Cucurbit Leaf Disease Classification using Compact Vision Transformer and Cubic Support Vector Machine with Metaheuristic Optimization

R. Monisha[a*], K. S. Tamilselvan[b], L. Karthiba[c]

Corresponding author – R. Monisha

[a,b]Department of Biomedical Engineering,

KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India – 641 407.

[a]e-mail: 21ftec004@kpriet.ac.in

[b]e-mail: tamilselvan@kpriet.ac.in ;

[c]Department of Plant Pathology,

Tamil Nadu Agricultural University, Coimbatore, Tamil Nadu, India – 641 003.

e-mail: karthiba.l@tnau.ac.in ;

Abstract

Cucurbits, a widely cultivated crop from the *Cucurbitaceae* family, account for about 6% of global vegetable production. Their yield, however, is susceptible to disease and pest attacks, contributing to food insecurity. Timely detection of these diseases is crucial to prune agricultural yield losses. This paper proposes a hybrid compact vision transformer (CVT) and cubic support vector model (CSVM) that employs Harris Hawks Optimization (HHO) for cucurbit leaf disease classification. Initially, image preprocessing is carried out using histogram equalization and GrabCut algorithm to effectively handle images with dispersed backgrounds. The preprocessed output is fed into the proposed hybrid model, where the CVT hyperparameters are optimized using the HHO algorithm. Finally, the features are extracted through CVT and classified using CSVM. The efficiency of the proposed model is highlighted by comparing it to various standard machine learning classifiers. Experimental study with the cucurbit dataset reveals that the proposed method attained 96.96% accuracy, contributing to sustainable crop production.

Keywords: Compact vision transformers, Cucurbit disease detection, Deep learning, Harris Hawks optimization, Hyperparameters, Machine learning.

1. Introduction

Cucurbits, from the *Cucurbitaceae* family, are a diverse group of vegetables encompassing approximately 90 genera and 950 species including pumpkins, gourds, etc. [1]. However, their growth is impeded by various diseases that affect the overall crop yield. Fungal diseases and viral diseases cause rapid defoliation and fruit deformities [2-3]. Hence, to prevent disease outbreaks and avoid major yield losses, early identification of cucurbit leaf disease is essential.

Since the 1990s, advanced image processing technologies have been utilized for providing accurate and realistic identification of plant diseases [4]. Nowadays, image processing techniques are utilized with machine learning (ML) approaches for capturing and classifying the plant diseases. ML models classify plant diseases based on features such as shape, color, statistics, and texture that are manually extracted from the leaf images [5-8]. Support Vector Machines (SVM), k-nearest neighbors, and Random Forests (RF) are the most commonly applied traditional ML approaches for plant disease classification [6]. However, the performance of ML models depends heavily on the quality of the handcrafted features from the input data [5]. But deep learning (DL) models overcome this issue by automatically learning features from plant images, eliminating the need for manual feature extraction [9,10].

DL techniques, particularly convolutional neural networks (CNNs) and vision transformers (ViTs), have been applied to image-related plant disease detection [11-15]. CNNs extract spatial and hierarchical features from leaf images using convolutional layers and then classify infected and non-infected leaves [16]. However, they focus less on global relationships across images. In contrast, ViTs divide the leaf images into small patches, convert them into tokens, and process these tokens using transformer encoders [17]. They specialize in capturing global relationships to understand the scattered patterns across different regions of the leaf throughout the entire image(s) using a self-attention mechanism. But they generally require large datasets to reach peak performance.

In contrast, Hassani et al. [18] proposed a compact vision transformer (CVT) for image classification tasks, which captures global information, is suitable for small datasets and edge-deployment. The CVT replaces the class token in the ViT with a sequence pooling method. This method aggregates information from all the tokens after or during training, thereby reducing the computations performed by the class token. Hence, they have smaller parameters and reduced overfitting. A lightweight CVT that runs on edge devices for plant disease detection in agricultural field, enables real-time diagnosis for farmers.

The performance of CVT depends on its choice of hyperparameters, such as attention heads, number of encoders, patch size, etc. [18]. Manual hyperparameter tuning, though common, is tedious. Recently, many studies have employed optimization algorithms for hyperparameter selection

in an ML or DL architectures [19-25]. Among these approaches, the Harris Hawks Optimization algorithm (HHO) [26] has emerged as a promising method. It follows the hunting method used by Harris Hawks to explore the multidimensional hyperparameter space and excels in balancing exploration and exploitation for avoiding local minima.

Several research gaps exist in plant disease classification using ML or DL models. Although CNN- and ViT-based approaches have reported high accuracies, they are either computationally expensive or require large datasets. then the exploration of different crops and disease coverage. Additionally, there is a lack of systematic hyperparameter optimization.

In this paper, a lightweight CVT is introduced to classify the diseases in cucurbit plants. The dataset undergoes preprocessing using CLAHE and Grabcut; the features are extracted by CVT and then classified using a cubic support vector machine (CSVM). The CSVM classifier allows the extracted features to be mapped with clearer decision boundaries. The proposed model achieved 96.96% accuracy with 0.257 million training parameters.

The remainder of this paper is organized as follows. The proposed scheme, including the materials and methods, is given in Section 2. The results of the performance validation of the proposed plan, along with the future scope, are presented in Section 3. Finally, this study is concluded in Section 4.

2. Proposed methodology

The proposed work is presented in this section. The proposed work consists of four steps, as depicted in Fig. 1. The dataset is subjected to augmentation and resizing. It is then preprocessed using the CLAHE and GrabCut. The features are extracted using a CVT, whose hyperparameters are optimized via HHO. The extracted features are classified using 10 ML classifiers, among which CSVM achieved the highest performance. Thus, we propose a CVT-HHO-CSVM hybrid combination that identifies cucurbit leaf diseases, which makes it ideal for other image classification and agriculture-related tasks.

2.1 Dataset

The images were collected in agricultural fields near Coimbatore, Tamil Nadu, India. They were taken using a One Plus Nord CE 2 Lite smartphone with 64-megapixels. Fig. 2 shows examples of the dataset. Multiple cucurbit species were covered, including bottle gourd, sweet pumpkin, and ridge gourd. All photographs were stored in the JPEG format. In addition to the images from the local field, images were incorporated from the Sweet Pumpkin Disease Recognition Dataset [27].

Various techniques such as random rotation, crop, and flip were applied to the original images with a random seed of 42. The input images were resized to a size of 128*128. The final dataset is

split into training (70%), validating (20%), and testing (10%) images. Table 1 provides the class distribution of both datasets.

2.2 Preprocessing

2.2.1 CLAHE

The dataset undergoes preprocessing using the Contrast Limited Adaptive Histogram Equalization (CLAHE) approach [28]. It enhances the contrast of an image while simultaneously reducing the noise. It is employed due to its efficiency in preserving edges while mitigating information loss.

2.2.2 GrabCut

The CLAHE output is fed into a segmentation algorithm: the GrabCut algorithm [29]. Segmentation of a color image is straightforward when all objects have distinct colors and boundaries, but it becomes a complex problem when the images in the dataset have diverse backgrounds. This makes it difficult for the classifier model to extract the features. GrabCut is applied to address such issues by segmenting the foreground from the background with graph cut and iterative energy minimization.

2.3 Disease classification using the proposed CVT-HHO-CSVM

2.3.1 Structure of the CVT

The proposed methodology is based on the CVT presented by Hassani et al. [18], a DL model designed for image processing applications. The architecture has five main parts: patch creation and embedding, positional encoding, transformer encoder, sequence pooling, and classification head, as illustrated in Fig. 3.

Patch creation and embedding: First, the input is split into a sequence of patches such that it becomes more accessible for the transformer to process the data. If the input has a dimension of $H \times W \times C$, where H is the height; W is the width; the dimension of patch is $P_h \times P_w$, then the number of non-overlapping patches generated is given by Eq. (1).

$$N = \frac{H \times W}{P_h \times P_w} \quad \text{—(1)}$$

The non overlapping patches are then flattened $\left(X_{flatten\_i}\right)$ into an 1D vector and is projected to an embedding dimension as described in Eq. (2).

$$X_{pd\_i} = W_l X_{flatten\_i} + b_l \quad \text{—(2)}$$

where each X_{pd_i} represents a linearly embedded vector for the i^{th} patch, W_l is the learnable weight matrix, and b_l is the learnable bias vector.

Positional encoding: Transformers lack the ability to gauge the position of split tokens that are supplied to them because they are not sequential. In this study, trainable positional encodings (PE_i) are added to the input embeddings to encode the position of each token individually as given in Eq. (3).

$$X_{input\_i} = X_{pd\_i} + PE_i \quad \text{—(3)}$$

where X_{input_i} is the positional encoded token.

Encoder block: The positionally encoded token X_{input_i} is fed to the encoder of the transformer. Every encoder is made up of four main components namely: MHSA (multi-head self-attention), MLP (multi-layer perceptron), layer normalization, and residual connections, as shown in Fig. 3.

