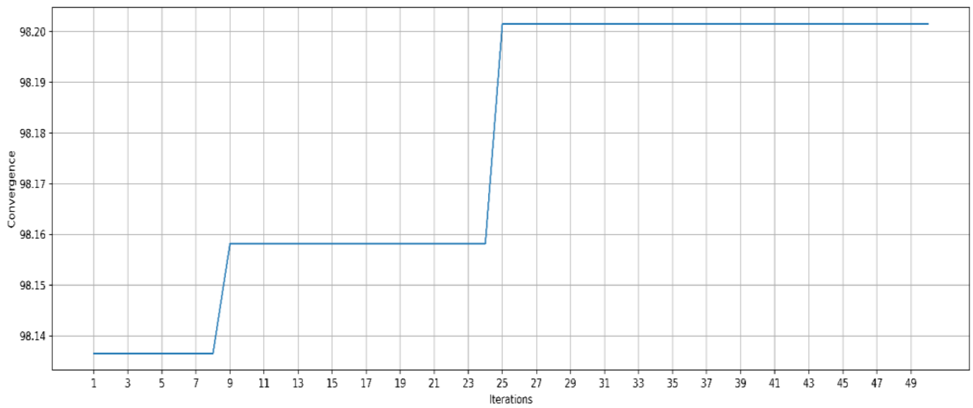

Figure 9: Convergence accuracy of GWO of feature selection

Table 1: Accuracy of the model for classifying plant diseases and health conditions

	TP	TN	FP	FN	Accuracy
Apple_Apple_scab	162.0	9065.0	1.0	2.0	99.967
Apple_Black_rot	147.0	9081.0	1.0	1.0	99.978
Apple_Cedar_apple_rust	159.0	9070.0	1.0	0.0	99.989
Apple healthy	248.0	8980.0	2.0	0.0	99.978
Background_without_leaves	169.0	9061.0	0.0	0.0	100
Blueberry healthy	209.0	9021.0	0.0	0.0	100
Cherry Powdery mildew	151.0	9078.0	1.0	0.0	99.989
Cherry healthy	144.0	9085.0	1.0	0.0	99.989
Corn_Cercospora_leaf_spot Gray_leaf_spot	155.0	9068.0	6.0	1.0	99.924
Corn Common rust	187.0	9041.0	0.0	2.0	99.978
Corn_Northern_Leaf_Blight	149.0	9076.0	1.0	4.0	99.946
Corn healthy	182.0	9048.0	0.0	0.0	100
Grape Black rot	182.0	9046.0	1.0	1.0	99.978
Grape Esca_(Black Measles)	207.0	9021.0	1.0	1.0	99.978
Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	158.0	9072.0	0.0	0.0	100
Grape healthy	160.0	9069.0	0.0	1.0	99.989

Orange_Haunglongbing_ (Citrus greening)	794.0	8434.0	1.0	1.0	99.978
Peach_Bacterial_spot	333.0	8894.0	0.0	3.0	99.967
Peach healthy	144.0	9085.0	1.0	0.0	99.989
Pepper, _bell_Bacterial_spot	128.0	9100.0	1.0	1.0	99.978
Pepper, _bell healthy	213.0	9015.0	1.0	1.0	99.978
Potato_Early_blight	155.0	9069.0	1.0	5.0	99.935
Potato_Late_blight	162.0	9062.0	3.0	3.0	99.935
Potato healthy	174.0	9056.0	0.0	0.0	100
Raspberry healthy	157.0	9073.0	0.0	0.0	100
Soybean healthy	769.0	8461.0	0.0	0.0	100
Squash_Powdery_mildew	273.0	8956.0	1.0	0.0	99.989
Strawberry Leaf scorch	174.0	9056.0	0.0	0.0	100
Strawberry healthy	175.0	9055.0	0.0	0.0	100
Tomato Bacterial spot	329.0	8900.0	1.0	0.0	99.989
Tomato_Early_blight	154.0	9066.0	4.0	6.0	99.892
Tomato_Late_blight	271.0	8948.0	5.0	6.0	99.881
Tomato_Leaf_Mold	133.0	9093.0	2.0	2.0	99.957
Tomato_Septoria_leaf_spot	237.0	8986.0	4.0	3.0	99.924
Tomato_Spider_mites Two-spotted_spider_mite	245.0	8977.0	4.0	4.0	99.913
Tomato_Target_Spot	214.0	9008.0	5.0	3.0	99.913
Tomato_Tomato_Yellow_Leaf_Curl_Virus	779.0	8449.0	1.0	1.0	99.978
Tomato_Tomato_mosaic_virus	154.0	9076.0	0.0	0.0	100
Tomato healthy	242.0	8987.0	1.0	0.0	99.989

Table 1 presents Accuracy of the model for classifying plant diseases and health conditions. The model demonstrates very strong performance in classifying plant diseases and health conditions, achieving an overall high accuracy of between 99.881% and 100% across all categories.

