Received: April 6, 2024. Revised: April 30, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Coati Optimized Transfer Leaning with Vision Transformer Model for Improving Deep Learner based Plant Diseases Detection

Pandurangan Indumathi[1]* **Rathinasamy Kumuthaveni[2]**

[1]*Department of Computer Applications,*
Kongunadu Arts and Science College, Coimbatore -641029, Tamil Nadu, India
[2]*Department of Artificial Intelligence and Machine Learning,*
KPR College of Arts Science and Research, Coimbatore -641407, Tamil Nadu, India
* Corresponding author's Email: induaarthiphd@gmail.com

Abstract: Plants provides a substantial contribution to the world's food supply. Plant diseases significantly damages the agriculture crops, reducing production and lowering crop grade and availability. Rapid and precise identification of leaf diseases is critical for long-term agricultural output increase. For this reason, Transfer Learning (TL)-based Model with Vision Transformer (TLMViT) was developed for plant diseases detection. This model utilizes the pre-trained models and ViT for feature extraction followed by Multi-Layer Perceptron (MLP) for classification. But, the weights of pre-trained CNN model layers were not properly fine-tuned leading to high computational complexity and lower accuracy. To solve this, an Extended Stochastic Coati Optimized Transfer Leaning with Vision Transformer (ESCOTLViT) model is developed to fine-tune the weights of pre-trained CNN models for efficient plant disease detection. In this method, Extended Stochastic Coati Optimizer (ESCO) is adopted to identify the ideal weight of layers of the pre-trained models and parameters like number of neurons, hidden units, epochs, learning rate, weight decay, batch size, dropout rate, number of partitions, number of clusters per batch, momentum, optimizer and loss function. ESCO differs from Coati Optimization Algorithm (COA) by dividing the population into two fixed groups and employing three sequential phases in each iteration. In ESCO, an initial population of each coati represents the weight layers of pre-trained CNN models. The computation of each coati's subsequent locations represents the search space for the best weights layers to identify the optimal values for pre-trained models. The evaluation of these individuals are done using a fitness function. Individuals with better fitness values are more likely to be selected for directing others with ideal positions i.e., layers in CNN variations for weight optimization. According to this ESCO, the weight layers of pre-trained CNN models in TLMViT are optimized for plant diseases detection. Finally, the test findings revealed that the ESCOTLViT model achieves 94.22%, 94.81% and 94.42% of accuracy on PlantVillage, PlantDoc and DiaMOS Dataset respectively compared to the existing models like Convolutional Neural Network-Vision Transformer (CNN-ViT), Optimum Mobile Network-based CNN (OMNCNN), Convolutional Block Attention Module (CBAM), DeepplantNet, Attention Mechanism with MobileNet V2 (AM-MNV2), Improved Quantum Whale Optimization with Principle Component Analysis and Deep Neural Network (IQWO-PCA-DNN) and TLMViT.

Keywords: Plant diseases deep learning, Coati optimizer, Pre-trained CNN models, Vision transformer.

1. Introduction

Agriculture is crucial for global food security, but challenges like population expansion, climate change, arable land scarcity and plant diseases pose additional obstacles [1]. These diseases reduce crop yield and food productivity, affecting national and global food production systems [2]. Rapid recognition and anticipation of plant diseases are essential for increasing food yield, managing illnesses, minimizing famine and ensuring adequate food supply [3].

Classical plant disease detection requires professional inspection and ongoing monitoring systems, which can be costly and time-consuming for large farms [4]. Farmers often lack access to experts, making consulting more expensive in some countries. Human bias and fatigue can also affect the accuracy of these procedures [5]. Image processing algorithms have been developed to detect plant diseases using collected images, including procurement, pre-processing, segmentation, feature extraction and categorization [6]. Pathologists can forecast plant diseases by analysing changes in color, texture, spot and size in captured images. This framework aids in high-precision plant analysis for repair and future disease prediction, benefiting farmers in various agricultural activities [7]. Despite its potential benefits, Image processing faces challenges like lower accuracy on larger data samples and the time-consuming and potentially biased process of human feature extraction [8].

In recent times, Artificial Intelligence (AI) models like Machine Learning (ML) and Deep Learning (DL) are increasingly used in plant pathology to identify, manage and prevent diseases and infestations [9]. When compared to the ML frameworks, DL models offer efficient plant disease prediction and classification, reducing crop treatment costs and increasing productivity without human intervention [10]. Some of the DL algorithms like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long- Short Time Memory (LSTM), Deep Belief Network (DBN) etc. These algorithms aid pathologists to detect plant diseases based on unique features from images, improving crop productivity, disease management and recognition accuracy [11].

Several deep learning models have been created to identify and categorize plant diseases based field images. For example, a hybrid model termed TL-based Model followed by a Vision Transformer (TLMViT) was developed for plant disease categorization [12]. Initially, THE dataset was expanded to address overfitting and increase training samples. Leaf features were extracted through two phases: initial features extraction using pre-trained CNN models (AlexNet, Res-Net 50, VGG-16, VGG-19 and Inception-V3) and deep features extraction using ViT model. Multi-Layer Perceptron (MLP) was used for plant disease classification. However, the hyper-parameters of the pre-trained models were not optimized, leading to high computational complexities and lower accuracy rate.

To solve this, ESCOTLViT model is proposed in this paper for the efficient plant diseases prediction and classification. In this paper, the ESCO [13] model is employed to select the optimal weight layers of the pre-trained models like number of neurons, hidden units, learning rate, weight decay, number of epochs, batch scale, dropout rate, partitions, clusters per batch, momentum, optimizer and loss function. The COA [14] proves its inefficiency while generating low maximum iteration and low population size circumstances. This can be resolved by ESC by expanding the sequential phase, references in the guided search, number of searches and shifting the fixed split to a stochastic split in the roles segregation and references used in COA. ESCO also implements a stochastic process for each unit to choose the searches that will perform. It differs from COA, which splits the population into two fixed groups, each performing its strategy. ESCO implements three sequential phases in every iteration. Two options can be chosen in every phase. ESCO has three references in its guided search: the global best unit, a randomly selected unit and a randomized unit within the search space.

