RESEARCH PAPER

Deepleafnet+: A Comprehensive Deep Learning Approach With Deeplabv3++E-Ahc Segmentation for Plant Leaf Disease Detection and Insecticide Recommendation

[1*]P. Saranya Devi and [2]A. Senthil Rajan

[1] Ph.D. Research Scholar, Department of computer science, Alagappa University, Karaikudi*
[2] Registrar, Professor & Head, Department of computer science, Alagappa University
[1]saranyadhamu123@gmail.com and [2]agni_senthil@yahoo.com
*[*]**Corresponding Author:** saranyadhamu123@gmail.com*

Received: 24th April, 2026; Revised: 20th May 2026; Accepted: 30th June, 2026; Available Online: 10th July, 2026

ABSTRACT

Plant Leaf disease has grown global concern, and researchers must pay attention to it. To conserve the harvests, plant leaf disease identification and detection is essential. Also recommending proper insecticide application for the plants based on the plant disease is more challenging. This paper provides a new method for identifying leaf diseases and recommending insecticides by combining Deep Learning (DL) at different levels. For this purpose, the advanced preprocessing techniques such as data augmentation, image enhancement, and normalization to obtain high-quality input data are used. For segmentation, DeepLabV3++E-AHC method segments the leaves with patterns and distinguishes them with the help of enhanced attention mechanisms and hybrid convolutions. A combination of deep features, texture, color, shape, and region-based features are extracted during the feature extraction step. Next, feature selection uses the hybrid Starling-Revamped Hummingbird Optimization (SRHO) model. The disease detection component uses a multi-backbone structure with self-attention blocks such as the LeafNet+ model to improve the classification's performance. Finally, reinforcement learning (RL) is applied to recommend the best insecticide based on the detected diseases. The LeafNet+ scheme enhances the efficiency of detecting leaf diseases and provides recommendations for efficient pest control. The experimental outcomes of the developed technique are compared with an existing technique, it has gained an accuracy of 0.997035, precision of 0.997027, and F1 score is 0.997037. The developed model can identify and detect the plant leaf disease and recommend insecticides to enhance agricultural development.

Keywords: Plant Leaf Disease Detection, Insecticide Recommendation, Deep Learning, Smart Agriculture, Hybrid Optimization.

How to cite this article: Saranya Devi P, Senthil Rajan A, Deepleafnet+: A Comprehensive Deep Learning Approach With Deeplabv3++E-Ahc Segmentation for Plant Leaf Disease Detection and Insecticide Recommendation. Int J Drug Deliv Technol. 2026;16(77s): 1690-1712; DOI: 10.25258/ijddt.16.77s.196
Source of support: Nil.
Conflict of interest: None

1. INTRODUCTION

A significant portion of India's Grass Domestic Product (GDP) about 15.87% comes from the agricultural sector, which also employs around 54.15% of the labor force [1]. Despite of significant advancements in recent years, plant leaf loss due to pests and diseases continues a major concern. This affects the yield, population, health, quality of living, and economy worldwide, and results in a loss for the farmer [2, 3]. The current period of enhanced global trade and climate change presents a new challenge for the search of effective diagnostic techniques for the detection and treatment of plant diseases [4]. Many dangerous "alien" species, such as bacteria, fungi, viruses, microorganisms, weeds, insects, worms, and plant materials, move unharmed with people and other items (including plant materials) [5]. As a result, they spread widely over the world and seriously disrupted agriculture. To address the disease infections at an early stage, it is imperative to identify and diagnose plant leaf disease [6]. It reduces the risk of disease spreading and stops new cases from emerging.

Many academics have recently focused on image-based plant illness identification. Accurate and disease identification in time are becoming more important as plant leaf waste from diseases rises [7]. The majority of people in these nations work in agriculture, either directly or indirectly. Application-driven crop disease detection techniques are essential in these regions as they allow farmers to recognize the sources of diseases and take preventative action to treat them [8, 9]. Early identification of plant diseases based on characteristics like leaf color, size, and development pattern is beneficial to farmers. A large number of individuals can use basic internet services [10]. For the purpose for convenience, more than 300 million individuals utilize the Internet for a variety of purposes. Governments have developed various resources, such as round-the-clock helplines for farmers, to address their inquiries; yet, residents in rural areas sometimes

**Author for Correspondence:* saranyadhamu123@gmail.com*

struggle to access these resources, making it difficult for them to find solutions to their issues [11, 12].A straightforward tool that enables farmers to capitalize separately on image-based detection of diseases would be advantageous [13].

While employing pesticides may provide a short-term fix, they may not be beneficial in the long run [14]. The pesticides contain unfavorable effects on the crop that may eventually harm inhabitants' health. In the present day, as AI applications are being used across numerous domains, it would be beneficial to use AI to assist in addressing agricultural problems [15]. Even if there are a vast number of plant leaf and diseases, it appears good to begin by identifying the illnesses that commonly affect Indian plant leaf [16]. Numerous studies have trained DL algorithms for diagnosing illnesses in various plant leaf using publicly available datasets. Nevertheless, custom datasets have also been used in several investigations. DL methods have several benefits for plant disease detection. It can distinguish between illness signs, identify different diseases and cases, gauge the severity of the condition, and create affordable remedies [17, 18].

The main aim of the developed model is to design an innovative framework for leaf disease detection and insecticide recommendation. The system incorporates hybrid feature selection designs, DL-based segmentation and feature extraction, and sophisticated preprocessing approaches. The objective is to precisely identify leaf diseases and select the best pesticides through RL, improving agricultural practices, utilizing cutting-edge frameworks like DeepLabV3++ and DenseNet along with cutting-edge optimization approaches. The created model's primary contributionsare described below.

1.1. Key Contribution

- Hybrid multi-stage feature extraction and selection pipeline - a combination of DenseNet features with texture, color, shape, and region-based measures and refining them with the SRHO optimizer (integration of Starling Murmuration Optimizer + Artificial Hummingbird Algorithm) to improve the quality of features and eliminate redundancy.
- Detailed methods of classifying the disease a multi-backbone network (ResNet, DenseNet, and Inception modules) with self-attention to enhance the recognition accuracy of complex patterns of leaf disease.
- Reinforcement-Learning-based insecticide recommendation the only difference between these is that the results of disease detection are mapped onto environmental conditions to produce adaptive and context-specific pest management recommendations instead of fixed pesticide recommendations.

The paper is organized in the following way: section 2 covers a comprehensive literature review; Section 3 covers an extensive problem definition; Section 4 explains an elaborated proposed approach; Section 5 covers detailed results and discussion; and Section 6 provides a conclusion.

2. RELATED WORKS

Using machine learning (ML, Ruchi et al [19] developed a real-time method to determine the type of illness in a plant leaf. A single-shot detector was employed to identify and locate the leaf, and a Deep Convolutional Neural Network (DCNN) model was suggested for plant leaf disease classification. About 96.88% of the diseases could be classified accurately. Furthermore, given its high success rate in thefield test, the recommended solution is fully deployable.

A technique called cotton crop disease detection using DL via TensorFlow (CottonCare) was created by Nimra and colleagues [20] to incorporate artificial intelligence into farming. Farmers find it helpful to diagnose cotton problems and advise using pesticides. The project's objective was to assist farmers in lowering their production costs and increasing yields, which would boost the national economy.

A DL-based method for diagnosing Apple disease was developed by Khan et al. [21]; it can accurately identify the symptoms in less time. The proposed system runs in two phases: the detection stage locates and detects each symptom from photos of sick leaves, and the first stage uses a custom lightweight classification algorithm to classify the input images as unhealthy. Transfer learning was used to initialize the identification and categorization models, which were subsequently trained on the ready-made dataset. The suggested method produced positive outcomes, with a classification accuracy of about 88%.

A unique mask R-CNN algorithm was created by Gary and colleagues [22], to identify rust and leaf diseases in apple orchards. Three distinct Mask R-CNN networking backbones models were trained and assessed for efficient segmentation, and illness detection. Mask R-CNN-based DLmethods were suggested to perform segmentation. According to the study, a Mask R-CNN system with a ResNet-50 backbone performs well on the job in terms of accuracy, especially when identifying tiny rust disease items on leaves.