The encoder is based on the self-attention method, where the model dynamically weighs the significance of each token relative to every other token, allowing it to prioritize key patches. Each token in a self-attention mechanism is represented by one of three vectors: the key K_i, query Q_i, and value V_i [17].

$$Q_i = X_{input\_i} W_i^Q; K_i = X_{input\_i} W_i^K; V_i = X_{input\_i} W_i^V \quad \text{—(4)}$$

The attention scores are determined by calculating the scaled dot product of the query and key vectors as given in Eqn (4). These scores are then scaled by the square root of the dimension of the key vectors and processed through a softmax function to generate a weight matrix. These weights determine the contribution of each token to the self-attention mechanism output. The output of self-attention layer A_i for i^{th} token is given in Eq. (5) [17].

$$A_i(Q,K,V) = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \quad \text{—(5)}$$

where d_k is dimensionality of key vector.

This enables each token to incorporate information from the other tokens, resulting in a global context. A single attention head in the MHSA block produces an output given by Eq. (6).

$$SAHead_i = A_i(Q,K,V) \quad \text{—(6)}$$

Following the computation of $SAHead_i$ values, they are concatenated across the feature dimension as given in Eq. (7).

$$MHSA\left(X_{input\_i}\right) = Concat\left(SAHead_1, SAHead_2, ..., SAHead_n\right) W^o \quad \text{—(7)}$$

where n represents the number of attention heads in a single MHSA layer, and where W^o is the learnable projection matrix that projects the concatenated output to a desired dimension. The output of the MHSA is stored in the X_{input_i} as given in Eq. (8).

$$X_{output\_i} = MHSA\left(X_{input\_i}\right) \quad —(8)$$

The MLP block consists of a simple feed-forward network with a GeLU activation function, as shown in Eq. (9-10).

$$H = GeLU\left(X_{output\_i} W_1 + b_1\right) \quad —(9)$$

$$X_{MLP} = HW_2 + b_2 \quad —(10)$$

Where W_i and b_i are weights and biases respectively. This architecture utilizes layer normalization before every sublayer (MHSA/MLP) and residual connections after every sublayer.

Sequence pooling: Sequence pooling aggregates the information from all the token embeddings via attention weights to generate a single vector that represents the entire sequence as given by Eq. (11) [18].

$$Z = f\left(X_{MLP}\right) \in R^{b \times n \times d} \quad —(11)$$

where Z is the output of the transformer encoder, b refers to batch size, n refers to sequence length, and d is the embedding dimension. A linear layer g is applied to this Z to attain attention weights as shown in Eq. (12).

$$Attention\_weights = \text{Softmax}\left(g\left(Z\right)^T\right) \in R^{b \times 1 \times n} \quad —(12)$$

The weighted sum of token embeddings is then computed as given in Eq. (13). The output $Z \in R^{b \times d}$ is subsequently produced, which is then fed into the classifier.

$$Z_{pool} = Attention\_weights \times Z \in R^{b \times d} \quad —(13)$$

Classification head: This takes the pooled outputs from the sequence pooling layer and applies a linear transformation to produce classification probabilities using the softmax activation.

2.3.2 Optimization of the CVT using HHO

HHO is a swarm-based metaheuristic optimization algorithm proposed by Heidari et al. [21]. This algorithm is founded on the Harris hawk's cooperative conduct and chasing modes in nature. In HHO, each hawk represents a potential solution and it works in three phases: exploration, energy estimation, and exploitation. The HHO algorithm searches for a near-optimal set of hyperparameters for optimizing the CVT. Each search agent (hawk) in the algorithm represents a set of hyperparameters (potential solutions). During each iteration, the model is instantiated using the

hyperparameters defined by a particle, trained on the training data, and evaluated on the validation set. The validation accuracy is a fitness score to tell how close we are to the objective.

Initialization

The HHO begins by initializing the agent population N. Each agent is assigned a random position in a multidimensional search space. The maximum number of iterations T, dimensionality of the problem d, and the bounds for dimension as $[lb, ub]$ are also defined.

Let the hyperparameter search space be $\Re^d$, where d=4 represents the problem dimensionality (the total number of hyperparameters to be optimized). The hyperparameter vector for each agent i is denoted in Eq. (14).

$$x_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4}] \quad \text{—(14)}$$

where projection dimension $x_{i1} \in$ [16, 256], number of self-attention heads $x_{i2} \in$ [1,8], and number of transformer layers $x_{i3} \in$ [1,8], patch size $x_{i4} \in$ [4, 16]. The objective is to find the near-optimal hyperparameter vector x^* that maximizes validation accuracy given by Eq. (15).

$$x^* = \arg\min_{x \in X} f(x) \quad \text{—(15)}$$

Fitness function

A fitness function $f(x)$ evaluates the position of each agent after a move and calculates the fitness score. In this study, the set of hyperparameters that gives higher validation accuracy for the CVT model serves as the objective as depicted in Eq. (16).

$$f(x) = validation\_accuracy\left(CVT(x)\right) \quad \text{—(16)}$$

Escaping energy

HHO balances exploration and exploitation through adjustment of "escaping energy," enabling robust global optimization across complex search spaces. The energy of the target is then estimated; based on its value, the hawks either continue exploration or switch to exploitation. The adaptation from exploration to exploitation is modelled using the escaping energy of the target E as shown in Eq. (17).

$$E = 2E_0\left(1 - \frac{t}{T}\right) \quad \text{—(17)}$$

where $E_0 \in [-1,1]$ is the initial state of its energy and t is the current iteration.

Case 1: Exploration Phase ($|E| \geq 1$)

During exploration, the hawks either perch or search the multidimensional space randomly. When the escaping energy is large, the algorithm performs exploration. Initially, the agents explore random locations expecting to find their target depending upon a q value as shown in Eq. (18).

$$Y(t+1)=\begin{cases} Y_{rnd}(t)-r_1\left|Y_{rnd}(t)-2r_2Y(t)\right| & q\geq 0.5 \\ \left(Y_{target}(t)-Y_m(t)\right)-r_3\left(lb_j+r_4\left(ub_j-lb_j\right)\right) & q<0.5 \end{cases} \quad \text{—(18)}$$

where t is the current iteration, $Y(t+1)$ is the positional vector of agent in the upcoming iteration, Y_{target} is the target's spot, $Y(t)$ is the current spot of vector of agents r_{1to4} and q are random numbers in (0,1) and is updated every iteration. $Y_{rnd}(t)$ is randomly selected agent from the present population. $Y_m(t)$ average spot of the current population of agents and it is obtained using Eq. (19).

$$Y_m(t)=\frac{1}{N}\sum_{i-1}^{N}Y_i(t) \quad \text{—(19)}$$

Case 2: Exploitation Phase (|E| < 1)

During exploitation, the hawks use one of the four chasing and attacking strategies to catch the prey depending upon E : soft besiege, hard besiege, soft besiege with progressive dives, and hard besiege with progressive dives. The hawks dynamically switch between these modes for adaptive convergence.

i. Soft besiege

This step is slowly sneaking closer to the target and is given by Eq. (20-21).

$$Y(t+1)=\Delta y(t)-E\left|JY_{target}(t)-Y(t)\right| \quad \text{—(20)}$$

$$\Delta Y(t)=Y_{target}(t)-Y(t) \quad \text{—(21)}$$

where $\Delta Y(t)$ is the difference between the position of the target and current location in iteration t, r_5 is a random number between 0 and 1, $J=2\left(1-r_5\right)$ is the random moving strength of the target while escaping. The J value changes randomly in every iteration to simulate the nature of rabbit (target) motions.

ii. Hard besiege

This part here tells the process of jumping directly and quickly to catch the target. Here the current positions are updated using Eq. (22).

$$Y(t+1)=Y_{target}(t)-E\left|\Delta Y(t)\right| \quad \text{—(22)}$$

iii. Soft besiege with progressive dives

This is to sneak slowly but also to do sudden dives and jumps as shown in Eq. (23).

$$P = Y_{target}(t) - E \mid JY_{target}(t) - Y(t) \quad \text{—(23)}$$

The diving pattern observed here is in accordance with levy-flight patterns given by Eq. (24)

$$Q = P + S \times LF(d) \quad \text{—(24)}$$

where S is the random vector of size $1 \times d$ and LF is levy-flight function given by Eqn. (25).

$$LF(x) = 0.01 \times \frac{m \times \sigma}{|n|^{\frac{1}{\beta}}}, \sigma = \left(\frac{r(1+\beta) \times \sin(\frac{\pi\beta}{2})}{r(\frac{1+\beta}{2}) \times \beta \times 2(\frac{\beta-1}{2})} \right)^{\frac{1}{\beta}} \quad \text{—(25)}$$

where m and n are random values within (0,1) and β is a constant value of 1.5.

Finally, the position of agents in the soft besiege phase can be executed using Eq. (26).

$$Y(t+1) = \begin{cases} P & if\ F(P) < F(Y(t)) \\ Q & if\ F(Q) < F(Y(t)) \end{cases} \quad \text{—(26)}$$

iv. Hard besiege with progressive dives

This is to jump quickly and attack the target by doing sudden dives and small loops which is given in Eq. (27).

$$Y(t+1) = \begin{cases} P & if\ F(P) < F(Y(t)) \\ Q & if\ F(Q) < F(Y(t)) \end{cases} \quad \text{—(27)}$$

where P and Q are derived using new rules as in Eq. (28) and Eq. (29).

$$P = Y_{target}(t) - E \mid JY_{target}(t) - Y_m(t) \quad \text{—(28)}$$

$$Q = P + S \times LF(d) \quad \text{—(29)}$$

After max iterations T is completed, the global best position returns the set of hyperparameters that produced the highest validation accuracy for the CVT. The flowchart representing the HHO-based hyperparameter selection for CVT is depicted in Fig. 4.