In ESCO, the central premise is to mimic two of the coati's natural behaviours: (i) chasing and devouring iguanas and (ii) evading enemies. The ESCO method employs an initial population of individuals assigning each coatis' position in the search space dictating the values of decision variables or weight optimization layers. The coatis' posture represents a potential solution in optimizing the pre-trained model's parameter. The fitness function measures an individual's ability to find the best solution and assesses their performance. Individuals with higher fitness values are more likely to be selected to guide others to optimal locations and adjust weights in CNN variants. Finally, the weight layers of AlexNet, Res-Net 50, VGG-16, VGG-19 and Inception-V3 models will be optimized by using ESCO to enhance leaf disease detection accuracy.

The rest of this article is arranged as the following: different works associated with the plant diseases identification and categorization models are presented in Section II. Section III explains the proposed ESCOTLViT model whereas Section IV displays its validity. Section V outlines the entire study and discusses the upcoming enhancement.

2. Literature survey

2.1 Survey on DL based plant leaf diseases classification models

Channel-Spatial Segmentation Network (CSSN) model [15] was developed for classifying leaf infections. But, the model's hyperparameters were not optimized properly leading to lower accuracy results. An Improved YOLOV3-Tiny model [16] was suggested for the early detection of turmeric plant diseases. But, lower accuracy was resulted as it was trained with limited dataset.

A CNN model was integrated with ViT to autonomously classify plant diseases using ViT [17]. However, its accuracy was not inefficient while performing on large-scale datasets. An OMNCNN was used to identify the conditions affecting plant leaves [18]. But its precision was less due to the limited number of samples.

A plant disease classification method was developed by using One-Shot Learning (OSL) and Siamese Neural Network (SNN) models [19]. However, this approach was sensitive to outliers which lowers the detection accuracy. A CBAM [20] was developed to classify the plant disease. But, models hyper-parameters were not properly fine-tuned, resulting in lower accuracy results.

A DeepplantNet model was developed for plant leaf diseases detection and classification [21] using convolutional layers and deeper layers for disease detection. But when the data samples increases, the model lowers its performance accuracy. A Deep Transfer Learning (DTL) method [22] was suggested for detecting heterogeneous diseases in plant images. But, lower accuracy and F1-score results were obtained when handling the larger datasets

The plant disease detection and classification algorithm [23] was developed by AM-MNV which extracts both spatial and channel dimensions for plant-disease recognition. However, the maximal training duration leads in high localization errors and lower accuracy rate. An IoT-enabled plant disease prediction system was developed using IQWO-PCA-DNN [24]. However, this model results with high computation time and lower sensitivity values.

2.2 Survey on metaheuristic optimization models

A new metaphor-free metaheuristic search model called Swarm Bipolar Algorithm (SBA) was introduced [25] based on splitting the swarm into two equal-sized swarms to diversify the searching process while performing intensification within the sub-swarms. But, the computational complexity was high compared to other metaheuristic models significantly lowers the accuracy.

A novel metaphor-free swarm-based metaheuristic model called Swarm Space Hopping Algorithm (SSHA) was presented [26] which constitutes of three searches i.e., two directed searches and one crossover-based search. But, computational complexity was high and utilizes only the uniform distribution which affects the performance efficiency.

A new swarm metaheuristic model called Migration-Crossover Algorithm (MCA) was developed [27] which uses crossover technique and neighborhood local search space. It uses the global finest solution as a reference in the first step, the middle between two stochastically chosen solutions as the reference in the second step and neighborhood search in the third step. However, third step was less significant than the first and second steps which certainly lowers the accuracy on larger datasets.

2.3 Research gap

An appropriate selection of parameters for DL models is the main issue of exiting works. Some of the works utilized metaheuristic algorithms to select optimal parameters. However, fast convergence or low diversity in the populations are the main issues of the metaheuristic algorithms. This paper utilizes a new metaheuristic model ESCO which overcomes the limitations of other metaheuristic optimization models including COA.

3. Proposed work

This section illustrates the complete framework of the suggested ESCOTLViT model for plant disease detection. In this method, the weighted layers of pre-trained CNN models in TLMViT [16] are fine-tuned using the ESCO to reduce the complexity and improve the classification accuracy for plant diseases detection. The Fig. 1 depicts the procedure of the developed method. Notation list for variables are listed in Table 1.

Table 1. List of Notations

Notations	Description
A_x	Initial Population
$F(A)$	Objective Function
$\boldsymbol{c}$	Entity Member
R	Range between 0 to 1
R_1	First Stage Threshold
R_2	Second Stage Threshold
R_3	Third Stage Threshold
T	Iteration
I	Constant Arbitrary
A_x	Equivalent unit
A_z	Arbitrarily Chosen Unit
$A_{\mathcal{U}}$	Upper limit
$A_{\mathcal{L}}$	Lower limit
A_B	Global best Unit
a	Group of Entity
$A_{\mathcal{LL}}$	Local Lower Limit
$A_{\mathcal{LU}}$	Local Upper Limit
A_{GR}	Randomized entity inside search space