Tanmay et al. [23] created a dual operator that combines CNN and the Transition Probability Function (TPF) to analyze the pest's image and recommend insecticide. The proposed method combines two crucial components of farming using CNN and ML: pest diagnosis and insecticide recommendation. Furthermore, a soil NPK sensor was used for the soil nutrient evaluation, and fertilizers were recommended based on the measured nutrient values. Results were provided immediately, and it took less than ten seconds to recommend an insecticide and eighty seconds to recommend a fertilizer.

Adesh and colleagues [24] developed a DL mode that involves labeling sick crop leaves according to the disease pattern. Pixel-based techniques wereused to process images of the affected leaves and enhance the information extracted from the photos. After that, features were retrieved, images were segmented, and crop illnesses were categorized with the help of patterns that were taken from the sick leaves. CNN was used to categorize illnesses.

Sunil et al [25] developed methods for image analysis using ML to identify tomato leaf illnesses. Tomato leaf samples with diseases were taken into consideration. After the tomato leaf specimens were resized, Histogram Equalization was applied to enhance the tomato quality of the samples. Contour tracing was a technique employed in determining the borders of leaf samples. The relevant characteristics of the leaf samples were extracted using several feature extraction approaches. Lastly, many ML techniques were applied to categorize the extracted features.

Sharma et al. [26] suggested a smart system of recommending pesticides, which processes image-based classification of the state of plants with leaf diseases and pest problems. The system incorporates the tasks of identifying the disease and a pesticide advisory system based on knowledge to propose appropriate treatments to minimize the dependency on manual tasks and wrong pesticide application. They showed good prospects of accuracy in their identification of both the diseases and pests in common crop species.

Rajeshram et al. [27] designed a model for the prediction of leaf diseases, pests, and pesticides based on deep learning and implemented in the form of an integrated CNN architecture. It tried to process the entire agricultural diagnostic pipeline, detection, classification, and treatment suggestion, in one automated process. They achieved better performance in disease classification across various classes by using large image datasets and hierarchical features on both effects.

3. PROBLEM DEFINITION

The development of insecticide resistance has become an increasingly important problem: over 55500 species of insects and mites have developed resistance to at least some of the insecticides used to control them. The most common problem inmodern agriculture is identifying crop diseases and suggesting the proper insecticides. That increases productivity and decreases the negative impact on the environment [28]. Diseases are a major threat to crop production because if they are not diagnosed early enough, they reduce the yield and quality of crops and cause huge losses. Conventional diagnosis of diseases was time-consuming and imprecise, and pesticide spraying was frequently overdone, causing damage to the ecosystem and people’s well-being [29]. DL and computer vision (CV), create a system that diagnoses the diseases in crops and suggests the recommendation for the insecticide application, thus saving money and increasing crop productivity. Table 1 below outlines the existing techniques, their objective, strengths, and weaknesses.

Table 1: Analysis of Existing Techniques

Author(s)	Technique	Aim	Advantages	Disadvantages
Ruchi, et al [19]	Deep CNN, Single Shot Detector	Real-time identification and localization of crop diseases using leaf images on Nvidia Jetson TX1.	High classification accuracy (96.88%), real-time deployment.	Limited to Nvidia Jetson TX1 hardware, may require high computational resources.
Nimra, et al [20]	Deep Learning (TensorFlow)	Diagnose cotton diseases, recommend pesticides, and integrate AI into agriculture.	Effective disease detection helps cost reduction, and yields improvement.	Focuses mainly on cotton, and may not generalize well to other crops.
Ilaria, et al [21]	Nano technologies, ICT	Review innovative diagnostic methods and IoT integration for plant disease detection.	Emphasizes field-use diagnostic tools and IoT integration for early detection.	Primarily a review; may lack in detailed experimental results.
Khan, et al [22]	Deep Learning (Two-stage model)	Detect and identify apple diseases with high accuracy and speed.	High accuracy (88%) and mAP (42%), handle multiple symptoms and varying spot sizes.	Detection performance may vary with spot size and image quality.
Gary, et al [23]	Mask R-CNN	Recognize and isolate rust and leaf illnesses in apple orchards.	Effective on detecting small rust disease objects, with good accuracy.	Limited to rust disease detection, may require

				extensive training data.
Tanmay, et al [24]	Dual Operator TPF, CNN	Identify pests recommend insecticides, and analyze soil nutrients.	High accuracy (>90%), quick recommendations for insecticides and fertilizers.	Limited to five pests; effectiveness may vary for different pest types.
Adesh, et al [25]	Deep Learning (CNN)	Identify crop diseases, and reduce crop loss due to diseases using image classification.	Utilizes extensive datasets, and effective image classification.	Relies heavily on image quality and preprocessing; may be limited to specific crops.
Sunil, et al [26]	ML (SVM, K-NN, CNN), Image Processing	Detect leaf diseases in tomato plants using various image processing and classification methods.	High classification accuracy (99.6% with CNN), integrates multiple feature extraction methods.	Focuses only on tomato plants; may not be generalizable to other crops.
Sharma et al. [27]	Image-based disease & pest classification with rule-based pesticide advice	To identify plant diseases/pests and recommend suitable pesticides	Good identification accuracy; reduces manual decision-making	Limited scalability to new crops; static pesticide suggestions
Rajeshram et al. [28]	CNN-based deep learning model for disease & pest detection with pesticide recommendation	To automate diagnosis and treatment suggestions in agriculture	High accuracy and integrated workflow	Not validated in real-field conditions; lacks dosage and eco-safety considerations

Based on the survey and analysis, there is a critical need for advanced, efficient, and reliable systems that can detect and diagnose crop diseases early, utilizing cutting-edge technologies to enhance agricultural practices and improve overall Plant management.

4. PROPOSED METHODOLOGY

The entire procedure is divided into several stages to optimize the process of disease recognition on the leaves and the proposal of insecticides. First, the data is preprocessed using data augmentation methods such as flipping and rotation of the images, gamma correction, noise reduction, and normalization of pixel values. For segmentation, DeepLabV3++E-AHC is used to segment complex patterns on the leaves. Feature extraction is processed using DenseNet, texture analysis, color features, shape analysis, and region-based metrics. These features are enhanced by selectingthe best features from the extracted features using the SRHO technique, which comprises the Starling Murmuration Optimizer (SMO) and Artificial Humming bird Algorithm (AHBA). The identification of diseases is improved by the multi-backbone network with ResNet, DenseNet, and Inception Modules accompanied by self-attention mechanisms.