• Healthy categories (e.g., "healthy") obtained 100% accuracy in several cases, including tomatoes, strawberries, soybeans, and corn. This demonstrates the model's capacity to correctly and easily recognize healthy situations. • For diseases with clear and distinct symptoms, such as Tomato_Yellow_Leaf_Curl_Virus and Grape_Leaf_Blight, the model reached near-perfect accuracy. This demonstrates the clarity of visual patterns for these diseases in training data.

• Categories, such as Tomato_Early_blight and Potato_Early_blight, showed slightly lower.

Compare the proposed model with machine learning algorithms

The results for machine learning comparison with the proposed model can be seen in Table 2 below. The results of the plant disease classification task present a comparative analysis of several machine learning models, demonstrating their efficacy in classifying plant diseases using various metrics, like accuracy, precision, recall, and F1 score. Below is the analysis of each model's performance.

Table 2: Comparison of performance of the proposed model and traditional machine learning

Model	Accuracy	Precision	Recall	F1 Score
XGBClassifier	88.99	88.79	88.71	88.75
KNeighborsClassifier	89.25	89.14	89.02	89.07
DecisionTreeClassifier	83.18	81.48	81.48	81.46
RandomForestClassifier	89.21	89.18	88.97	89.07
Proposed model	99.44	99.32	99.31	99.31

The findings show the performance of traditional models such as Random Forest and KNN is satisfactory, But the proposed model is significantly better, achieving an accuracy of 99.44%. This brings questions to methods and techniques used in the proposed model since it establishes a new standard in plant disease classification.

The results discussed in this study demonstrate that several aspects of the proposed model may be very promising for practical use, including the enhancement of agricultural productivity by recognizing diseases on plants at the right time. However, it is crucial to consider the complexity of the model, its interpretability, and the computational demands it may require when implemented. Future investigation could tackle the examination sensitivity of the proposed model to other datasets, exploration of how it performs when future diseases come into play, as well as evaluation of its performances when in use in the real world.

Compare the proposed model with other models

The proposed model is compared with other existing methods for plant disease classification in terms of accuracy in Table 3. The results further show that the proposed model has an impressive accuracy of 99.44%, thus outperforming all other listed methods. To elaborate on this, the Transfer Learning using VGG16 achieved 98.4%. another Transfer Learning model VGG19 that was developed for paddy leaf disease detection achieved 96.4 %. while optimization model developed with Particle Swarm Optimizer (PSO) achieved 98.83%. In addition, the fine-tuning of MobileNetV2 achieved 94.4%. These findings confirm the effectiveness of the proposed model in achieving enhanced classification and could be used as a useful tool for increasing the rate of accuracy of plant disease detection.

Table 3: Comparison results of the proposed model with other plant disease classifications

Ref	Method Summary	Result (Accuracy)
[11]	Transfer Learning for Multi-Crop Leaf Disease Classification using CNN (VGG16)	98.4%
[12]	Transfer Learning-Based AI Model for Paddy Leaf Disease Detection	96.4%
[13]	Optimization of Deep Learning Model for Plant Disease Detection Using Particle Swarm Optimizer (PSO)	98.83%
[17]	Detection of plant leaf disease using advanced deep learning architectures	94.4%
Proposed Model		99.44%

CONCLUSION

This work introduces a hybrid intelligent model for plant disease classification, which is based on deep transfer learning and Grey Wolf Optimization to overcome the drawbacks of existing detection techniques. Through the application of advanced feature extraction and selection, the built model effectively acquires an accuracy of 99.44% which surpasses commonly used machine learning methods and other dominant works. These results show the effectiveness of the proposed model in tackling complex classification problems with high accuracy. The model's potential suggests that the designed model can easily be applied to real-life farming contexts to bring about early disease detection and low crop loss. As future work, it could consider using it on different datasets, real-time monitoring, and in combination with existing IoT-based agriculture platforms to increase the application potential.

REFERENCES

[1] S. Ashwinkumar, S. Rajagopal, V. Manimaran, and B. Jegajothi, "Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks," *Materials Today: Proceedings,* vol. 51, pp. 480-487, 2022.

[2] V. Singh, N. Sharma, and S. Singh, "A review of imaging techniques for plant disease detection," *Artificial Intelligence in Agriculture,* vol. 4, pp. 229-242, 2020.

[3] H. Orchi, M. Sadik, and M. Khaldoun, "On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey," *Agriculture,* vol. 12, no. 1, p. 9, 2021.

[4] I. Buja *et al.*, "Advances in plant disease detection and monitoring: From traditional assays to in-field diagnostics," *Sensors,* vol. 21, no. 6, p. 2129, 2021.