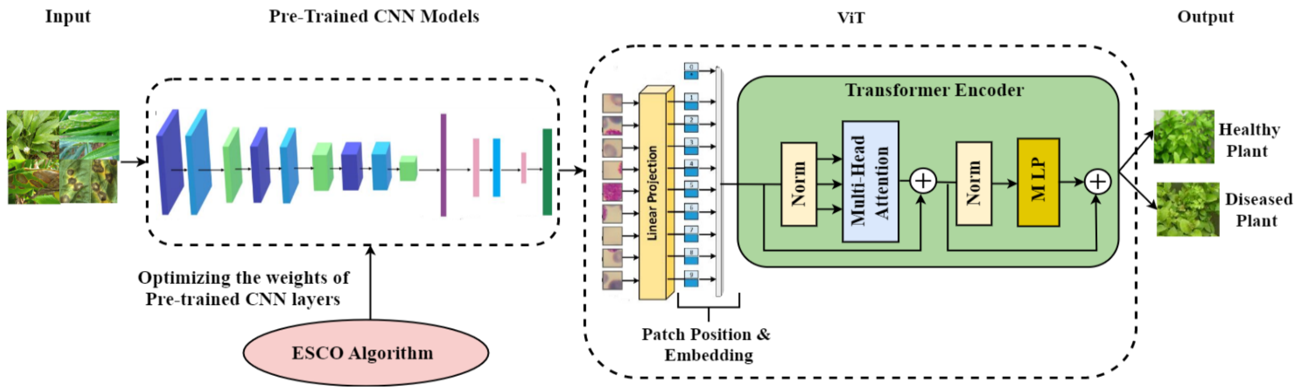

Figure. 1 Entire Procedure of the developed model

3.1 Fine-tuning pre-trained CNN weights using ESCO

The ESCO is employed in the structure of pre-trained models to optimize the weight layers for reducing the complexity in the detection of plant diseases.

The ESCO aims to improve the existing COA. The enhancement stems from two terms: extending and stochastic. First, ESCO broadens the plan outlined in the COA. Second, ESCO exhibits greater stochasticity than COA. ESCO lists three extensions. First, ESCO's COA contains three sequential phases rather than two. Second, ESCO conducts six searches rather than three, as specified in the COA. Third, ESCO employs three references in its guided searches: the global best unit (iguana on the tree), a randomized unit within the search space (iguana on the ground) and a randomly chosen unit.

ESCO handles role diversity differently than the COA. ESCO performs role segregation throughout all phases. It differs from the COA, which only implements role segregation in the first phase. ESCO uses a stochastic approach to role separation. It also differs from COA in that role segregation is a static process in which the first half of the population conducts a guided search for the optimal global unit. In comparison to the randomized unit, the second half of the population conducts directed searches.

The reasons for selecting this method are as follows. First, because many weak metaheuristics conduct repeated searches, additional searches may provide a better potential for improvement. Second, many faulty metaheuristics that aim to improve exploration capability use a randomly picked unit as a reference. Third, a stochastic approach to role segregation is chosen over a static split to prevent a monotone search, which is conducted in a static split in COA.

The ESCO is divided into three parts that occur in sequence. During the first and second phases, each unit conducts guided searches.

During the third phase, each unit does a random search. Each step contains two possible searches. In the first phase, each unit conducts a guided search for the global best unit or a randomized unit inside the search space. During the second phase, each unit conducts a directed search relative to a randomly selected unit. However, there are two alternatives in the second step. The guided search might begin with either the corresponding unit or a randomly picked unit. In the third phase, each unit conducts a neighborhood search. However, two alternatives can be used for the local search boundary. The options chosen in each step are stochastically determined depending on a threshold. If a randomly produced number drops below the threshold, the first choice is chosen. Otherwise, the second option is selected.

Each process generates a candidate. The proposed ESCO employs a tight acceptance-rejection process, with a candidate replacing the current unit only if they outperform the current unit. The last value of the global best unit is used to determine the final answer. Meanwhile, Eqs. (1) to (16) determines the detailed formalization of each process within the algorithm.

The initialization step involves two processes. The first step is to generate an initial unit randomly inside the search space, as specified in Eq. (1).

$$A_x = A_{\mathcal{L}} + R(A_{\mathcal{U}} - A_{\mathcal{L}}) \tag{1}$$

In COA, the objective fitness function (F) has each of its optimal location placed in the hyperparameters as number of neurons (f_1), number of hidden units (f_2), learning rate (f_3), weight decay (f_4), number of epochs (f_5), batch size (f_6), dropout rate (f_7), number of partitions (f_8), number of clusters per batch (f_8), momentum (f_9), optimizer (f_{10}), loss function (f_{11}). In COA, every data pattern A in Eq.

(2) in a data collection is considered a coati's location and the number of iguanas in these positions, A contains $F(A)$ in Eq. (3):

$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \\ A_{10} \\ A_{11} \end{bmatrix} \tag{2}$$

$$F(A) = \begin{bmatrix} f(A_1) \\ f(A_2) \\ f(A_3) \\ f(A_4) \\ f(A_5) \\ f(A_6) \\ f(A_7) \\ f(A_8) \\ f(A_9) \\ f(A_{10}) \\ f(A_{11}) \end{bmatrix} \tag{3}$$

The ideal solution defines the location of coati to search for most with iguanas. In COA, the fitness function value is used to assess the functionality of a candidate decision (coati's location) which is used in ESCO. The member of the population that assesses the finest outcome for the targeted operation is considered as the ideal member of the population and it is modified with every iteration.

Then, the second step is to update the global best unit from the following fitness operation F, as specified in Eq. (4).

$$A'_B = \begin{cases} A_x, F(A_x) < f(A_y) \\ A'_B, otherwise \end{cases} \tag{4}$$

The guided searches in the first phase are formalized using Eqs. (5) to (7). Eq. (5) states that a reference is created within the search space (GR). Eq. (6) formalizes the guided search for the global best unit. Eq. (7) formalizes the guided search with respect to the randomized unit in the search space. Meanwhile, Eq. (8) formalizes the updating procedure for the relevant unit.

$$A_{GR} = A_{\mathcal{L}} + R(A_{\mathcal{U}} - A_{\mathcal{L}}) \tag{5}$$

$$\mathcal{C} = \{A_x + R\,(A_B - 2A_x) \tag{6}$$

$$\mathcal{C} = \begin{cases} A_x + R(A_{GR} - 2A_x), F(A_{GR}) < F(A_x) \\ A_x + R(A_x - 2A_{GR}), otherwise \end{cases} \tag{7}$$

$$A'_x = \begin{cases} \mathcal{C}, F(\mathcal{C}) < F(A_x) \\ A_x, otherwise \end{cases} \tag{8}$$