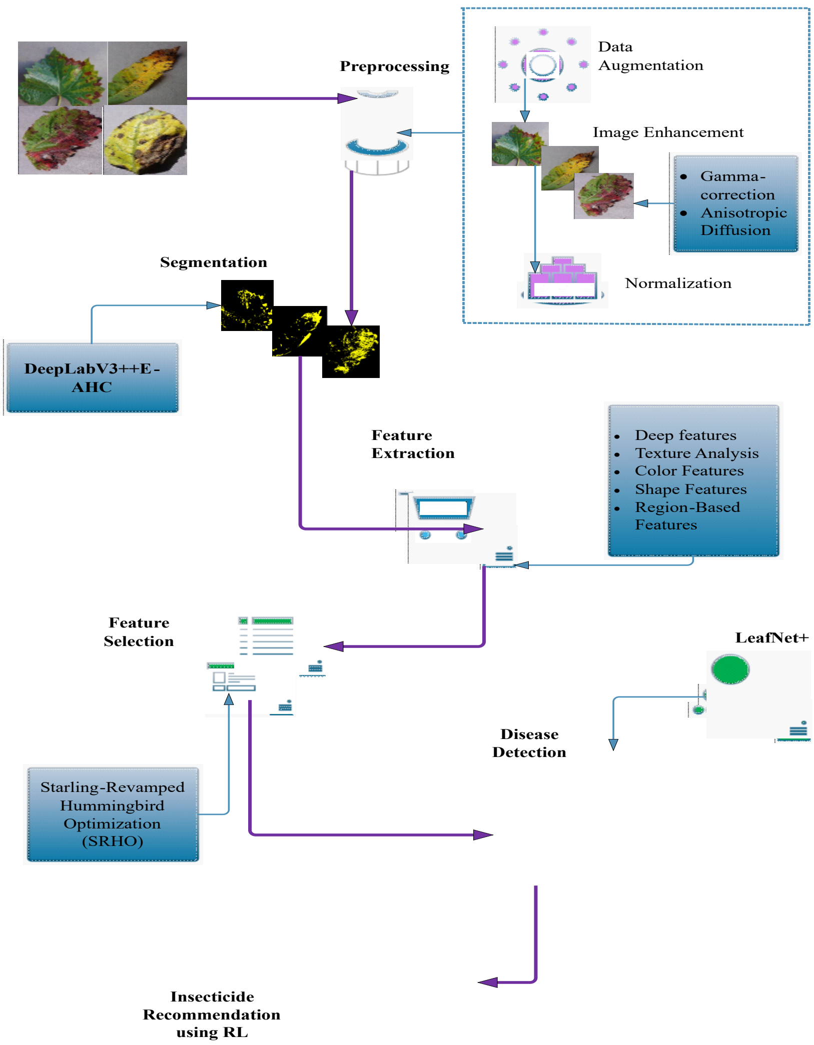

Figure.1: Proposed model architecture

Finally, RL is used to suggest insecticides based on the detected disease and the environmental conditions, to enhance pest management. The overview of the developed technique is shown in Figure.1 above. This developed mode enhances the accuracy of disease diagnosis with context-specific advice on pest management.

4.1 Preprocessing

4.1.1. Data Augmentation

Data augmentation is a significant process used in ML and CV that helps to expand the amount of data in the dataset by applying some operations on the existing data. This is beneficial in enhancing the stability and the transferability of models. Data augmentation is processed using flipping, rotation, and color space transformations. Vertical flipping spins the image 180 degrees, whereas horizontal flipping flips the image from left to right. Images are rotated randomly within a predetermined range at a random angle. While contrast adjustment adjusts the variation in pixel intensities, brightness adjustment modifies the image's lighting level. Hue, saturation, and value are all adjusted while the image is changed to the HSV color system; the saturation adjustment focuses on the color intensity of the image.

4.1.2. Image Enhancement

Gamma Correction (GC) and anisotropic diffusion algorithms (ADA) are applied to enhance the contrast and to reduce the noise.

- ***Gamma correction for contrast enhancement***

GC is an additional nonlinear intensity modification function utilized for contrast manipulation. GCis defined by the Eq. (1).

$$G_c = C.R^{\chi} \tag{1}$$

Let, C be the constant, R be thenon-negative intensity of the input pixels, G_c be the resulting pixel intensity, and χ be the power factor designed to provide a series of curves that highlight various intensity zones.

- ***Anisotropic Diffusion for noise reduction [30]***

The time development of an image d, which has been processed by a nonlinear anisotropic diffusion scheme is governed by the following partial differential Eq. (2).

$$d_t = div\left(S.\nabla d\right) \quad (2)$$

Let, d_t be the a variation on the picture d to the processing duration t, ∇d be thegradient in connection with the coordinate vector $z\left(\dim\left(n,1\right)\right)$, n equals the image's space's dimension, and S be the square matrix, $\dim\left(n,n\right)$ is denoted as diffusivity. They define the nature of the diffusion process in the development aspect. The isotropic process produces an intraregional smoothing between the well-preserved edges and prevents the general blurring of the linear diffusion. The modification to the diffusion equation is described in eq. (3) and (4) below.

$$d_t = div\left(s\left(\left|\nabla d\right)\nabla d\right) \quad (3)$$

With

$$s\left(\nabla d\right) = \frac{1}{1+\left|\nabla d\right|^2 / u^2} \quad (4)$$

Let, u be the user-defined parameter. The enhanced image is updated by the normalization process to standardize the data.

4.1.3. Normalization
The normalized pixel values fall between 0 and 1 using Min-max normalization.

Min-max normalization: Min-max normalization is an important method of scaling the data so that the range of values of a feature falls within a particular range. It scales the values within a certain range of [0, 1] and alters features. Finally, normalize the data using Eq. (5).

$$M' = \frac{M - M_{\min}}{M_{\max} - M_{\min}} \quad (5)$$

Let, M be the enhanced image, $M_{\max}$ be the maximum value, and $M_{\min}$ be the minimum value of the enhanced image. The sample preprocessed and segmented images are shown in Figure.2 below.

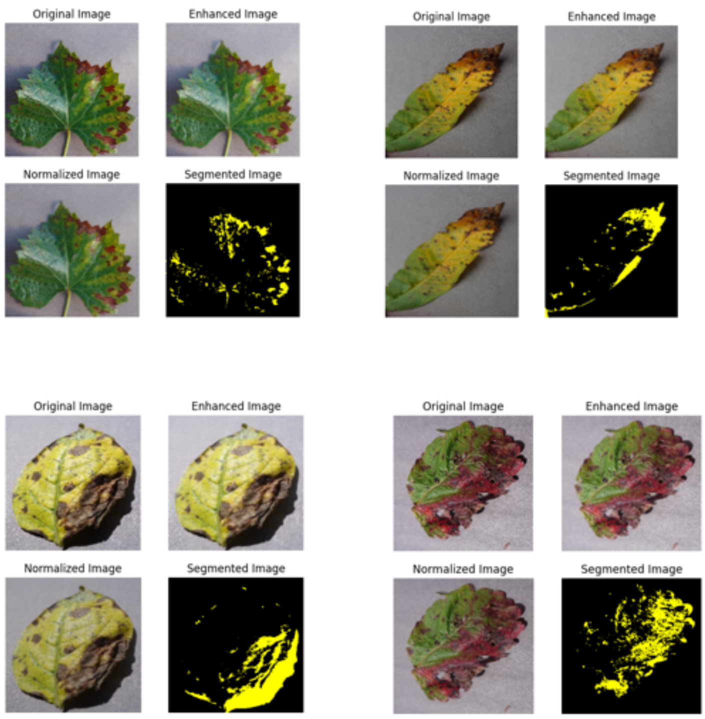

Figure.2: Sample preprocessed and segmented image results

4.2 Segmentation using DeepLabV3++E-AHC

4.2.1 DeepLabV3++ with Enhanced Attention and Hybrid Convolutions (DeepLabV3++E-AHC)

Use DeepLabV3 with ASPP and SRA and Resnet (ResNet backbone with SE blocks and hybrid dilated convolutions) and Transposed Convolutional Layers (in the upsampling) to segment the complex patterns of leaves.

- ***Modified ResNet Backbone***

The ResNet architecture is employed with modifications such as SE (Squeeze-and-Excitation) blocks and hybrid dilated convolutions to improve the feature representation and context awareness.

Squeeze-and-Excitation (SE) Blocks: SE blocks dynamically adjust the channel-wise feature responses. The recalibration is conducted using Eqs. (6,7).

$$y = \text{glb}\, avgpool\left(z\right) \tag{6}$$

$$k = \sigma\left(w\left(t2\right).\text{Re}lu\left(w\left(t1\right).y\right)\right) \tag{7}$$

Let, z be the input feature map, $w\left(t1\right)$and $w\left(t2\right)$ be the learnable weights, σbe the sigmoid activation function, and k be the weight of recalibration for every channel.Equation (8) displays the output after recalibration.

$$z' = k.z \tag{8}$$

- ***DeepLabV3++with ASPP***

Atrous Spatial Pyramid Pooling (ASPP): ASPP applies dilated convolution to extract multi-scale features. The dilated convolution applies convolutional filters at different dilation rates to learn features on different scales. It is useful in analyzing different spatial hierarchies and enhances the segmentation process. The convolution operation for a dilated convolution is expressed in Eq. (9).

$$x\left(i\right) = \sum_{s} z\left(i + d.s\right).c\left(s\right) \tag{9}$$

Let, $x\left(i\right)$be the output feature situated at i, z be the feature map input, c be the convolutional kernel, d be the dilation rate. The various dilation rates are used in ASPP to capture characteristics at various scales.