2.3.3 Classification

To distinctly identify each disease of the cucurbits, classifiers are essential. The features extracted from the CVT were given to the CSVM classifier. To evaluate the performance of the classifier, it was compared with other classifiers namely: Decision Tree (DT), RF, SVM, Gaussian Naïve Bayes (GNB), Quadratic SVM (QSVM), Gradient Boosting Machine (GBM), Light GBM (LGBM), and CatBoost (CB).

3. Results and Discussion

Hereafter the proposed model refers to the hybrid of CVT-HHO-CSVM, where the final classification layer of CVT is replaced with CSVM classifier.

3.1 Experimental setup

The performance of the proposed work is evaluated in this section. It is implemented on a Windows 11 operating system possessing Intel core i5 CPU. The python code was simulated using Tensorflow 2.16.2 and Keras 3.4.1. HHO search ranges for each hyperparameter are: patch size (4,16), projection dimension (16, 256), number of transformer layers (1, 8), number of self-attention heads (1, 8). The Nadam optimizer, was chosen along with categorical cross-entropy loss. Table 2 shows hyperparameter values for the proposed model.

3.2 Performance metrics

The model was evaluated using the four most commonly used performance indices: accuracy, precision, recall, and f1-score as given in Eqs. (30-33). The confusion matrix is used to discern the performance of individual classes.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad —(30)$$

$$Precision = \frac{TP}{TP + FP} \quad —(31)$$

$$Recall = \frac{TP}{TP + FN} \quad —(32)$$

$$f1 - score = \frac{2 \times precision \times recall}{precision + recall} \quad —(33)$$

Here the terms True Negative (TN), False Negative (FN), False Positive (FP), and True Positive (TP) are used.

3.3 Experimental outcomes

3.3.1 Results of preprocessing

The dataset initially undergoes preprocessing steps as shown in Fig. 1. Fig. 5. shows the step-by-step procedure encountered by each image in the dataset during preprocessing.

3.3.2 Results of feature extraction

The segmented images are given to the CVT for feature extraction. The smallest CVT architecture proposed by Hassani et al. is considered [18], which has one encoder layer, a projection dimension of 16, and the Nadam optimizer. To demonstrate the potential of CVT for feature extraction, a comparative analysis was performed against existing DL models: LeNet and SqueezeNet.

The confusion matrices of three models are presented in Fig. 6. CVT achieved 90.41% accuracy with balanced performance across all classes. In class D, 26 out of 202 images were misclassified as other classes. In contrast, SqueezeNet exhibited better performance in predicting class D, but lowest with class C, while LeNet yielded the lowest results overall. The results reinforce that CVT maintains a consistent performance across diverse classes compared to other models.

The performance metrics of three models are presented in Fig. 7. CVT showcased the best performance with 90.41% accuracy, 91.7% precision, 91.5% recall, and 91.5% f1-score. The results signify the model's ability to identify true positives correctly with curtailing false positives and negatives across all the classes. SqueezeNet performed more similar to CVT, with a slightly lower accuracy of 88.9%. Its recall (90%) was marginally lower than CVT, along with a lower precision (90.5%) led to an overall f1-score of 89.7%. LeNet, lagged behind all the models. It achieved 72.2% accuracy with the lowest scores in precision (74.7%), recall (72.7%), and f1-score (73%). These results exhibit the model's limited capacity in handling complex feature representations, particularly of the cucurbit dataset.

TP, TN, FP and FN of the analyzed multiclass classification values of the selected CVT is exhibited in Table 3. Every entry pertains to one of the healthy or diseased classes of cucurbit dataset. For case in point, in the class A, TP value of 177 stipulates that 177 class A images are correctly predicted. To indicate that the model is authentic in classifying the cucurbit diseases, the values must be larger for true predictions and smaller for false predictions. Class B is better predicted among other classes whereas class C the least.

Few cases of class B misclassification error have been observed while performing classification, in which the model has generated the least number of FP and FN. In the case of multiclass classification, a dataset class that has been wrongfully classified as any of other classes is labeled as FP and the inability of the model to classify the image of a specific class is depicted as FN.

3.3.3 Results of hyperparameter optimization

The performance of cucurbit disease classification can be further improved through effective hyperparameter optimization. All metaheuristic algorithms employed in this work were executed for 50 iterations and a population size of 10. The fitness of each solution was evaluated based on the classification accuracy of the CVT model after three training epochs.

Particle swarm optimization

A flock of birds utilizes a subtle mixture of knowledge from their personal instincts and group awareness while navigating long distances in search of food. In 1995, this unique search behavior served as an inspiration to develop "particle swarm optimization" (PSO) algorithm for Kennedy and Eberhart [30]. PSO begins with initializing a population of particles where each particle represents a

potential solution to the problem. In every iteration, the particles update their position and velocity based on their experience and the neighbors' experience to find optimal solutions.

A comparative analysis of the CVT-HHO and CVT-PSO is presented in Fig. 8 and 9. HHO achieves the highest accuracy of 96.1%, outperforming PSO's 94.4%. In terms of precision and recall, HHO also maintains a slight edge, with precision of 96.7% and recall of 96.2%, compared to 96.2% and 95.7% for PSO, respectively. The f1-score further reflects HHO's superiority, achieving 96.2% versus 96% from PSO. These results demonstrate that optimization of CVT using HHO not only improves classification accuracy but also ensures more consistent and reliable predictions.

3.3.4 Results of classification

The overall performance analysis of all ten ML classifiers on the cucurbit dataset is provided in Table 4. These findings support the integration of metaheuristic optimization and ML with attention-based deep feature extraction, as proposed in this study. Among all the classifiers, CSVM outperformed others across almost all metrics, achieving the highest accuracy of 96.96%, f1-score of 97.2%, precision of 96.7%, and recall of 97.2%, indicating its better generalization and prediction capacity.

Closely following CSVM, SVM also showed outstanding results with an accuracy of 96.8%, precision and f1-score of 97%, and recall of 96.5%, highlighting its strong capability in handling complex, non-linear data distributions. Similarly, CB, LGBM, and QSVM achieved high accuracies of 96.4% and consistent f1-scores of 96.5%, reflecting their efficiency and reliability in classification, due to their ensemble or kernel-based nature.

Other ensemble models like RF and XGB also demonstrated competitive performance, with RF achieving 96.2% accuracy and XGB at 95.8%, both maintaining a good balance between precision and recall. These results suggest that ensemble methods can leverage multiple weak learners to produce strong predictive performance. On the other hand, traditional models like GNB and DT showed relatively lower performance, with accuracies of 95.1% and 94.1%, respectively.

The confusion matrix for the CVT-HHO-CSVM is presented in Fig. 10. The results demonstrate the proposed model's capacity to generalize, be consistent during training, and maintain accuracy while dealing with disease data. Class B showed the highest classification accuracy with 97%, indicating it is the most easily distinguishable. In contrast, Class D had the highest misclassification rate, particularly confused with Class A and C, suggesting overlapping features between them.

The receiver operating characteristics (ROC) graph is given in Fig. 10 for the proposed model. Here all the four classes have curves above the random classifier which means that the model has high sensitivity and specificity. The AUC values for class A, B, and C are all 1.00, indicating perfect discrimination between the positive and negative instances for these classes. Similarly, class D

achieved AUC scores of 0.99, reflecting near-perfect classification performance. All ROC curves are tightly clustered near the top-left corner of the plot, signifying a high true positive rate and a low false positive rate across varying thresholds. These results suggest that the model demonstrates excellent generalization and robustness in distinguishing between the multiple classes.

Table 5 presents the proposed model training performance metrics for all the four cucurbit leaf disease categories. For class A, the CVT-CSVM attained 96% precision which shows that almost 96% of the estimated images were perfect. The 98% recall points out that the model can predict 98% of total actual class A samples piloting to an f1-score of 97%, which strikes an effective equilibrium between precision and recall. For class B, the model performed even better, with a precision of 100% and a recall of 97%, showing that it can correctly predict the majority of samples. The f1-score of 99% indicates a strong balance between precision and recall. For class C, the proposed model performs better, with a precision of 96% and a recall of 99%. The f1-score of 98% demonstrates the suggested model's performance in balancing false positives and false negatives of class C. Finally, class D has the highest precision value of 97% but the lowest recall of 93%. The f1-score of 95% indicates that the model efficiently differentiates mosaic virus.

Fig. 11 shows the five-fold cross validation results for all the classifiers on the dataset. Since we have a limited dataset, in order to ensure generalizability, five-fold cross validation is conducted. The graph clearly shows that model provides a consistent performance of 95% to 98% for all classifiers.

3.4 Existing works

The proposed work on the cucurbit dataset is compared with various datasets including a pepper dataset (Osprey Optimization - OO) [20], Plant Village subset [22] (Enhanced Ganet Optimization Algorithm - EGOA), and DiaMOS dataset [23] (Extended Stochastic Coati Optimization - ESCO). Table 6 shows the performance analysis of the proposed method on different datasets contrasted with existing approaches.

3.5 Ablation study

To highlight the performance of the alterations performed to achieve the final model, ablation experiments were conducted on the cucurbit dataset as shown in Table 7. Comparing each method step-by-step, this table shows the models and its performance on the dataset. The proposed method achieves 7.24% higher with the classifier and optimization from the original model. This study indicates that the proposed methodology is valid for plant disease detection.