The guided search in the second phase is formalized in this section. (9) formalizes the randomly selected unit from the population. Because the uniform random method is utilized, all units have an equal chance to choose. Eq. (10) formalizes the guided search for the appropriate unit in relation to the randomly picked unit. In contrast, Eq. (11) formalizes the directed search of the randomly picked unit relative to the matching unit.

$$A_z = IA \tag{9}$$

$$\mathcal{C} = \begin{cases} A_x + I(0,1).(A_z - 2A_x), F(A_z) < F(A_x) \\ A_x + I(0,1).(A_x - 2A_z), otherwise \end{cases} \tag{10}$$

$$\mathcal{C} = \begin{cases} A_z + I(0,1).\ (A_z - 2A_x), f(A_z) < f(A_x) \\ A_z + I(0,1).(A_x - 2A_z), otherwise \end{cases} \tag{11}$$

The random search in the third phase is represented by Eq. (12) to (14). As part of the neighborhood search, a candidate is arbitrary produced at the appropriate unit. Eq. (12) validates the local lower boundary calculation and Eq. (13) formalizes the local upper boundary computation. Eq. (14) depicts the random search by using both local bounds.

$$A_{\mathcal{LL}} = \begin{cases} \frac{A_{\mathcal{L}}}{T}, I(0,1) < R_3 \\ A_{\mathcal{L}}\left(1 - \frac{T}{T_{max}}\right), otherwise \end{cases} \tag{12}$$

$$A_{\mathcal{LU}} = \begin{cases} \frac{A_{\mathcal{U}}}{T}, I(0,1) < R_3 \\ A_{\mathcal{U}}\left(1 - \frac{T}{T_{max}}\right), otherwise \end{cases} \tag{13}$$

$$\mathcal{C} = A_x + \left(1 - 2I(0,1)\right).I(A_{\mathcal{LL}}, A_{\mathcal{LU}}) \tag{14}$$

Once the performance is executed, the optimal decision obtained from all algorithm iterations is derived as the final outcome. The optimum fitness operation yields the finest potential outcome, with the fitness operation with the lowest value being the ideal answer. The processes outlined above are continued

Received: April 6, 2024. Revised: April 30, 2024.

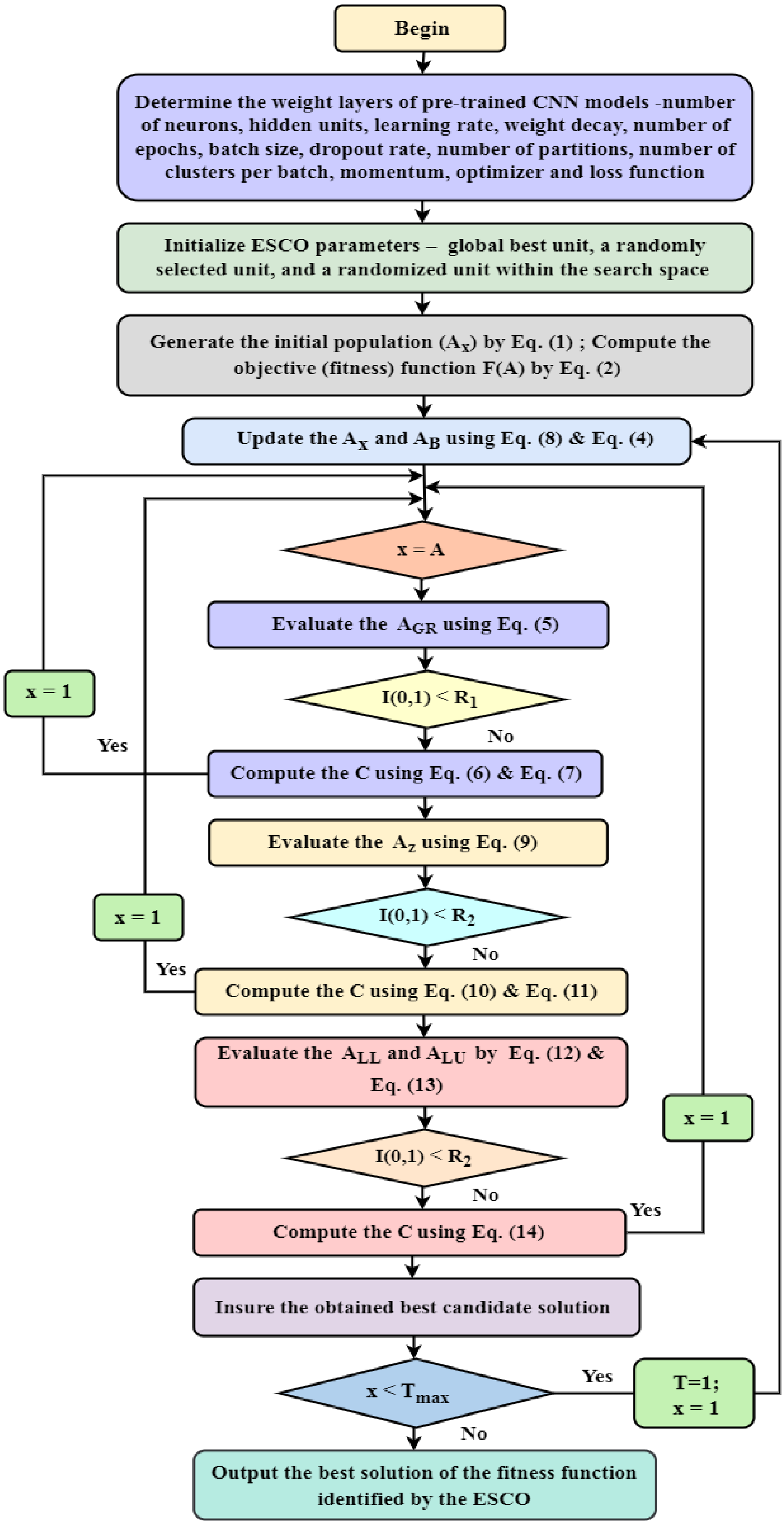

Figure. 2 Flowchart of ESCO for weight layer selection of pre-trained CNN models

until the ideal solution is reached.