- ***Transposed Convolutional Layers***

*Upsampling: T*ranspose of convolutions (deconvolutions) is applied to expand the size of feature maps in space. The output x for a transposed convolution using Eq. (10).

$$x\left(i\right) = \sum_{j} z\left(i - j\right).c\left(j\right) \tag{10}$$

Let, $x\left(i\right)$be the output at the position i, z be the input feature map, c be the transposed convolution kernel. It is used to increase the size of the segmented output to match the size of the input for better pixel-level segmentation. With the help of these mathematical principles, the DeepLabV3++E-AHC model uses multi-scale feature extraction, attention mechanisms, and accurate upsampling to segment complex leaf patterns accurately.

- ***Stretch-Relax Attention***

Attention Mechanism: The attention mechanism works on the feature maps and emphasizes the relevant areas of feature maps. The attention coefficients are calculated depending on the feature maps in theStretch-Relax Attention. The attention weight $\vec{\alpha}$ is calculated by eq. (11).

$$\vec{\alpha}_i = \frac{\exp\left(\vec{f}_i\right)}{\sum_j \exp\left(\vec{f}_j\right)} \tag{11}$$

Let, $\vec{f}_i$ be the attribute at the given location i, and $\vec{\alpha}_i$ be the position's weight of attention i.Equation (12) expresses the attention mechanism's output.

$$A_o = \sum_i \vec{\alpha}_i.\vec{z}_i \tag{12}$$

Let, $\vec{z}_i$be the feature at position i.This is supposed to help increase the importance of the more informative features while decreasing the importance of the less informative features. This can enhance the model's performance in the intricate patterns by directing it to the important areas of the leaf structures. Figure.3: shows Segmentation using DeepLabV3++E-AHC

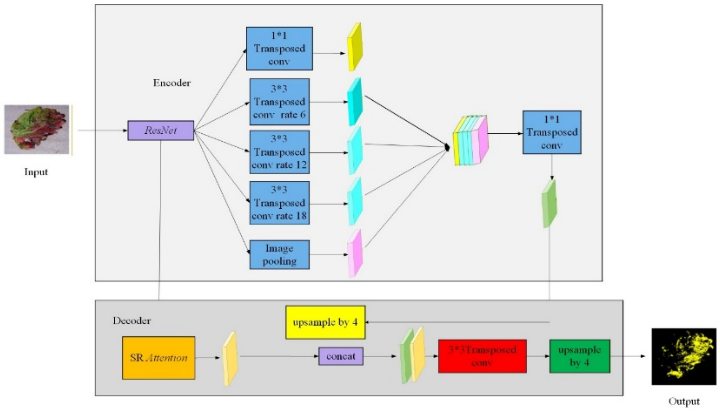

Figure.3: Segmentation using DeepLabV3++E-AHC

4.2.1 Evaluation

Dice similarity coefficient (DSC) and Intersection over Union (IoU) are employed to evaluate the precision and segmentation quality of the model in capturing the detailed structures of leaves.

- ***Dice Similarity Coefficient (DSC)***

Calculate the similarity between the segmentation obtained using the proposed method and the actual segmentation. It is calculated in Eq. (13).

$$dsc = \frac{2.|\overline{p} \cap \overline{g}|}{|\overline{p}| + |\overline{g}|} \quad (13)$$

Let, $\overline{p}$ and $\overline{g}$ be the sets of pixels in the segmentations based on ground truth and prediction, $|\overline{p} \cap \overline{g}|$ be the number of overlapping pixels, and $|\overline{p}|$, and $|\overline{g}|$ be the amount of pixels in the segmentations according to prediction and ground truth.

- ***Intersection over Union (IoU)***

The ratio of the overlap of the predicted and the ground truth segmentations is calculated as the union of the two. It is calculated as Eq. (14).

$$iou = \frac{|\overline{p} \cap \overline{g}|}{|\overline{p} \cup \overline{g}|} \quad (14)$$

Let, $|\overline{p} \cup \overline{g}|$ be the total quantity of pixels in the segmentations derived from the ground truth and the prediction. This combination of techniques is ideal for segmentation problems such as segmentation of the leaf pattern which requires both the small details and the overall context. The evaluation metrics (DSC and IoU) assist in evaluating the effectiveness of the model and its capacity to depict leaf features.

4.3 Feature Extraction

In the feature extraction task, extraction of many features from the segmented results are done to attain good data representation. Below is a thorough explanation of every feature extraction technique:

4.3.1. DenseNet

Based on the proposed idea of dense connectivity, DenseNet is a deep learning architecture. Every layer gets feed-forward from all the previous layers and feeds the feature maps to all the following layers. It enhances the feature reuse and gradient flow since this model has a denser connectivity pattern than the previous one. DenseNet is used to extract high-level features of images. The high density of the connections is beneficial in identifying intricate patterns and features which is useful in tasks that involve contextual analysis. Specific layers output from the penultimate layer are used as features for further analysis. The architecture of the DenseNet is shown in Figure.4 below.

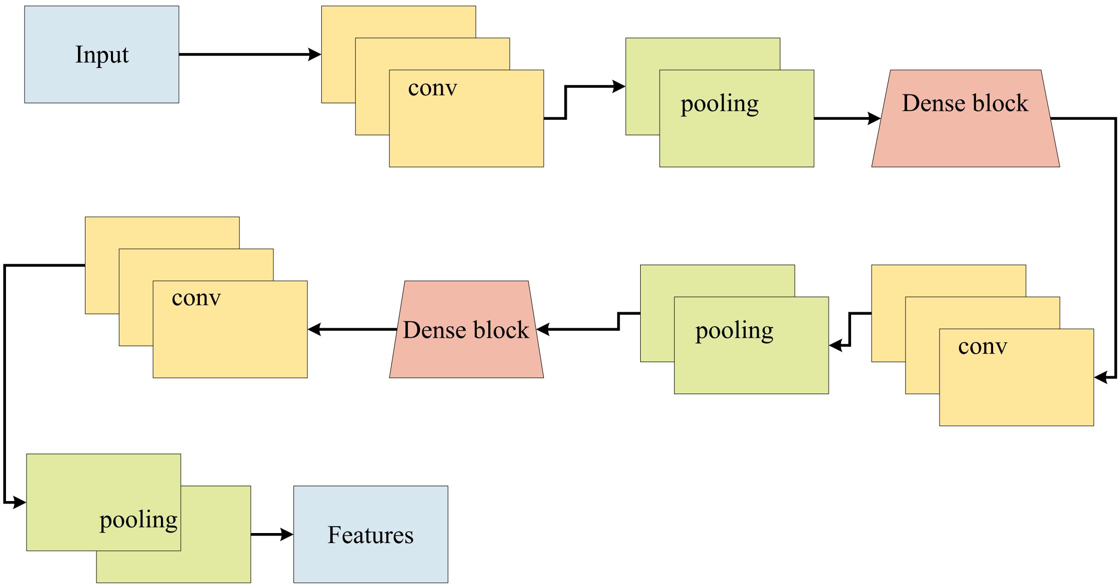

Figure.4: DenseNet Architecture

4.3.2 Texture Analysis

- ***Run-Length Matrix (RLM) Features:***

RLM determines the sizes of the regions of the same intensity in horizontal, vertical, and diagonal directions [31]. SRE and LRE are measures of the proportion of short and long runs in the image. The GLRLM statistics, namely the Gray Level RLM, gives information about the coarseness and regularity of the texture. It is measured using eq. (15).

$$glrlm(a,b)=\sum_{k}C(a,b,k) \tag{15}$$

Let, $C(a,b,k)$ be the number of runs of length and , k be the direction specified by a and b.