[5] A. Terentev, V. Dolzhenko, A. Fedotov, and D. Eremenko, "Current state of hyperspectral remote sensing for early plant disease detection: A review," *Sensors,* vol. 22, no. 3, p. 757, 2022.
[6] A. Gojon *et al.*, "Approaches and determinants to sustainably improve crop production," *Food and Energy Security,* vol. 12, no. 1, p. e369, 2023.
[7] K. Chen *et al.*, "Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data," *Water research,* vol. 171, p. 115454, 2020.
[8] R. A. Arun and S. Umamaheswari, "Effective multi-crop disease detection using pruned complete concatenated deep learning model," *Expert Systems with Applications,* vol. 213, p. 118905, 2023.
[9] D. Liptzin *et al.*, "An evaluation of carbon indicators of soil health in long-term agricultural experiments," *Soil Biology and Biochemistry,* vol. 172, p. 108708, 2022.
[10] A. Ghorbani, L. Riazi, A. Daneshvar, and R. Radfar, "Feature engineering with gray wolf algorithm and fuzzy methods for friend recommender system in social networks," *Journal of Industrial and Systems Engineering,* vol. 14, no. 3, pp. 109-120, 2022.
[11] A. S. Paymode and V. B. Malode, "Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG," *Artificial Intelligence in Agriculture,* vol. 6, pp. 23-33, 2022, doi: 10.1016/j.aiia.2021.12.002.
[12] V. Gautam *et al.*, "A Transfer Learning-Based Artificial Intelligence Model for Leaf Disease Assessment," *Sustainability,* vol. 14, no. 20, 2022, doi: 10.3390/su142013610.
[13] A. Elaraby, W. Hamdy, and M. Alruwaili, "Optimization of Deep Learning Model for Plant Disease Detection Using Particle Swarm Optimizer," *Computers, Materials & Continua,* vol. 71, no. 2, pp. 4019-4031, 2022, doi: 10.32604/cmc.2022.022161.
[14] A. Tabbakh and S. S. Barpanda, "A Deep Features Extraction Model Based on the Transfer Learning Model and Vision Transformer "TLMViT" for Plant Disease Classification," *IEEE Access,* vol. 11, pp. 45377-45392, 2023, doi: 10.1109/access.2023.3273317.
[15] A. Nayak, S. Chakraborty, and D. K. Swain, "Application of smartphone-image processing and transfer learning for rice disease and nutrient deficiency detection," *Smart Agricultural Technology,* vol. 4, 2023, doi: 10.1016/j.atech.2023.100195.
[16] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture,* vol. 9, pp. 22-35, 2023, doi: 10.1016/j.aiia.2023.07.001.
[17] R. Sharma, M. Mittal, V. Gupta, and D. Vasdev, "Detection of plant leaf disease using advanced deep learning architectures," *International Journal of Information Technology,* vol. 16, no. 6, pp. 3475-3492, 2024, doi: 10.1007/s41870-024-01937-4.
[18] W. Shafik, A. Tufail, C. De Silva Liyanage, and R. A. A. H. M. Apong, "Using transfer learning-based plant disease classification and detection for sustainable agriculture," *BMC Plant Biology,* vol. 24, no. 1, p. 136, 2024.
[19] A. Naseer, M. Amjad, A. Raza, K. Munir, N. A. Samee, and M. A. Alohali, "A Novel Transfer Learning Approach for Detection of Pomegranates Growth Stages," *IEEE Access,* vol. 12, pp. 27073-27087, 2024, doi: 10.1109/access.2024.3365356.
[20] T. Ibarra-Pérez *et al.*, "A Performance Comparison of CNN Models for Bean Phenology Classification Using Transfer Learning Techniques," *AgriEngineering,* vol. 6, no. 1, pp. 841-857, 2024, doi: 10.3390/agriengineering6010048.
[21] M. Rezaei, D. Diepeveen, H. Laga, M. G. K. Jones, and F. Sohel, "Plant disease recognition in a low data scenario using few-shot learning," *Computers and Electronics in Agriculture,* vol. 219, 2024, doi: 10.1016/j.compag.2024.108812.
[22] M. T. El-Saadony *et al.*, "Plant growth-promoting microorganisms as biocontrol agents of plant diseases: Mechanisms, challenges and future perspectives," *Frontiers in plant science,* vol. 13, p. 923880, 2022.
[23] A. H. Alsaeedi *et al.*, "Dynamic Clustering Strategies Boosting Deep Learning in Olive Leaf Disease Diagnosis," *Sustainability,* vol. 15, no. 18, p. 13723, 2023. [Online]. Available: https://www.mdpi.com/2071-1050/15/18/13723.
[24] H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, "Transfer learning for medical image classification: a literature review," *BMC medical imaging,* vol. 22, no. 1, p. 69, 2022.

[25] A. S. A. Mohammad R Aziz, "Feature Selection of The Anomaly Network Intrusion Detection Based on Restoration Particle Swarm Optimization," *International Journal of Intelligent Engineering and Systems,* vol. 15, no. 5, pp. 592-600, 2022, doi: 10.22266/ijies2022.1031.51.

[26] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools," *Smart Agricultural Technology,* vol. 3, p. 100083, 2023.

[27] Z. Chang, K. Jia, T. Han, and Y.-M. Wei, "Towards more reliable photovoltaic energy conversion systems: A weakly-supervised learning perspective on anomaly detection," *Energy Conversion and Management,* vol. 316, p. 118845, 2024.