3.6 Discussion

The proposed model achieved 96.96% accuracy outperforming other nine benchmark classifiers in classifying cucurbit leaf diseases. This result affirms the strength of combining DL-based feature extraction from a CVT with the cubic kernel function capability of SVM, enhancing robustness. The

confusion matrix analysis indicates strong per-class performance, particularly in distinguishing class B, which had the highest f1-score. Some misclassifications occurred between visually similar disease types such as class A, C and D, likely due to overlapping visual symptoms. Nevertheless, the overall class-wise f1-scores remained consistently above 95%, demonstrating the model's effectiveness in handling intra-class variability.

Table 8 shows the importance of preprocessing on the original dataset. There has been a 1.14% increase in accuracy with appropriate preprocessing.

As mentioned in Table 9, the proposed model is lightweight with 1 mb space and 0.257 million training parameters and can be used for real-world applications.

Future work may explore ensemble learning or attention-guided interpretability to enhance trust and explainability in field deployments. The model's ability to categorize plant diseases beyond cucurbits should also be investigated. Additional optimization approaches with reduced computational complexity should be introduced to explore the efficacy of hyperparameter search methods, and other model hyperparameters can be examined in future studies

4. Conclusion

In this paper, an HHO-based CVT is presented for cucurbit disease detection. The model facilitates the usage of CVT for feature extraction, utilizing the HHO algorithm for effective hyperparameter optimization and CSVM for classification. The implementation results on the cucurbit diseases dataset prove that the proposed model demonstrates superior performance across various metrics. The model outperformed several well-known lightweight DL architectures, such as LeNet, SqueezeNet, with 96.96% accuracy, 97.2% precision, 96.7% recall, and 97.2% f1-score. In the future, the proposed model, presented for cucurbit disease detection, shall be extended to other plants and image processing applications. Although HHO helped in selecting the near-optimal set of hyperparameters, it contributes to increased complexity, which may be considered a minor limitation. Future work may explore object detection and automatic segmentation networks for detecting the diseased lesions.

Conflict of interest: The authors of this publication declare there are no competing interests.

Submission declaration: The authors declare that the work submitted has not been published previously.

Contribution: All authors have read and approved the manuscript.

R Monisha - Conceptualization, Data curation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing.

KS Tamilselvan - Conceptualization, Methodology, Supervision, Writing – review & editing.

L Karthiba – Dataset curation, Dataset classification.

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

1. Schaefer, H. and Renner, S. S. "Phylogenetic relationships in the order cucurbitales and a new classification of the gourd family (*Cucurbitaceae*)", *Taxon*, 60(1), pp. 122-138 (2011). https://doi.org/10.1002/tax.601011
2. Cohen, Y., Van den Langenberg, K.M., Wehner, T.C., et al. "Resurgence of *Pseudoperonospora cubensis*: the causal agent of cucurbit downy mildew", *Phytopathology*, 105(7), pp. 998-1012 (2015). https://doi.org/10.1094/PHYTO-11-14-0334-FI
3. Mondal, B., Mondal, C. K., and Mondal, P. "Deaseas of cucurbits and their management", In *Stresses of cucurbits: Current status and management*, pp. 115-222. Singapore: Springer Singapore. (2020). https://doi.org/10.1007/978-981-15-7891-5_3
4. Singh, V., Sharma, N., and Singh, S. "A review of imaging techniques for plant disease detection", *Artificial Intelligence in Agriculture*, 4, pp. 229-242 (2020). https://doi.org/10.1016/j.aiia.2020.10.002
5. Cuevas, E. and Rodríguez, A. N. "Image processing and machine learning, volume 2: advanced topics in image analysis and machine learning", In *Image Analysis and Machine Learning*, Chapman and Hall/CRC, (2024). https://doi.org/10.1201/9781032662466
6. Sarkar, C., Gupta, D., Gupta, U., et al. "Leaf disease detection using machine learning and deep learning: review and challenges", *Applied Soft Computing*, 145, 110534 (2023). https://doi.org/10.1016/j.asoc.2023.110534
7. Ahmed, I. and Yadav, P. K. "Plant disease detection using machine learning approaches", *Expert Systems*, 40(5), e13136 (2023). https://doi.org/10.1111/exsy.13136
8. Sahu, S. K. and Pandey, M. "An optimal hybrid multiclass svm for plant leaf disease detection using spatial fuzzy c-means model", *Expert systems with applications*, 214, 118989 (2023). https://doi.org/10.1016/j.eswa.2022.118989
9. Upadhyay, A., Chandel, N.S., Singh, K.P., et al. "Deep learning and computer vision in plant disease detection: a comprehensive review of techniques, models, and trends in precision agriculture", *Artificial Intelligence Review*, 58(3), 92 (2025). https://doi.org/10.1007/s10462-024-11100-x
10. Salka, T.D., Hanafi, M.B., Rahman, S.M., et al. "Plant leaf disease detection and classification using convolution neural networks model: a review" *Artificial Intelligence Review*, 58(10), pp. 1-66 (2025). https://doi.org/10.1007/s10462-025-11234-6

11. Liu, Y., Wang, Z., Wang, R., et al. “Flooding-based mobilenet to identify cucumber diseases from leaf images in natural scenes”, *Computers and Electronics in Agriculture*, 213, 108166 (2023). https://doi.org/10.1016/j.compag.2023.108166
12. Raufer, L., Wiedey, J., Mueller, M., et al. “A deep learning-based approach for the detection of cucumber diseases”, *PloS one*, 20(4), e0320764 (2025). https://doi.org/10.1371/journal.pone.0320764
13. Zeng, Q., Niu, L., Wang, S., et al. “SEViT: a large-scale and fine-grained plant disease classification model based on transformer and attention convolution”, *Multimedia Systems*, *29*(3), pp. 1001-1010 (2023). https://doi.org/10.1007/s00530-022-01034-1
14. Li, R., Su, X., Zhang, H., et al. “Integration of diffusion transformer and knowledge graph for efficient cucumber disease detection in agriculture”, *Plants*, 13(17), 2435 (2024). https://doi.org/10.3390/plants13172435
15. Wang, Y., Li, T., Chen, T., et al. “Cucumber downy mildew disease prediction using a cnn-lstm approach”, *Agriculture*, 14(7), 1155 (2024). https://doi.org/10.3390/agriculture14071155
16. Krizhevsky, A., Sutskever, I., and Hinton, G. E. “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, 25, (2012). https://doi.org/10.1145/3065386
17. Dosovitskiy, A. “An image is worth 16x16 words: transformers for image recognition at scale”. *arXiv preprint arXiv:2010.11929*, (2020). https://doi.org/10.48550/arXiv.2010.11929
18. Hassani, A., Walton, S., Shah, N., et al. “Escaping the big data paradigm with compact transformers”, *arXiv:2104.05704*, (2021). https://doi.org/10.48550/arXiv.2104.05704
19. Daniya, T. and Vigneshwari, S. “Rider water wave-enabled deep learning for disease detection in rice plant”, *Advances in Engineering Software*, 182, 103472 (2023). https://doi.org/10.1016/j.advengsoft.2023.103472
20. Begum, S.S.A. and Syed, H. “GSAtt-CMNetV3: pepper leaf disease classification using osprey optimization”, *IEEE Access*, 12, pp. 32493-32506 (2024). https://doi.org/10.1109/ACCESS.2024.3358833
21. Vijay, C. P. and Pushpalatha, K. “DV-PSO-Net: A novel deep mutual learning model with heuristic search using particle swarm optimization for mango leaf disease detection”, *Journal of Integrated Science and Technology*, 12(5), pp. 804-804 (2024). https://doi.org/10.62110/sciencein.jist.2024.v12.804
22. Nagachandrika, B., Prasath, R., and Joe, I. P. “An automatic classification framework for identifying type of plant leaf diseases using multi-scale feature fusion-based adaptive deep

network", *Biomedical Signal Processing and Control*, 95, 106316 (2024). https://doi.org/10.1016/j.bspc.2024.106316

23. Indumathi, P. and Kumuthaveni, R. "Coati optimized transfer leaning with vision transformer model for improving deep learner based plant diseases detection", *International Journal of Intelligent Engineering & Systems*, 17(4), (2024). 10.22266/ijies2024.0831.21
24. Mehzabeen, S. M. and Gayathri, R. "Heuristically improvised rice disease classification framework based on adaptive segmentation with the fusion of lstm layer into multi-scale residual attention network", *Biomedical Signal Processing and Control*, 99, 106875 (2025). https://doi.org/10.1016/j.bspc.2024.106875
25. Rajareddy, G. N., Mishra, K., Satti, S. K., et al. "A digital twin-enabled fog-edge-assisted IoAT framework for *Oryza Sativa* disease identification and classification", *Ecological Informatics*, 87, 103063 (2025). https://doi.org/10.1016/j.ecoinf.2025.103063
26. Heidari, A.A., Mirjalili, S., Faris, H., et al. "Harris hawks optimization: algorithm and applications", *Future Generation Computer Systems*, 97, pp. 849–872 (2019). https://doi.org/10.1016/j.future.2019.02.028
27. Mafi, M.M.H.M. and Khasru, M.A. "Sweet pumpkin disease recognition dataset", *Mendeley Data*, V1 (2024). 10.17632/7397w9h5tb.1
28. Setiawan, A.W., Mengko, T.R., Santoso, O.S. et al. "Color retinal image enhancement using CLAHE", *Int. Conf. on ICT for Smart Society*, Jakarta, Indonesia, pp. 1–3 (2013). https://doi.org/10.1109/ICTSS.2013.6588092
29. Rother, C., Kolmogorov, V. and Blake, A. "GrabCut: interactive foreground extraction using iterated graph cuts", *ACM Transactions on Graphics*, 23(3), pp. 309–314 (2004). https://doi.org/10.1145/1015706.1015720
30. Eberhart, R. and Kennedy, J. "A new optimizer using particle swarm theory", *Proc. 6th Int. Symp. Micro Machine and Human Science, Nagoya*, Japan, pp. 39–43 (1995). https://doi.org/10.1109/MHS.1995.494215

Figure captions:

Fig. 1. Illustration of the proposed scheme.