- ***Gray Level Co-occurrence Matrix (GLCM) Features:***

The GLCM [32] describes the pixel intensities' spatial dispersion to obtain the texture features. Sobel computes the difference between a pixel and its neighbor and emphasizes the changes in the texture. Correlation measures the extent of the linear relationship between the pixel and the neighboring pixel intensities and strengthens the patterns present in the image. Energy measures the total of squared elements in the GLCM, a measure of textural uniformity; the higher the energy, the more homogeneous the region. Angular second moment measures the degree to which the elements of the GLCM are distributed along the diagonal, which shows pixel pairs with similar intensities and high values representing smooth texture**.**

4.3.3 Color Features

- ***Color Correlogram:***

Color Correlogram estimates the probability density of color co-occurrence in an image. It gives the histogram of color pairs at various distances and thus shows the distribution of colors. It is measured using Eq. (16).

$$Co(\overline{c}_1,\overline{c}_2,D)=\frac{C(\overline{c}_1,\overline{c}_2,D)}{T_p} \tag{16}$$

Let, $\overline{c}_1$ and $\overline{c}_2$ be the color values, Co be the color correlogram, T_p be the threshold value of the pixel image, D be the distance between pairs, and $C(\overline{c}_1,\overline{c}_2,D)$ be the number of color pairs separated in distance D between colors $\overline{c}_1$ and $\overline{c}_2$.

- ***Color Contrast:***

sColor Contrast measures the difference in color intensity or kind between different regions or pixels. It helps distinguish objects based on their color differences using Eq. (17).

$$CC=\sqrt{\sum_{i,j}\left(CI(a,b)-\overline{CI}\right)^2} \tag{17}$$

where $CI(a,b)$ is the color intensity at the pixel (a,b) and $\overline{CI}$ is the average color intensity.

4.3.4 Shape Features

$$cnv = \frac{A(cnv\_hull)}{A(s)}$$

Let, cnv be the convexity, $A(cnv\_hull)$ be the area of the convex hull, and $A(s)$ be the area of shape.

$$S_d = \frac{A(s)}{A(cnv\_hull)}$$

4.3.5 Region-Based Features

$$MI = \frac{1}{n}\sum_{i=1}^{n}\tilde{I}_i$$

Let, $\tilde{I}_i$ be the intensity of the pixel i and n be thenumber of pixels within the area.

$$SD = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\tilde{I}_i - \tilde{I}^*\right)^2}$$

Let, $\tilde{I}^*$ be the mean intensity. When these different features are integrated, it offers a complete representation of the data making it easier to analyze and classify data for tasks such as image segmentation, object detection, and pattern recognition.

4.4. Feature Selection using Proposed SRHO Model

Combination of Starling Murmuration Optimizer (SMO) [33] algorithm and Artificial Hummingbird Algorithm (AHA) [34].

Optimising systems is the focus of the swarm-based metaheuristic known as the AHA method. Three numerical

$$u_i = lw + R.(up - lw)$$

Let, lw and up be the lower and upper bounds, R be the vector at random in [0, 1], and u_i be the i^{th} food source

$$vt_{ij} = \begin{Bmatrix} 0 & if\,(i \neq j) \\ 1 & i = j \end{Bmatrix}$$

- ***Convex Hull Features:***

The smallest convex border surrounding a shape is its convex hull. Features consist of:

Convexity: Quantifies the extent to which a shape deviates beyond convexity using Eq. (18).

(18)

Solidity: Determines the percentage of the shape's surface that the convex hull occupies using Eq. (19).

(19)

Mean Intensity: The mean pixel level of intensity in a given area is measured using Eq. (20).

(20)

Standard Deviation of Intensity: Quantifies the variation in intensity levels within a given area using Eq. (21).

(21)

models representing the three different hummingbird foraging behaviors, territorial foraging, directed foraging, and migrating foraging are shown in this subsection. The best characteristics from the retrieved features are identified and chosen using these foraging behaviors.

4.4.1 Process of SRHO model

A group of h A hummingbird is perched on h food sources, and started at random in the manner using Eq. (22).

(22)

position. The initialization of the food source visit table is processed using Eq. (23).

(23)

Let, $i = j$, $vt_{ij} = 1$ shows that a hummingbird is consuming food at its designated feeding location. for $(i \neq j)$, $vt_{ij} = 0$ shows that the j^{th} food source and is stayed by the i^{th} hummingbird during this iteration.

4.4.2 Guided foraging

Because hummingbirds are naturally inclined to the food source with the greatest amount of nectar, a target supply needs an extensive amount of replenishment and a long interval between visits until that specific hummingbird visits it. Therefore, hummingbirds guided foraging habits and locate the food sources with the highest visit level and choose as their target food source the ones at which they obtain the greatest frequency of nectar refills. After determining where to feed, the hummingbird takes off in the direction of its target. All birds can fly, however only hummingbirds are skilled in axial and diagonal flight. These patterns of flight can be transferred to a d-D space, where Eq. (24) defines axial flight and Eq. (25) defines diagonal flight.

$$d_i = \begin{cases} 1 & if\left(i = rand_i\left(\left[1, D\right]\right)\right) \\ 0 & else \end{cases} \tag{24}$$

$$d_i = \begin{cases} 1 & if\left(i = p_j\right) \\ 0 & else \end{cases} \tag{25}$$

Let, $rand_i\left(\left[1, D\right]\right)$produces a random number between 1 to D, p_j generates an arbitrary integer permutation from 1 to k. A hummingbird uses its flight capabilities to reach its target food source, which leads to the acquisition of a candidate food supply. Consequently, a food source is updated regarding the target food supply selected from all available sources. The mathematical formula that represents the targeted foraging behavior and a possible food source is derived in equation (26):

$$v_i\left(t+1\right) = u_{i,tar}\left(t\right) + A.D.\left(u_{i,tar}\left(t\right)\right) \tag{26}$$

Let, $v_i\left(t+1\right)$be the i^{th} food source position at the time t, $u_{i,tar}\left(t\right)$be the location of the intended food supply and Abe the guided factor, which falls between 0 and 1 from the normal distribution and D be the distance. Eq. (26) represents humming bird-guided foraging through varying flight patterns and allows every current food source to adjust its position about the destination food source. The updated i^{th} food source position is determined in Eq. (27).

$$u_i\left(t+1\right) = \begin{cases} u_i\left(t\right) & F\left(u_i\left(t\right) \le F\left(v_i\left(t+1\right)\right)\right) \\ v_i\left(t+1\right) & F\left(u_i\left(t\right) > F\left(v_i\left(t+1\right)\right)\right) \end{cases} \tag{27}$$

Let, $F\left(.\right)$be the fitness function value.

4.4.3 Territorial foraging

A hummingbird is probably going for a novel food source rather than visiting different sources of food following the visit of its target food source and consuming the flower nectar. As a result, a hummingbird can easily go to a nearby area within its range, where it might discover a fresh food supply or a potential improvement over the one it already has. In this phase, update the cohesion operator $\zeta_i\left(t\right)$from the whirling search strategy of SMO to enhance the easy finding of a new food source process. Eq. (28) provides the mathematical equation that represents the hummingbirds' local search for a potential food source as part of their territorial foraging approach.

$$v_i\left(t+1\right) = u_i\left(t\right) + \zeta_i\left(t\right) + B.D.\left(u_i\left(t\right)\right) \tag{28}$$

Let, Bbe the territorial factor, it falls within the normal distribution (0,1). $\zeta_i\left(t\right)$ be the cohesion operator which is measured using eq. (29).

$$\zeta_i\left(t\right) = \cos\left(c\left(t\right)\right) \tag{29}$$

Let, $c\left(t\right)$be the random number in the range of 0 to 1. Equation (29) makes it simple for every hummingbird to use its unique flight abilities to locate a new food source in the instant neighbourhood based on its position. The visit table needs to be modified by following the implementation of the territorial foraging method.

4.4.4 Migration foraging

When there is not enough food in the region a hummingbird visits, it may frequently move to a deeper food source to dine. The migration coefficient is defined by the AHA technique. A hummingbird forages from the source that replenishes nectar at the slowest rate possible to a randomly produced new source, as described by Equation (30):

$$u_{wor}\left(t+1\right) = lw + R.\left(up - lw\right) \tag{30}$$

Let, u_{wor} be the food source. AHA is processed based on the foraging habits of hummingbirds. It combines three essential foraging behaviors: guided foraging, migrating foraging, and territorial foraging. The first step is to initialize a set of food sources based on the random. During guided foraging, hummingbirds use axial, diagonal, and omni directional flight patterns to update their positions while they search for food sources with high nectar replenishment rates. Territorial foraging is searching locally for better food supplies and facilitating exploration using SMO cohesion operators. When existing sources are exhausted, migratory foraging uses a migration coefficient to guide hummingbirds to new locations. By imitating these foraging habits, this technique effectively selects features to optimize and refine the search process.