This approach helps to improve the weighted layers of a pre-trained CNN models to lessen the complexity for effective plant diseases. The pseudocode of ESCO for optimizing the weight layers of pre-trained CNN models is described in Algorithm 1. Also, an overall workflow of the e ESCO for fine-tuning the weight layers is shown in Fig. 2.

Algorithm 1: Optimizing the weight layers using ESCO

Input: Set of hyperparameters for pre-trained CNN models (i.e., number of neurons, number of hidden units, learning rate, weight decay, number of epochs, batch size, dropout rate, number of partitions, number of clusters per batch, momentum, optimizer and loss function).
Output: Optimal weight layers
1. **Start**
2. **for** $x = 1: N(A)$
3. Initialize A_x using Eq. (1)
4. Determine the objective function $F(A)$ using Eq. (2) and Eq. (3)
5. Identify A_{best} using Eq. (4)
6. **for** $T = 1: T_{max}$
7. **for** $x = 1: N(A)$
// Stage 1
9. Obtain A_{GR} by Eq. (5)
10. **If** $I(0,1) < R_1$ **then**
11. Produce $\mathcal{C}$ as per Eq. (6)
12. **else**
13. Produce $\mathcal{C}$ as per Eq. (7)
14. Update A_x and A_B using Eq. (8) & Eq. (4)
// Stage 2
15. Determine A_z by Eq. (9)
16. **If** $I(0,1) < R_2$ **then**
17. Produce $\mathcal{C}$ as per Eq. (10)
18. **else**
19. Produce $\mathcal{C}$ as per Eq. (11)
20. Update A_x and A_B using Eq. (8) & Eq. (4)
// Stage 3
21. Evaluate $A_{\mathcal{LU}}$ and $A_{\mathcal{LL}}$ by Eq. (12) & Eq. (13)
22. Generate $\mathcal{C}$ as per Eq. (14)
23. Update A_x and A_B using Eq. (8) & Eq. (4)
24. **end for**
25. Save the optimal candidate solution identified so far
26. **end for**
27. Determine the feasible solution by ESCO (Fine-tuning the weights of pre-trained CNN layers)
28. **end for**

---

4. Experimental results

4.1 Dataset description

PlantVillage Dataset [28]: The PlantVillage collection developed by the non-profit PlantVillage project, includes approximately 54,000 photos of 38 crop types, including cassava, tomato, pepper and potato. For experimental purposes, 22787 images are linked to 15 plant diseases, including pepper-bell bacterial spot, pepper-bell healthy, potato early blight, potato late blight, tomato bacterial spot, tomato early blight, tomato late blight, tomato leaf mold, tomato 2-spotted spider mites, tomato target spot, tomato yellow leaf curl virus, tomato mosaic virus and tomato health.

PlantDoc-Dataset [29]: This dataset comprises 2,598 data points from 13 plant species and up to 17 disease classifications, representing around 300 person hours of labor in annotating internet scraped images. After processing, 2564 images were collected from 17 classes for testing purposes.

DiaMOS Dataset [30]: This dataset constitutes of 3505 images of fruit and leaves affected by four severity level of diseases like leaf spot, leaf curl, slug damage and healthy leaf. By leaving the fruit diseases images, 3006 plant diseases images are adopted for the experimental task.

4.2 Experimental setup and performance metrics

The implementation of both proposed and existing models are carried out in Python 3.11 and executed on a system with an Intel® CoreTM i5-4210 CPU @ 3GHz, RAM and a 1TB HDD running on Windows 10 64-bit, evaluated with the Plant Village, Plant-Doc and DiaMOS datasets (given in section 4.1). Similarly, Table 2 depicts the parameter values utilized for simulating both existing and proposed model to measure performance.

All three datasets are divided into 80% for training and 20% for testing. In this section, efficiency of the proposed ESCOTLViT model is evaluated and compared with existing models like CNN-ViT [17], OMNCNN [18], CBAM [20], DeepplantNet [21], AM-MNV2 [23], IQWO-PCA-DNN [24] and TLMViT [12] on PlantVillage, PlantDoc and DiaMOS datasets. The model's efficiency in predicting PPD is measured using various performance metrics, which are provided below.

Accuracy: It is the fraction of proper partition and categorization of diseased plant samples (plant images) over the total samples tested.

Table 2. List of optimal hyperparameters for both proposed and existing models

Parameters	Search Space	Optimal Range
Pre-Trained Models(AlexNet, Res-Net 50, VGG-16, VGG-19 and Inception-V3)		
Number of Neurons	[32, 64, 96, 128, 160, 192]	128
Number of hidden units	[64, 128, 256, 512]	256
Learning rate	[0.01, 0.001, 0.0001]	0.001
Weight decay	[0.0002, 0.0004, 0.0005, 0.0006]	0.0005
Number of epochs	[25, 50, 75, 100]	75
Batch size	[32, 64, 128, 512]	64
Dropout rate	[0.1, 0.2, 0.3, 0.5]	0.5
Number of partitions	[50, 100, 150, 200]	100
Number of clusters per batch	[3, 4, 5, 6]	5
Momentum	[0, 1]	0.7
Optimizer	[Stochastic gradient descent, Adam]	Adam
Loss Function	[Cross-entropy, Mean Squared Error (MSE)]	MSE
ViT		
Patch Size	[2, 4, 6, 8]	4
Projection Dimension	[32, 64, 96, 128]	96
Number of heads	[2, 3, 4, 5]	4
Transformer Layers	[4, 8, 12, 16]	12
MLP		
Number of neurons in initial hidden layers of MLP	[256, 512, 768, 1024]	1024
Number of neurons in second hidden layers of MLP	[128, 256, 384, 512]	512
ESCO		
Test suite Dimensions	[10, 30, 50, 100]	100
Maximum Number of Iterations	[20. 40, 60, 80, 100]	80

$$Accuracy = \frac{True\ Positive\ (TP)+True\ Negative\ (TN)}{TP+TN+False\ Positive\ (FP)+False\ Negative\ (FN)} \quad (11)$$

In Eq. (11), TP represents the amount of healthy plant samples appropriately categorized as healthy, whereas TN is the proportion of abnormal plant samples accurately categorized as corresponding diseased classes. Similarly, FP is the percentage of diseased plant samples categorized as healthy and FN is the ratio of normal plant samples categorized as abnormal.