Fig. 2. Example images from the dataset.

Fig. 3. Structure of the CVT [18].

Fig. 4. Flowchart for hyperparameter selection.

Fig. 5. Image processing examples (a) Input samples,

(b) CLAHE samples, (c) GrabCut samples.

Table captions:

Figures:

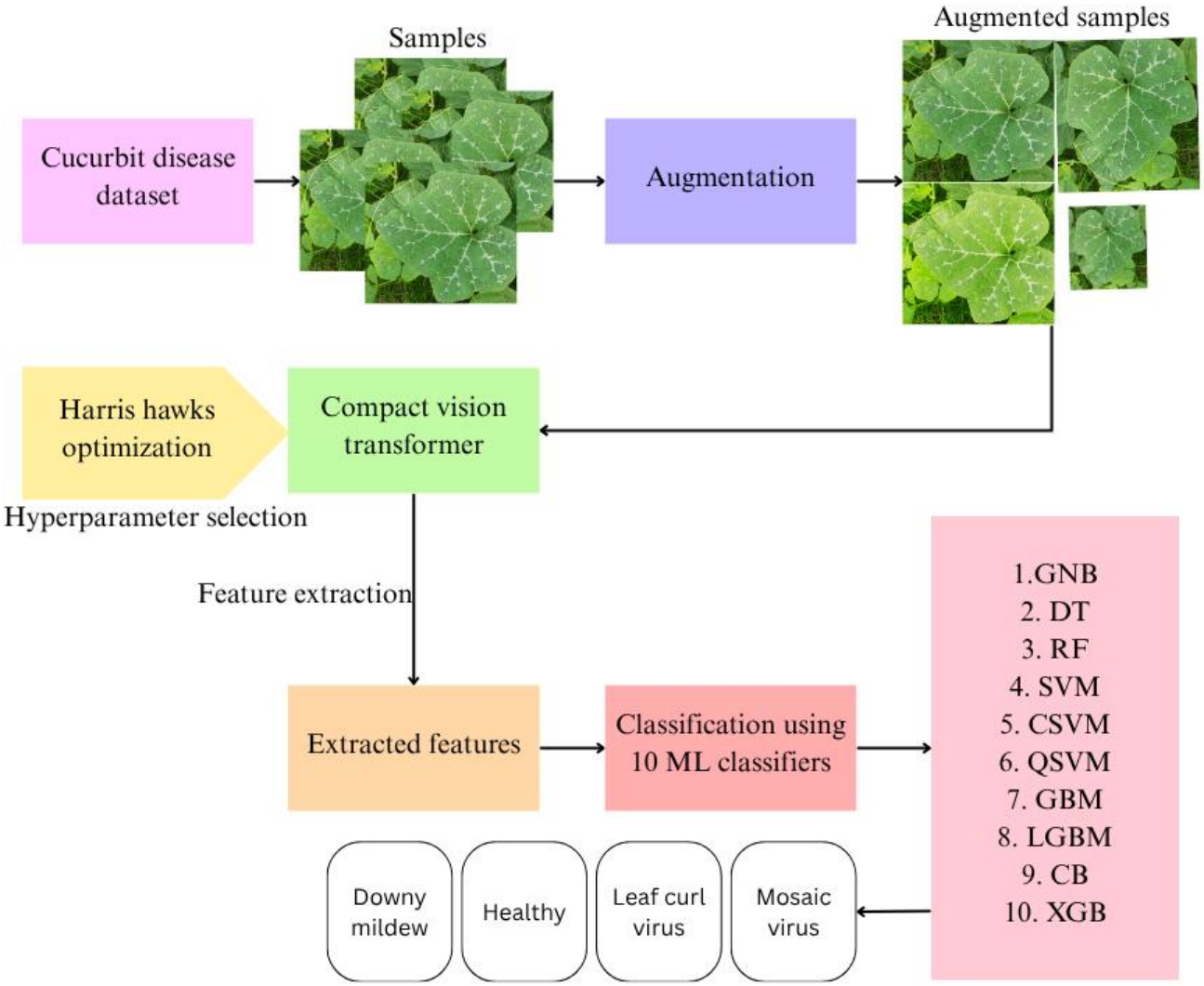
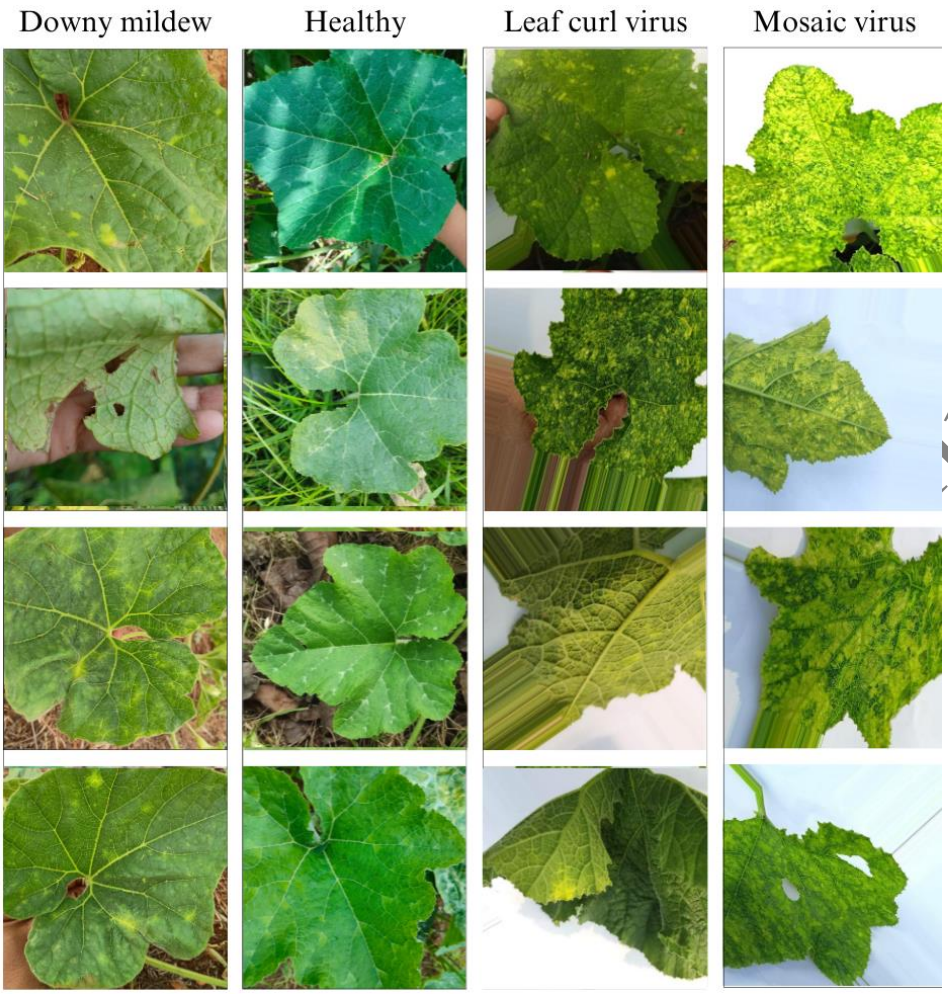


Fig. 1.