Algorithm: Feature Selection using SRHO	
Input:	Extracted feature set
Output:	Optimal feature subset
Step 1:	Initialize the hummingbird population and generate food source positions using Eq. (22).
Step 2:	Initialize the Visit Table for food sources as defined in Eq. (23).
Step 3:	Evaluate the fitness of each food source and record the best one as the global best.
Step 4:	Perform guided foraging toward the target food source using axial/diagonal flight in Eq. (24) and Eq. (25), and update the position using Eq. (26).
Step 5:	Replace the food source if the new candidate gives higher fitness according to Eq. (27).
Step 6:	Apply territorial foraging with local search using the SMO cohesion operator in Eq. (28) and Eq. (29).
Step 7:	Relocate the worst food source to a new random search region using Eq. (30).
Step 8:	Update the global best feature subset until convergence and return the final optimal subset.

4.5 Disease Detection via LeafNet+Components of LeafNet+

To improve performance, create a new LeafNet+ architecture in this phase and incorporate several cutting-edge components.:

Input Images: Raw input images are used to begin the procedure.

Convolutional Layers (conv2D): Convolution techniques are used by these layers to retrieve the images' basic features. They are essential for recognizing patterns like textures and borders.

Batch Normalization and ReLU Activation: The output of the Conv 2D is subjected to batch normalization with a momentum of 0.3, which normalizes the layer's inputs and speeds up training. The network learns intricate patterns by introducing non-linearity through the application of the ReLU activation function.

Max Pooling: This layer keeps the most noticeable features while shrinking the feature maps' spatial dimensions. It facilitates the model's invariance to slight input translations.

DenseNet Block: This block establishes feed-forward connections between every layer and every other tier. Deeper networks operate better as a result of enhanced feature propagation and a decreased vanishing gradient issue.

Squeeze and Excitation Block: This block recalibrates feature responses at the channel level by simulating interdependencies between channels. It improves the network's capacity to concentrate on the most educational elements.

Residual Block: Residual blocks allow the network to learn residual functions by reducing the vanishing gradient issue. It aids in the training of deeper networks. As a result, the network can learn complicated tasks more successfully.

Inception Modules: These components capture multi-scale features by using parallel convolutions of different sizes. As a result, the network may learn characteristics at different scales and perform better overall.

Self-Attention Block: It enables the network to assign varying weights to various regions of the input image. By concentrating attention on crucial areas and capturing dependencies throughout the entire image.

Global Average Pooling: Important features are preserved and the dimensionality of the feature maps is decreased in this layer. It helps to lessen overfitting and strengthens the model's resistance to spatial translations.

Fully Connected Layers (FC1 and FC2): Based on the features that the preceding layers gathered and processed, these last layers function as classifiers, generating the final classification results.

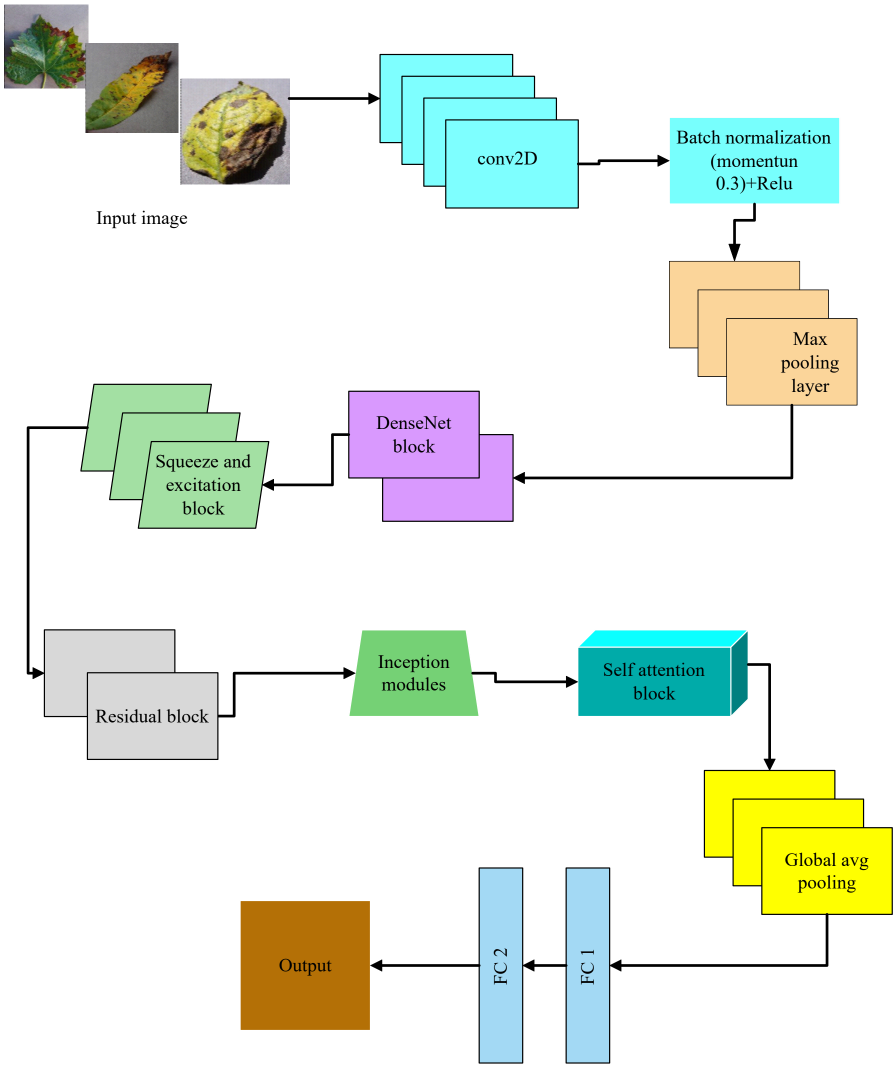

Figure.5: Architecture of LeafNet+ model

Figure.5 diagram illustrates the architecture of deep learning that is suitable for image classification. It starts with the input images and these are followed by several convolutional layers referred to as conv2D. These layers convolve the input images to get features from the images fed into the network. The output from the convolutional layers is followed by batch normalization using a momentum of 0. 3 and then the ReLU activation function which adds non-linearity to the network.Subsequently, the normalized outputs are passed through a max pooling layer to decrease the spatial size while preserving the most significant characteristics. The DenseNet block directly connects each layer to all other layers in a feed-forward manner to enhance feature propagation and reduce the vanishing gradient issue. After passing through the DenseNet block, the data is passed through a Squeeze and Excitation block where the channel-wise feature responses are re-weighted based on the interdependencies between the channels. Due to the vanishing gradient issue, deeper networks are trained more easily through the inclusion of a residual block in the design, which allows the framework to learn the residual functions. By implementing convolutions of different sizes, the inception modules in the design enable the network to record features at many scales. After that, the network has a self-attention block that helps the network to pay more attention to some parts of the input, which improves the model's capacity to capture dependencies across the image. The processed data is then subjected to global average pooling, which downsamples the dimensions, but retains the key features and enhances the model's translation invariance. The last layers include two fully connected (FC) layers, which are FC1 and FC2 that serve the purpose of classifiers. The output of these layers is the final classification result to predict crop disease. This all-embracing architecture integrates several sophisticated deep-learning parts to deliver high performance in image categorization.