Precision: Out of all the positive predictions produced by the model, it estimates the proportion of real positive predictions, which are samples of plant diseases that are successfully anticipated in Eq. (12).

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

Recall: The ratio of accurate projections to total positive occurrences in the dataset is calculated.

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

F1-score: It represents the partials medians of precision and recall.it is represented in Eq. (14):

$$F1 - score = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (14)$$

Fig. 3 portrays the performance of ESCOTLViT model against different existing models which is tested on PlantVillage dataset. It is noticed that the accuracy of ESCOTLViT is 94.22% which is 25.14%, 19.59%, 17.76%, 14.39%, 11.09%, 7.53% and 2.34% greater than other existing models respectively. The precision of ESCOTLViT is 90.95% improved by 21.06%, 18.64%, 15.08%, 11.98%, 9.43%, 6.64% and 3.67% than other models accordingly. The recall of ESCOTLViT is 94.38% enhanced by 25.44%, 19.18%, 16.3%, 14.49%, 10.89%, 9.01% and 2.18% in contrasted to existing models respectively. Finally, F1-Score of ESCOTLViT is 92.47% increased to 23.19%, 19.61%, 15.76%, 13.64%, 10.46%, 7.59% and 3.09% than other model respectively. From this analysis, it is observed that the proposed ESCOTLViT provides best performances than other existing models on Plant Village dataset for plant diseases detection.

Fig. 4 portrays the performance of ESCOTLViT model against various existing models on PlantDoc dataset. It is noticed that the accuracy of ESCOTLViT is 94.81% increased up to 23.74%, 20.86%, 16.40%, 13.26%, 10.81%, 8.14% and 2.13% than other existing models respectively. The precision of ESCOTLViT is 94.73% improved by 26.10%, 21.35%, 15.51%, 12.11%, 9.79%, 6.99% and 2.36% compared to the other algorithms, accordingly. The recall of ESCOTLViT is 94.93% enhanced by 23.71%, 19.89%, 14.74%, 13.43%, 8.65%, 5.70% and 2.09% in contrasted to other plant diseases respectively. Finally, the F1-Score of

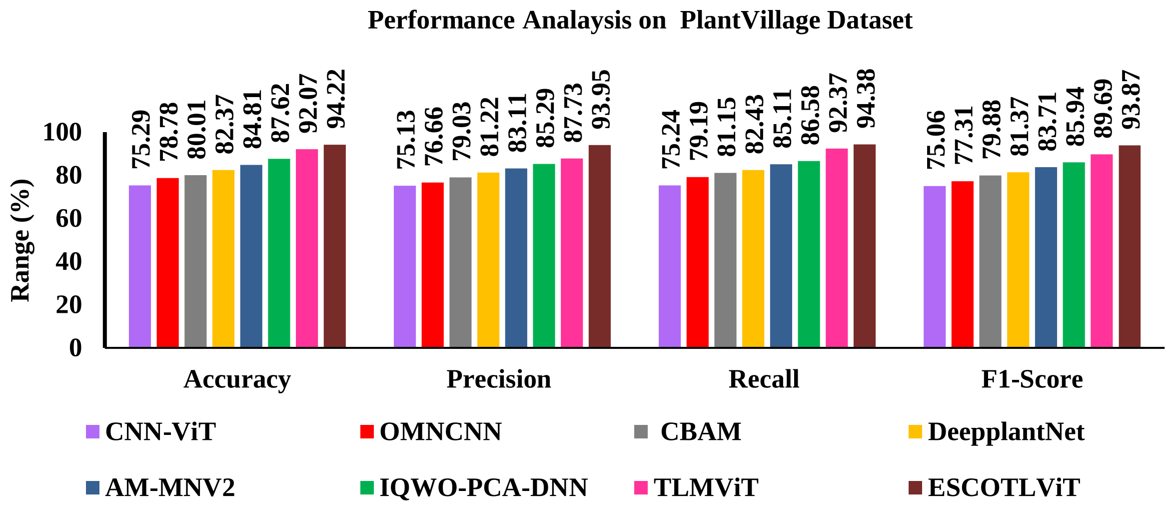

Figure. 3 Performance of proposed and classical DL models for Plant Village dataset