572
573 **Fig. 2.**

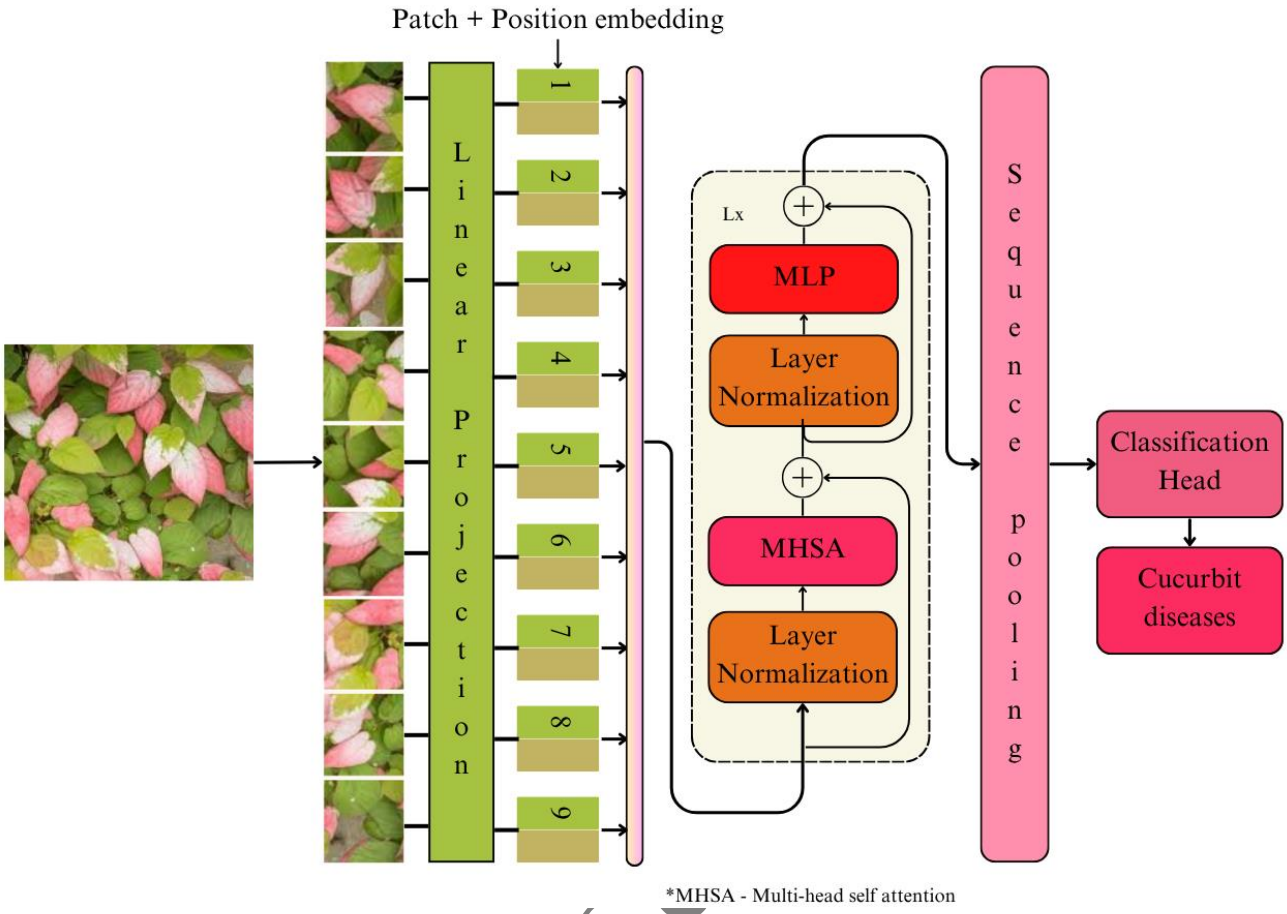


Fig. 3.

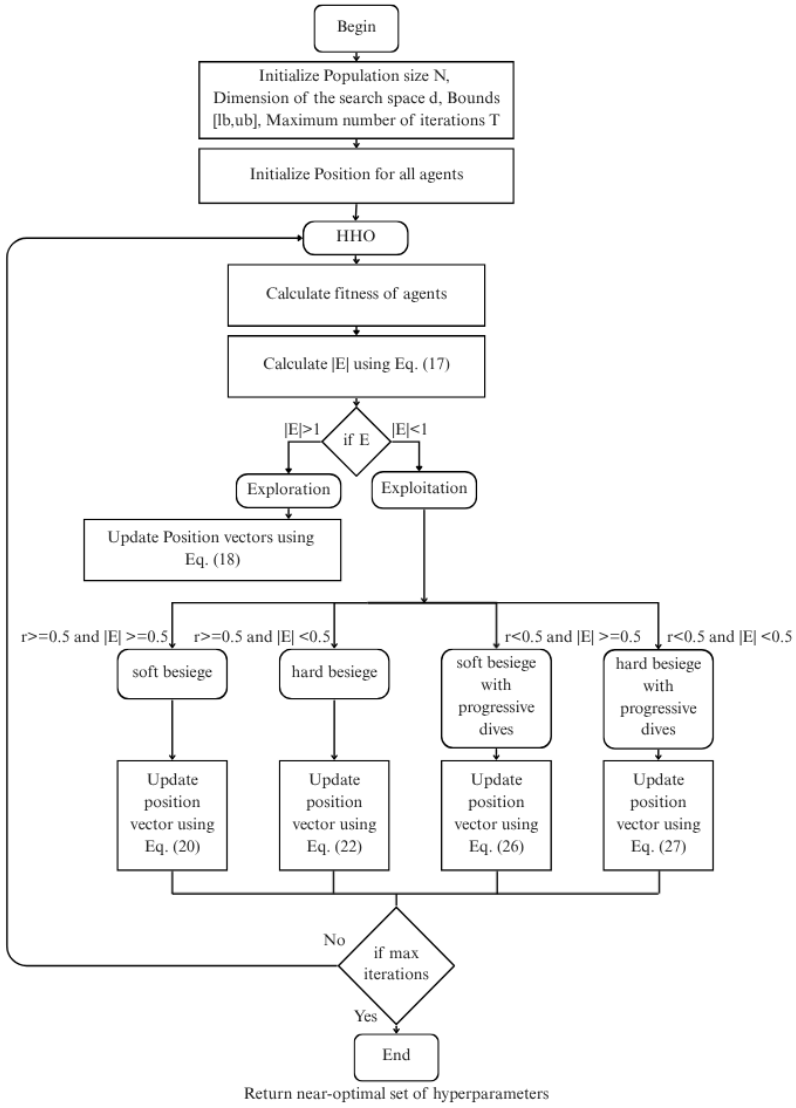


Fig. 4.

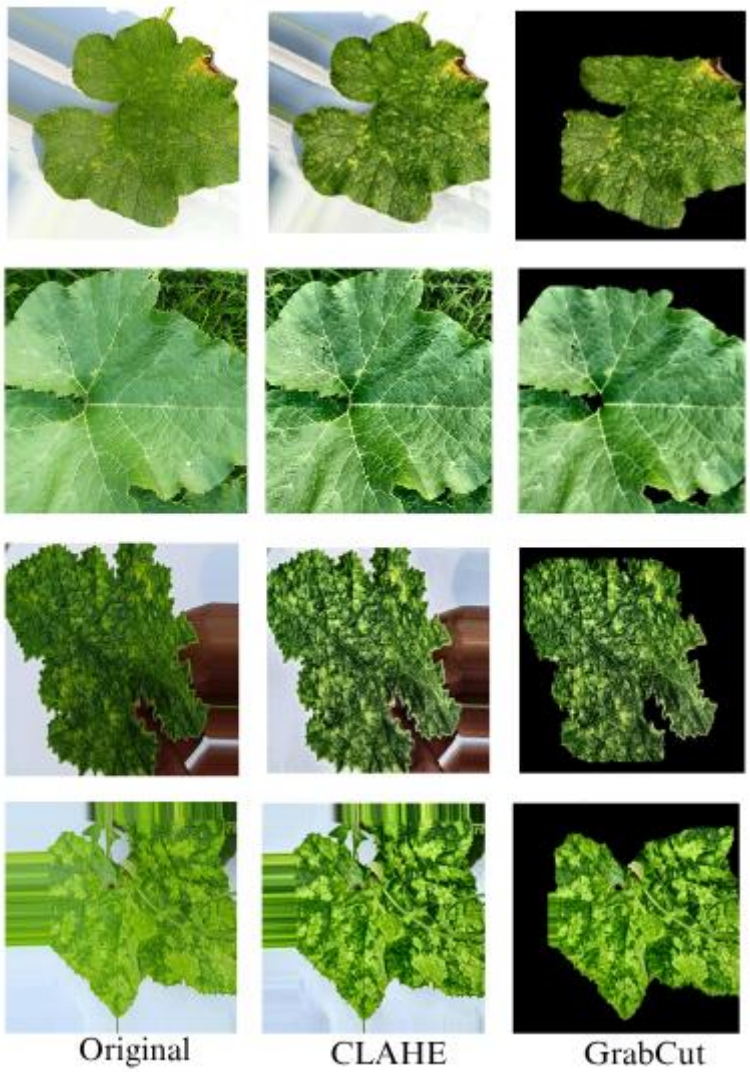


Fig. 5.