4.6 Insecticide Recommendation using PPO-Based Reinforcement Learning

The RL [35] model is applied to optimize insecticide recommendation policies on the basis of detection of disease and environmental factors. The deep learning model will identify the crop disease and then this crop disease will act as the primary input to the RL agent. The agent keeps on interacting with the agricultural environment and prescribes insecticides based on the identified leaf illness, the intensity of the disease, and the weather. The essence of it is to choose the most productive insecticide treatment, which provides maximum crop protection while minimising cost and environmental harm. S is the state space of the leaf disease detection results and other ecological data (e.g. humidity, temperature and disease severity index). Figure 6 shows the PPO-based insecticide recommendation system.

The action space A will be made up of the available insecticide options and application settings that include dosage, type, and timing. The reward behaviour quantifies the effects of every suggestion in a proportional manner, balancing the dangers of illness, the cost of cure, and sustainability, and penalising decisions that lead to environmental risks or unproductive chemical consumption. PPO is a technique that is used to train the RL agent to stabilize the learning process and provide efficient convergence. PPO optimizes the clipped surrogate to ensure that policy updates are not very large and, therefore, learning is safer and more reliable.

Mathematical PPO formulation is as follows:

$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \operatorname{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \quad (31)$$

where

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \quad (32)$$

π_θ represents current policy, $\pi_{\theta_{\text{old}}}$ represents the past policy, $\hat{A}_t$ is the estimated utility of the policy, and ϵ is the clipping coefficient that limits aggressive policy changes.

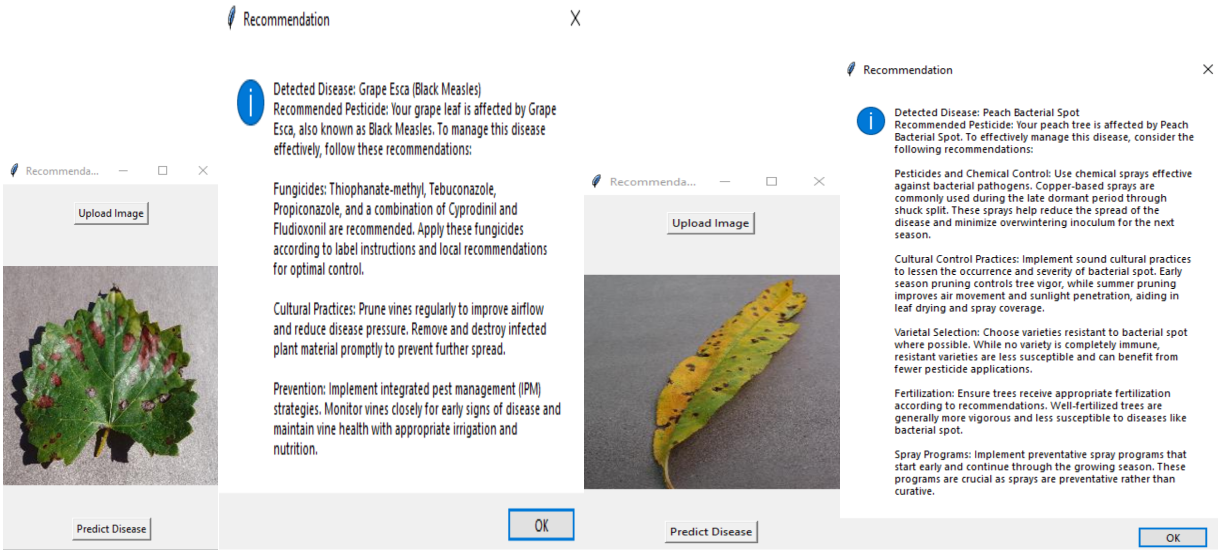

Figure.6: Sample images with recommended results

After training, the PPO-RA model is incorporated into the decision support system to give real-time advice. The trained agent independently proposes the type, dosage and the time of application of insecticides that have the maximum effect in preventing plant disease, yet at the same time sustaining economic viability and environmental accountability. Recommended insecticides output relative to leaf disease samples is depicted in Figure 6.

Algorithm: PPO-RA - Insecticide Recommendation	
Input:	Disease detection results and environmental parameters.
Output:	Recommended insecticide type, dosage, and timing.
Step 1:	The disease type and environmental factors are encoded as the state s_t and given to the PPO agent.
Step 2:	The agent selects an action a_t corresponding to insecticide type, dosage, and timing using the current policy.
Step 3:	A reward is calculated based on disease control effectiveness, cost, and environmental impact.
Step 4:	The policy is updated using the PPO clipped objective in Eq. (31) and probability ratio in Eq. (32) to stabilize learning.
Step 5:	The process continues until convergence, after which the trained model generates real-time insecticide recommendations.

5. RESULTS AND DISCUSSION

The created model has been used with the Python tool, and its effectiveness is confirmed by comparing it to other existing DL techniques in terms of Jaccard score, accuracy, hamming loss, recall, and other metrics. Five different plant leaf diseases are included in the data set, and they are tested and trained using two different learning rates: 70 and 80. Figure. 7 displays the two learning rates' confusion matrices.

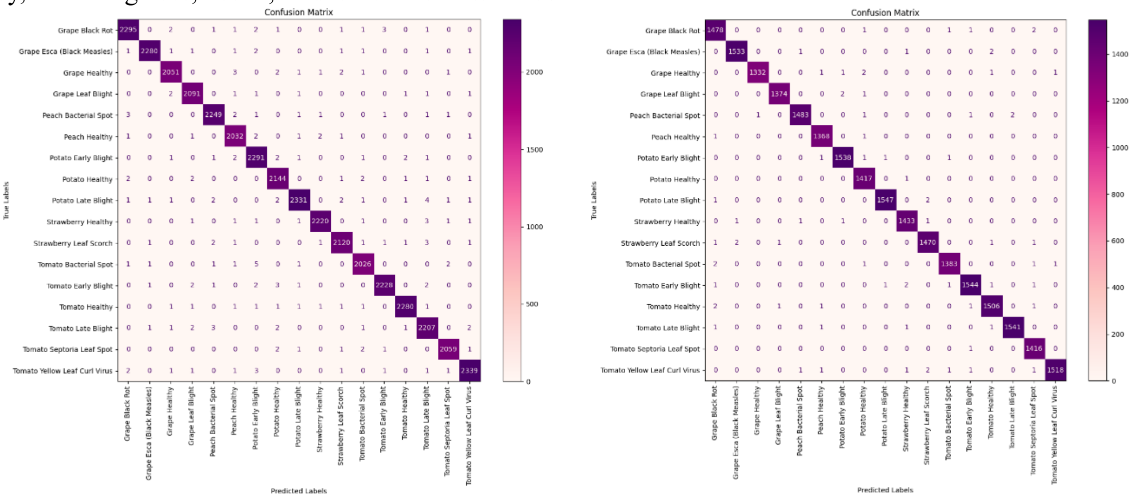

Figure.7: Confusion matrices of two learning rates

5.1 Dataset description

The Updated Plant Village Dataset [36] has been employed in the Python system. This extensive collection includes nine different species of plant leaves, both healthy and injured. This collection consists of over 70,000 high-quality images of both healthy and diseased leaves of plants from nine distinct species. There are three data divisions (train, test, and validation) with identical categories for each split. The diversity of the information, which encompasses a large variety of plant species, illnesses, and growth phases, makes it exceptional.

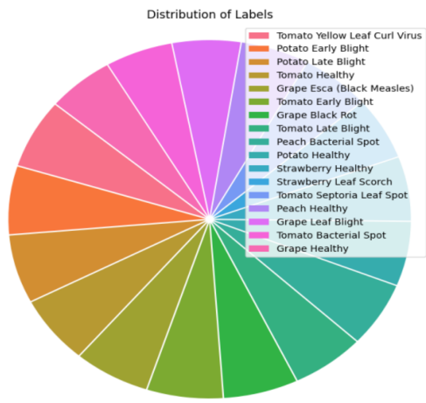

Figure.8: Labels distribution

Five species are selected from the dataset for this study, including potatoes, tomatoes, grapes, strawberries, and peaches. Figure. 8 displays the label distribution of the dataset that is used.