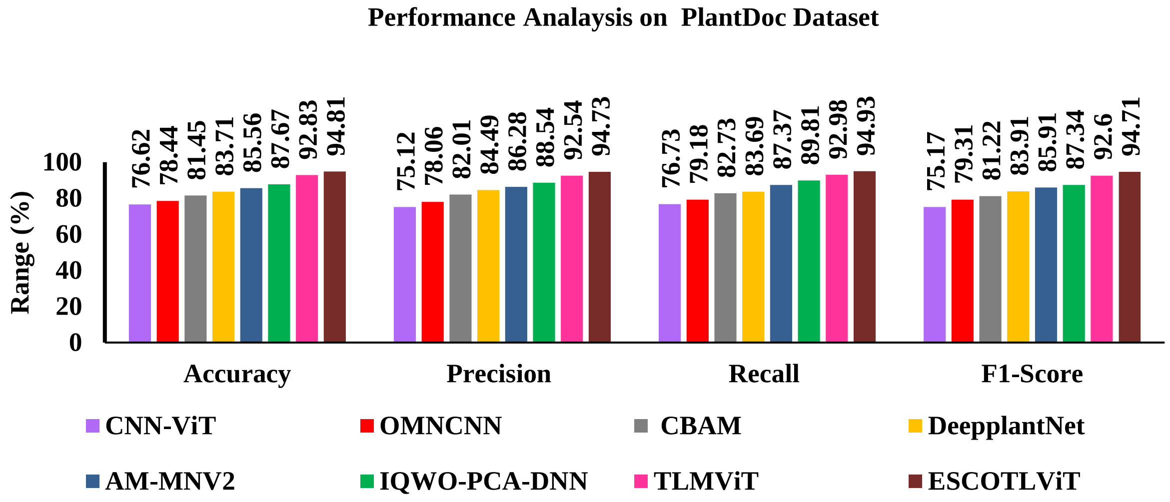

Figure. 4 Performance of proposed and classical DL models for Plant Doc dataset

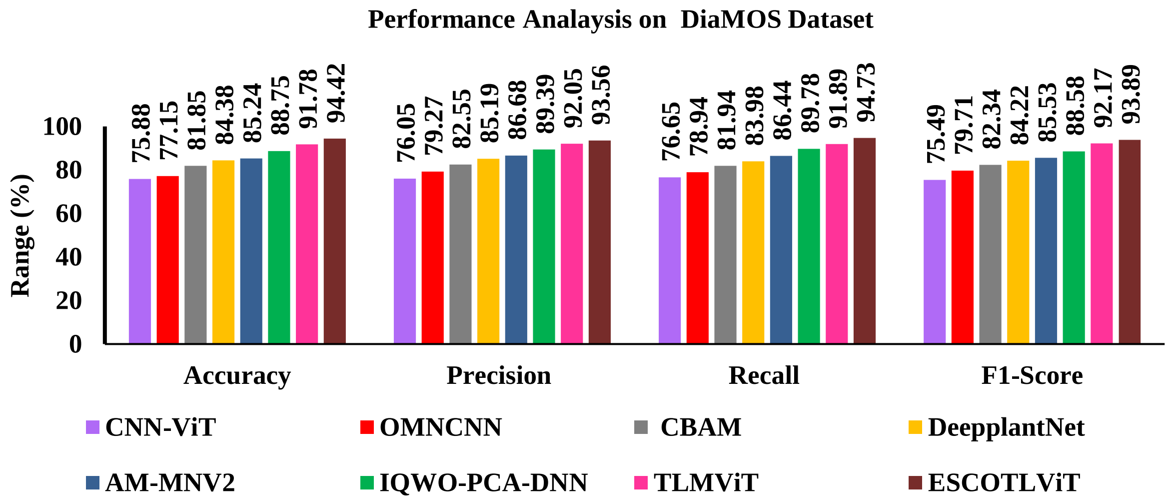

Figure. 5 Performance of proposed and classical DL models for DiaMOS dataset

ESCOTLViT is 94.71% which is increased to 25.99%, 19.41%, 16.60%, 12.87%, 10.24%, 8.43% and 2.27% than other detection models respectively. From this analysis, it is observed that the proposed ESCOTLViT provides best performances than other existing models on Plant Doc dataset for plant diseases detection.

Fig. 5 portrays the performance of ESCOTLViT model against various existing models on DiaMOS Dataset. It is noticed that the accuracy of ESCOTLViT is 94.42 % increased up to 24.43 %, 22.38%, 15.36%, 11.90%, 10.77%, 6.39% and 2.88% than other existing models respectively. The precision of ESCOTLViT is 93.56% improved by 23.02%, 18.02%, 13.34%, 9.83%, 7.94%, 4.66% and 1.64 %compared to the other algorithms, accordingly. The recall of ESCOTLViT is 94.73% enhanced by 23.59%, 18.50%, 15.61%, 12.80%, 9.59%, 5.51% and 3.09% in contrasted to other plant diseases respectively. Finally, the F1-Score of ESCOTLViT is 93.89% which is increased to 24.37%, 17.79%, 14.03%, 11.48%, 9.77%, 5.99% and 1.875 than other detection models respectively. From this analysis, it is observed that the proposed ESCOTLViT provides best performances than other existing models on DiaMOS dataset for plant diseases detection.

In the literature, CNN-ViT [17], DeepplantNet [21], AM-MNV2 [23], IQWO-PCA-DNN [24] and TLMViT [12] models have utilized PlantVillage dataset. Similarly, OMNCNN [18] and CBAM [20] utilized PlantDoc and DiaMOS dataset respectively. Hence, this work evaluates proposed and existing models on PlantVillage, PlantDoc and DiaMOS dataset using the parameters as per Table 2. From the above comparison, it is proved that the proposed ESCOTLViT model obtains efficient results on all three datasets for the plant diseases prediction. This is because the weight layers of pre-trained CNN models are optimized using ESCO which results enhanced accuracy results for plant diseases detection and classification.

5. Conclusion

In this paper, ESCOTLViT model is developed to enhance the classification accuracy and lowers the complexity for efficient plant disease detection. The adopted ESCO optimizes the weights layers of pre-trained CNN models by considering factors like number of neurons, hidden units, learning rate, weight decay, epochs, batch size, dropout rate, partitions, clusters per batch, momentum, optimizer and loss function. ESCO divides the population into two fixed groups and employs three sequential phases. ESCO uses a fitness function to evaluate individuals and select the best weight layers for optimal results. At last, the ESCOTLViT model outperforms with other existing models CNN-ViT, OMNCNN, CBAM, DeepplantNet, AM-MNV2, IQWO-PCA-DNN and TLMViT with accuracy of 94.22%, 94.81% and 94.42% on PlantVillage, PlantDoc and DiaMOS Dataset respectively.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, methodology, software, validation, Indumathi; formal analysis, investigation, Kumuthaveni; resources, data curation, writing—original draft preparation, Indumathi; writing—review and editing, Indumathi; visualization, supervision, Kumuthaveni;

References

[1] L. Praburaj, F. Design and T. Nadu, “Role of agriculture in the economic development of a country”, *Shanlax International Journal of Commerce*, Vol. 6, No. 3, pp. 1-5, 2018.