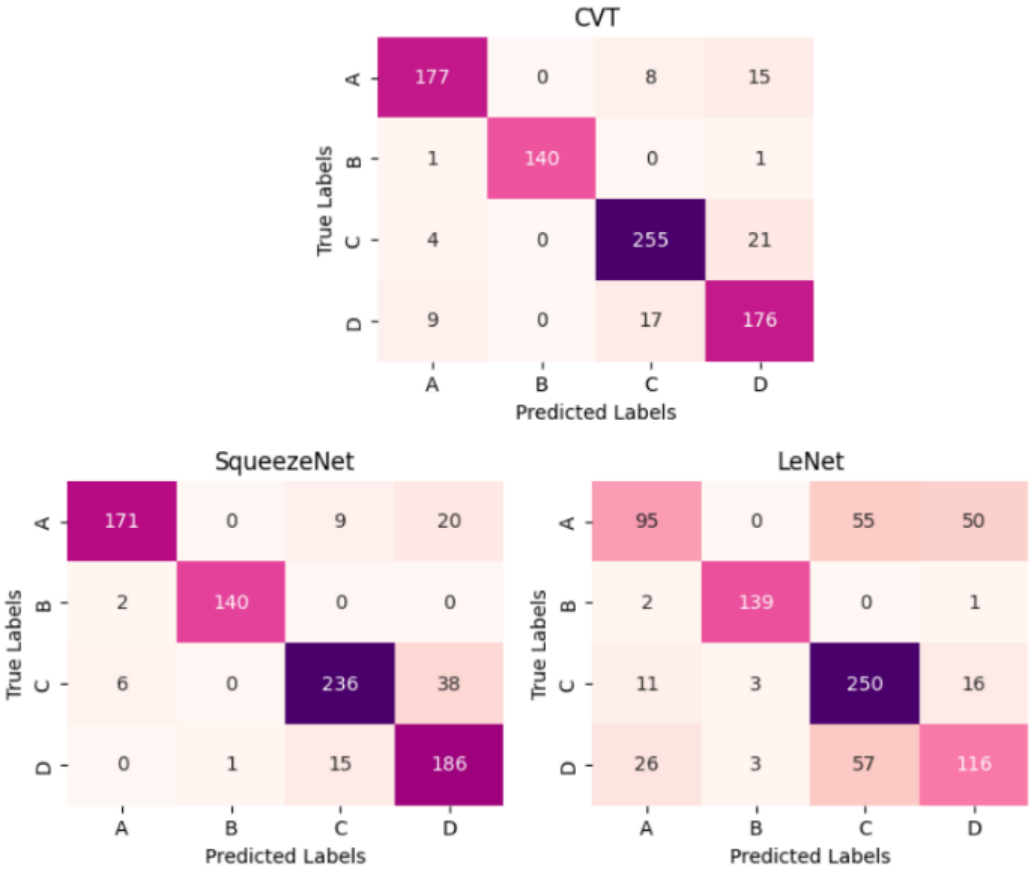

CVT
True Labels
Predicted Labels
A
B
C
D
177 0 8 15
1 140 0 1
4 0 255 21
9 0 17 176
SqueezeNet
171 0 9 20
2 140 0 0
6 0 236 38
0 1 15 186
LeNet
95 0 55 50
2 139 0 1
11 3 250 16
26 3 57 116

Fig. 6.

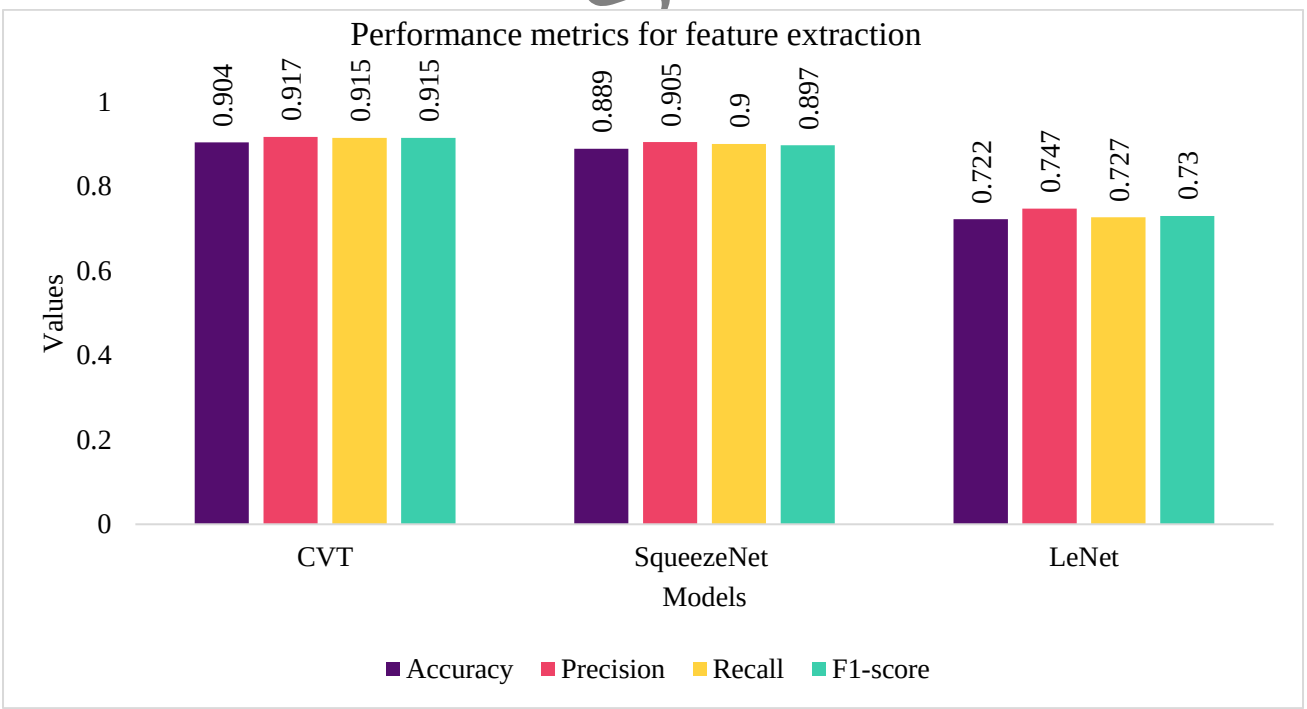

Performance metrics for feature extraction
Values
Models
CVT
SqueezeNet
LeNet
0.904 0.917 0.915 0.915
0.889 0.905 0.9 0.897
0.722 0.747 0.727 0.73
Accuracy
Precision
Recall
F1-score

Fig. 7.

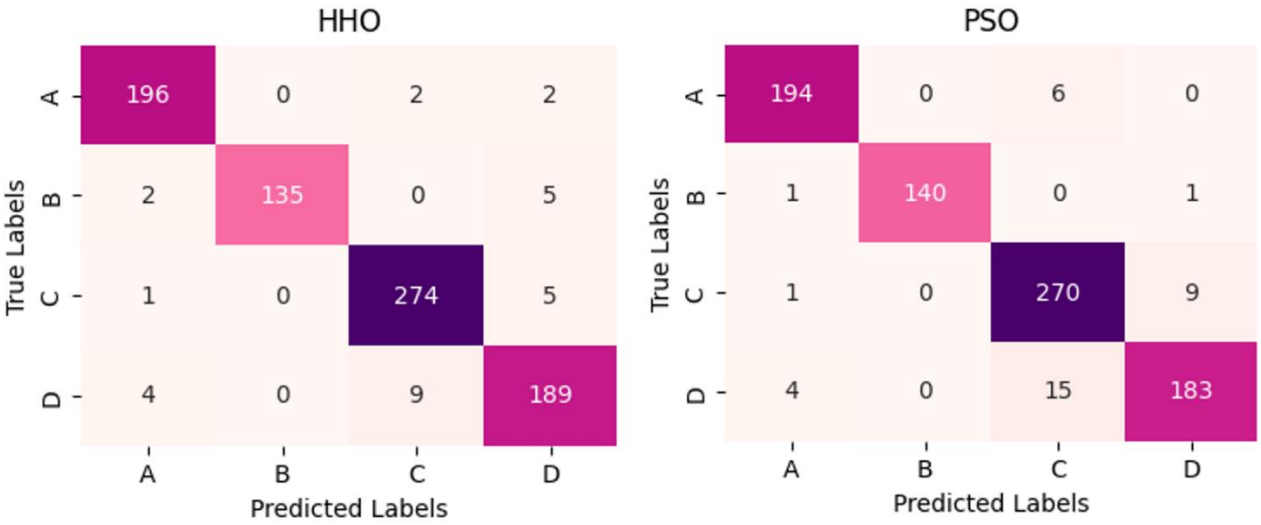


Fig. 8.

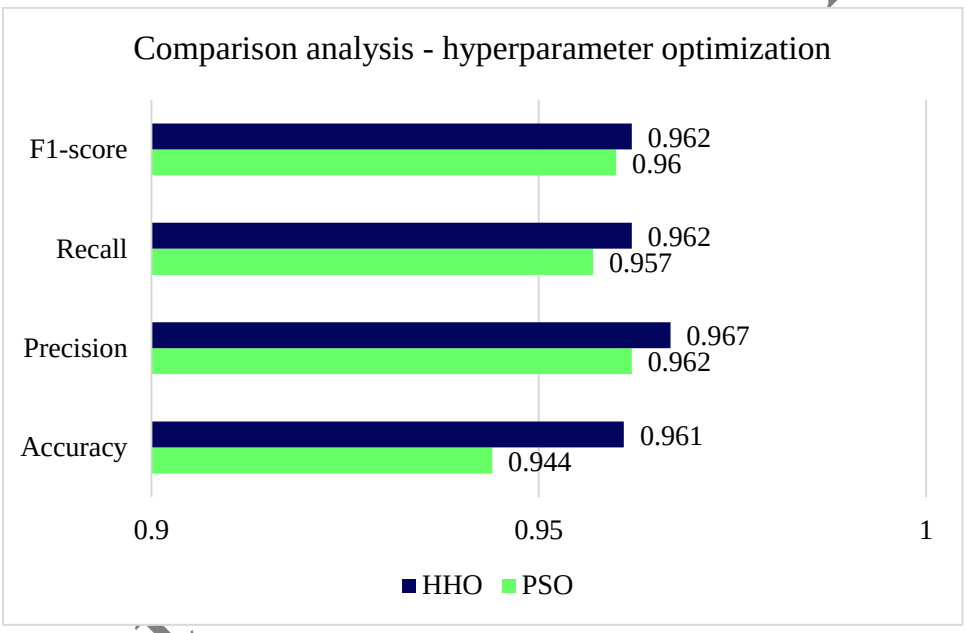


Fig. 9.

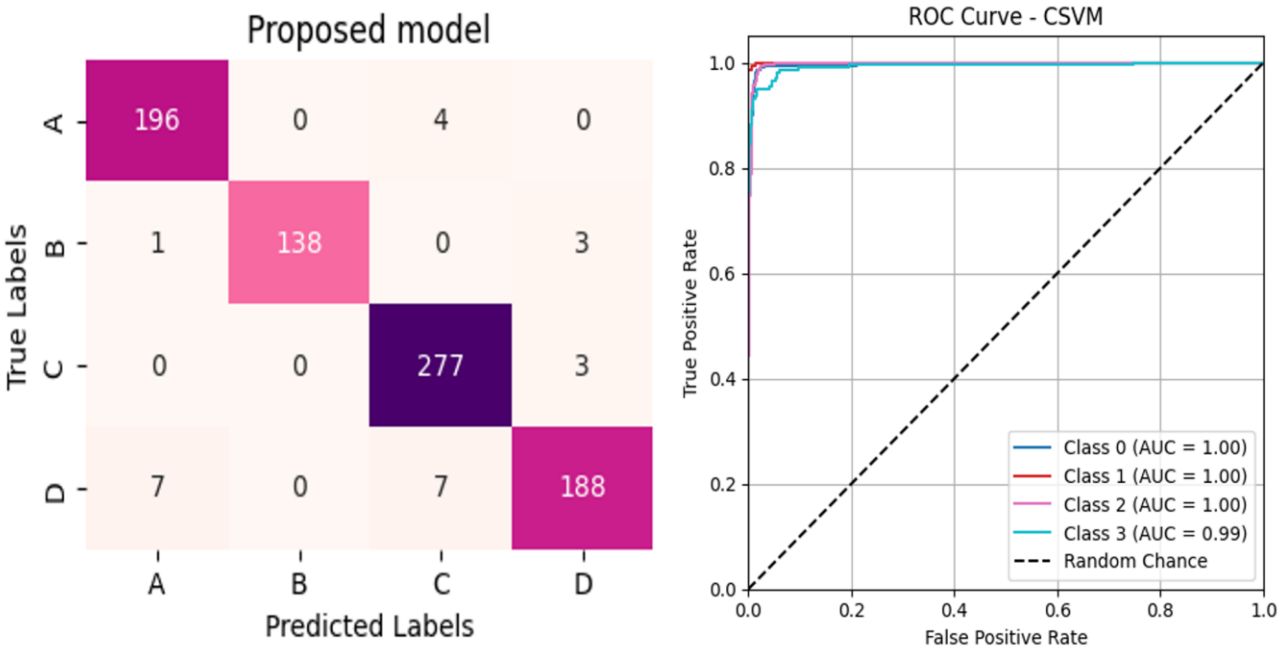


Fig. 10

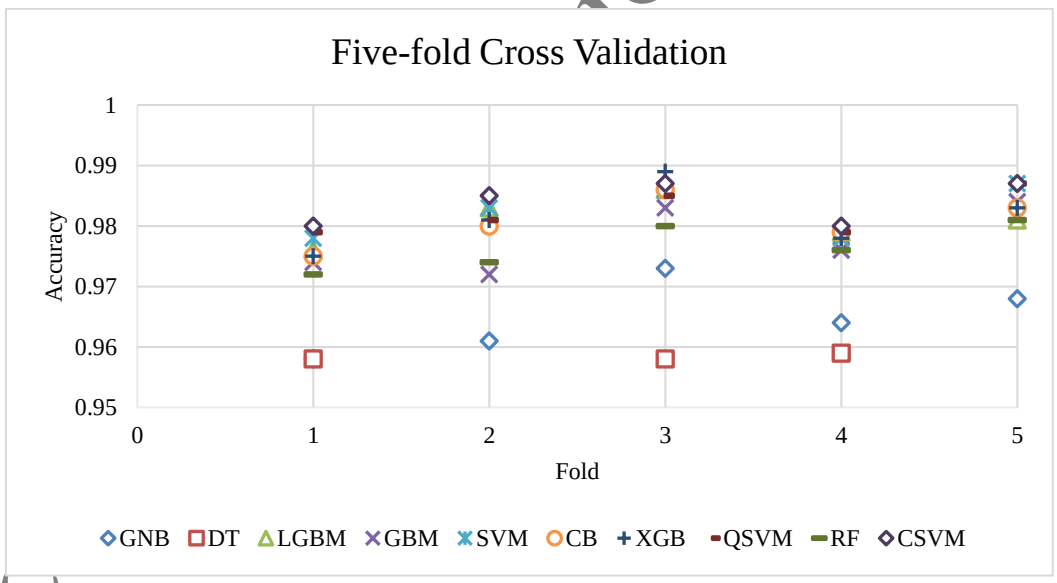


Fig. 11. Five-fold cross validation of classifiers.

Tables:

Table 1

Class	Name	Training images	Validating images	Testing images	Total images
A	Downy mildew	700	200	100	1,000
B	Healthy	498	142	71	711

C	Leaf curl virus	980	280	140	1,400
D	Mosaic virus	707	202	101	1,010
Total		**2,885**	**824**	**412**	**4,121**

Table 2.

S. No	Hyperparameter	Value	S. No	Hyperparameter	Value
1	Projection dimension	112	7	Learning rate	0.001
2	Transformer layers	1	8	Optimizer	Nadam
3	Self-attention heads	3	9	Image size	128
4	Patch size	8	10	Stochastic depth rate	0.1
5	Batch size	64	11	Weight decay	0.0001
6	Dropout rate	0.1	12	Label smoothing	0.1

Table 3.

CVT	TP	FP	FN	TN
Class A	177	14	23	610
Class B	140	0	2	682
Class C	255	25	25	519
Class D	176	37	26	585

Table 4.

S. No	Classifier	Accuracy	Precision	Recall	F1-score
1	DT	0.941	0.942	0.94	0.942
2	NB	0.951	0.957	0.947	0.955
3	GBM	0.956	0.96	0.952	0.957
4	XGB	0.958	0.96	0.957	0.957
5	RF	0.962	0.967	0.962	0.965
6	LGBM	0.964	0.97	0.962	0.965
7	QSVM	0.964	0.967	0.962	0.965

8	CB	0.964	0.97	0.962	0.965
9	SVM	0.968	0.97	0.965	0.97
10	CSVM	0.969	0.972	0.967	0.972

601 **Table 5.**

Class	Accuracy	Precision	Recall	F1-score
A	0.941	0.96	0.98	0.97
B	0.97	1.00	0.97	0.99
C	0.951	0.96	0.99	0.98
D	0.904	0.97	0.93	0.95

602 **Table 6.**

Ref	Dataset name	Algorithm	Accuracy (%)
[23]	DiaMOS	ESCO	0.944
[22]	Plant village subset – multiple	EGOA	0.947
[20]	Pepper dataset	OO	0.978
Proposed model	DiaMOS	HHO	0.949
Proposed model	Plant village subset – multiple	HHO	0.973
Proposed model	Pepper dataset	HHO	0.985

603 **Table 7.**

Model	Accuracy
CVT	0.904
CVT-HHO	0.961
CVT-HHO-CSVM	0.969

604 **Table 8.**

Model	Accuracy
Proposed model – Original dataset	0.951
Proposed model – CLAHE	0.958
Proposed model – CLAHE + Grabcut	0.969

605 **Table 9.**

Model	Training parameters	Memory size (mb)
LeNet	1,628,216	6.21
SqueezeNet	737,476	2.81
Proposed model	257,157	1

606

Biographies:

R. Monisha has completed B.E in Electronics and Communication Engineering at Sri Krishna College of Engineering and Technology, Coimbatore, India, in 2019. She has completed M.E in VLSI Design at Narasu's Institute of Technology, Salem, India, in 2021. She is currently a full-time Research Scholar at KPR Institute of Engineering and Technology, Coimbatore, India. Her area of research and interests include image processing, machine learning, deep learning, and precision agriculture. She has published several book chapters, research articles, and patents during this period.

K. S. Tamilselvan is currently a Professor in Biomedical Engineering Department at KPR Institute of Engineering and Technology, Coimbatore, India. He completed his Ph.D at Anna University in the image processing domain. He is guiding several research projects including undergraduate and postgraduate. Over the years he has published several SCI E, Scopus indexed articles, and Patents. His area of interests includes image processing, medical imaging, precision agriculture, and agricultural devices.

L. Karthiba is currently an Assistant Professor (Plant Pathology) in Department of Pulses, Centre for Plant Breeding and Genetics, Tamil Nadu Agricultural University, Coimbatore, India. She has completed her Ph.D at Tamil Nadu Agricultural University. Her area of interests includes plant pathology, biological control, and sustainable agriculture. Over the years, she has guided undergraduate and postgraduate students, supervised research projects, and published papers in reputed journals. She has over 8 SCI publications and has more than five years of teaching experience.