[2] S. K. Goyal, J. P. Rai and S. R. Singh, “Indian agriculture and farmers–problems and reforms”, *Indian Agriculture and Farmers*, Vol. 246, No. 2019, pp. 79-87, 2016.

[3] A. A. Gomaa and Y. M. Abd El-Latif, “Early prediction of plant diseases using CNN and GANs”, *International Journal of Advanced Computer Science and Applications*, Vol. 12, No. 5, 2021.

[4] K. Raymaekers, L. Ponet, D. Holtappels, B. Berckmans and B. P. Cammue, “Screening for novel biocontrol agents applicable in plant disease management–a review”, *Biological Control*, Vol. 144, No. 104240, 2020.

[5] M. Jeger, R. Beresford, C. Bock, N. Brown, A. Fox, A. Newton and J. Yuen, “Global challenges facing plant pathology: multidisciplinary approaches to meet the food security and environmental challenges in the mid-twenty-first century”, *CABI Agriculture and Bioscience*, Vol. 2, No. 1, pp. 1-18, 2021.

[6] A. Sinha and R. S. Shekhawat, “Review of image processing approaches for detecting plant diseases”, *IET Image Processing*, Vol. 14, No. 8, pp. 1427-1439, 2020.

[7] A. H. Kulkarni and A. Patil, “Applying image processing technique to detect plant diseases”, *International Journal of Modern Engineering Research*, Vol. 2, No. 5, 3661-3664, 2012.

[8] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images", *Biosystems Engineering*, Vol. 144, pp. 52-60, 2016.
[9] G. S. Sujawat, "Application of Artificial Intelligence in detection of diseases in plants: A Survey", *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, Vol. 12, No. 3, pp. 3301-3305, 2021.
[10] M. Nagaraju and P. Chawla, "Systematic review of deep learning techniques in plant disease detection", *International Journal of System Assurance Engineering and Management*, Vol. 11, pp. 547-560, 2020.
[11] M. Chohan, A. Khan, R. Chohan, S. H. Katpar and M. S. Mahar, "Plant disease detection using deep learning", *International Journal of Recent Technology and Engineering*, Vol. 9, No. 1, pp. 909-914, 2020.
[12] 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*, 2023.
[13] P. D. Kusuma and A. Dinimaharawati, "Extended Stochastic Coati Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 3, pp. 482-494, 2023, doi: 10.22266/ijies2023.0630.38.
[14] M. Dehghani, Z. Montazeri, E. Trojovska, and P. Trojovsky, "Coati Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm for Solving Optimization Problems", *Knowledge-Based Systems*, Vol. 259, pp. 1-43, 2023.
[15] B. Natesan, A. Singaravelan, J. L. Hsu, Y. H. Lin, B. Lei and C. M. Liu, "Channel–spatial segmentation network for classifying leaf diseases", *Agriculture*, Vol. 12, No. 11, pp. 1-20, 2022.
[16] V. Devisurya, R. Devi Priya and N. Anitha, "Early detection of major diseases in turmeric plant using improved deep learning algorithm", *Bulletin of the Polish Academy of Sciences. Technical Sciences*, Vol. 70, No. 2, 2022.
[17] Y. Borhani, J. Khoramdel and E. Najafi, "A deep learning based approach for automated plant disease classification using vision transformer", *Scientific Reports*, Vol. 12, No. 1, pp. 1155, 2022.
[18] 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.
[19] M. H. Saad and A. E. Salman, "A plant disease classification using one-shot learning technique with field images", *Multimedia Tools and Applications*, pp. 1-26, 2023.
[20] P. Alirezazadeh, M. Schirrmann and F. Stolzenburg, "Improving Deep Learning-based Plant Disease Classification with Attention Mechanism", *Gesunde Pflanzen*, Vol. 75, No. 1, pp. 49-59, 2023.
[21] N. Ullah, J. A. Khan, S. Almakdi, M. S. Alshehri, M. Al Qathrady, N. El-Rashidy 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, No. 1, pp. 1-16, 2023.
[22] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, "Multi-Class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern", *IEEE Access*, Vol. 11, No. 1, pp. 62307- 62317, 2023.
[23] H. Wang, S. Qiu, H. Ye and X. Liao, "A Plant Disease Classification Algorithm Based on Attention MobileNet V2", *Algorithms*, Vol. 16, No. 9, pp. 1-19, 2023.
[24] M. Sowmiya and S. Krishnaveni, "IoT enabled prediction of agriculture's plant disease using improved ϖ quantum whale optimization DRDNN approach", *Measurement: Sensors*, Vol. 27, No. 1, pp. 1-9, 2023.
[25] P. D. Kusuma and A. Dinimaharawati, "Swarm Bipolar Algorithm: A Metaheuristic Based on Polarization of Two Equal Size Sub Swarms", *International Journal of Intelligent Engineering & Systems*, Vol. 17, No. 2, pp. 377-389, 2024, doi: 10.22266/ijies2024.0430.31.
[26] P. D. Kusuma and M. Kallista, "Swarm Space Hopping Algorithm: A Swarm-based Stochastic Optimizer Enriched with Half Space Hopping Search", *International Journal of Intelligent Engineering & Systems*, Vol. 17, No. 2, pp. 670-682, 2024, doi: 10.22266/ijies2024.0430.54.
[27] P. D. Kusuma and M. Kallista, "Migration-Crossover Algorithm: A Swarm-based Metaheuristic Enriched with Crossover Technique and Unbalanced Neighbourhood Search", *International Journal of Intelligent Engineering & Systems*, Vol. 17, No. 1, pp. 698-710, 2024, doi: 10.22266/ijies2024.0229.59.
[28] https://www.kaggle.com/emmarex/plantdisease
[29] https://github.com/pratikkayal/PlantDoc-Dataset

Received: April 6, 2024. Revised: April 30, 2024.

[30] G. Fenu and F. M. Malloci, "DiaMOS plant: A dataset for diagnosis and monitoring plant disease", *Agronomy*, Vol. 11, No. 11, pp. 1-12, 2021.

 DOI: 10.22266/ijies2024.0831.21