5.2 Performance analysis

Accuracy, precision, recall, Mathew's correlation coefficient (MCC), F1-score, R2-score, Jaccard score, Cohen's kappa, and Hamming loss are the performance metrics that are utilized to assess the constructed model efficiency. Popular DL approaches like MobileNet-v2 [34], Inception-v3 [35], U-Net [36], and DenseNet-121 [37] are used to validate these measures. The constructed model efficiency is shown in Table 2 and Table 3 and is validated using two learning rates 70 and 80 in the experimental comparative results. The graphical representation of the created model using current methods is shown in Figure. 9 to 17.

Table 2: Comparison of performance with 70 learning rates

Performance metrics	MobileNet-v2	Inception-v3	U-Net	DenseNet-121	LeafNet+ (Proposed)
Accuracy	0.982876	0.977746	0.967675	0.975262	0.994951
Precision	0.982814	0.977764	0.967661	0.975212	0.994932
Recall	0.982917	0.977753	0.967683	0.975229	0.994962
F1 Score	0.982862	0.977753	0.967666	0.975216	0.994946
R2 Score	0.963874	0.952769	0.931276	0.950403	0.99034
MCC	0.981803	0.976353	0.96565	0.973712	0.994635
Cohen's Kappa	0.981803	0.976352	0.965649	0.973712	0.994634
Hamming Loss	0.017124	0.022254	0.032325	0.024738	0.005049
Jaccard Score	0.966307	0.956481	0.937371	0.951641	0.989945

Table 3: Comparison of performance with 80 learning rates

Performance metrics	MobileNet-v2	Inception-v3	U-Net	DenseNet-121	LeafNet+ (proposed)
Accuracy	0.986576	0.983731	0.969265	0.979523	0.997035
Precision	0.986548	0.983703	0.969249	0.979593	0.997027
Recall	0.986591	0.983736	0.969217	0.979532	0.997049
F1 Score	0.986565	0.983715	0.969219	0.979553	0.997037
R2 Score	0.972886	0.967429	0.93515	0.956561	0.993113
MCC	0.985735	0.982712	0.967341	0.978241	0.996849
Cohen's Kappa	0.985735	0.982711	0.967339	0.97824	0.996849
Hamming Loss	0.013424	0.016269	0.030735	0.020477	0.002965
Jaccard Score	0.973493	0.967959	0.940287	0.959937	0.994092

The LeafNet+ model had the highest accuracy scores. Concerning MobileNet-v2 (0.982876), Inception-v3 (0.977746), U-Net (0.967675), and DenseNet-121 (0.975262), it achieved a much higher accuracy of 0.994951 in 70 learning rates. In 80 learning rates, the accuracy of the suggested model increased to 0.997035, outperforming DenseNet-121 (0.979523), Inception-v3 (0.983731), U-Net (0.969265), and MobileNet-v2 (0.986576).

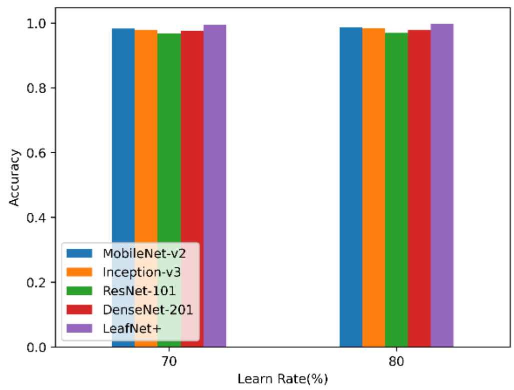

Figure.9: Accuracy comparison

The LeafNet+ model performed better in terms of precision. It outperformes MobileNet-v2 (0.982814), Inception-v3 (0.977764), U-Net (0.967661), and DenseNet-121 (0.975212) with a precision of 0.994932 in learning rate 70. The suggested approach outperformed MobileNet-v2 (0.986548), Inception-v3 (0.983703), U-Net (0.969249), and DenseNet-121 (0.979593) in terms of precision, for 80 learning rates. Figure.10: shows that Precision comparison

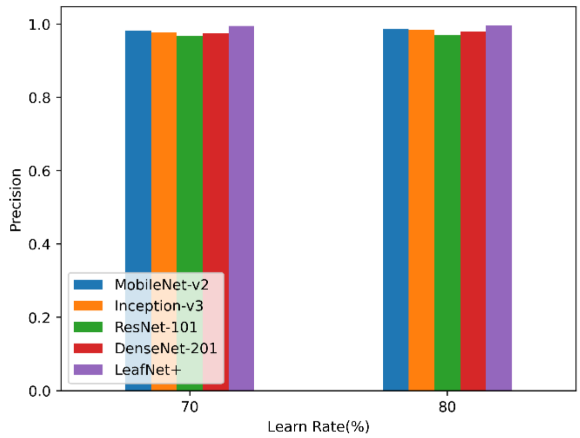

Figure.10: Precision comparison

The efficiency of the LeafNet+ model is established by measuring recall. Compared to MobileNet-v2 (0.982917), Inception-v3 (0.977753), U-Net (0.967683), and DenseNet-121 (0.975229), it yielded a recall of 0.994962 in learning rate 70. The suggested model's recall, at 0.997049, is even greater than that of MobileNet-v2 (0.986591), Inception-v3 (0.983736), U-Net (0.969217), and DenseNet-121 (0.979532) for 80 learning rates. Figure.11: shows that Recall comparison

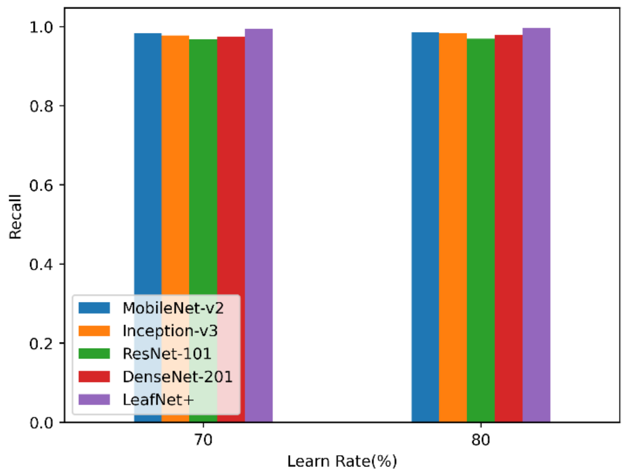

Figure.11: Recall comparison

TheLeafNet+ model outperformed MobileNet-v2 (0.982862), Inception-v3 (0.977753), U-Net (0.967666), and DenseNet-121 (0.975216) with an F1 Score of 0.994946 in 70 learning rates. The suggested model scored 0.997037, higher than the scores of Inception-v3 (0.983715), U-Net (0.969219), DenseNet-121 (0.979553), MobileNet-v2 (0.986565), and Inception-v3 (0.983715) for 80 learning rates. Figure.12: shows that F1-score comparison

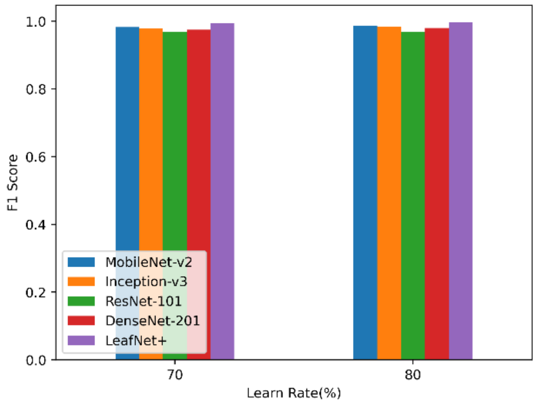

Figure.12: F1-score comparison

The LeafNet+ model has the highest R2 Score values, which indicate the percentage of variation explained by the framework. With a score of 0.99034 in 70 learning rate, it outperformed MobileNet-v2 (0.963874), Inception-v3 (0.952769), U-Net (0.931276), and DenseNet-121 (0.950403) by a significant margin. The suggested model's R2 Score, for 80 learning rates, is 0.993113, which is higher than that of MobileNet-v2 (0.972886), Inception-v3 (0.967429), U-Net (0.93515), and DenseNet-121 (0.956561). Figure.13: shows that R2-score comparison

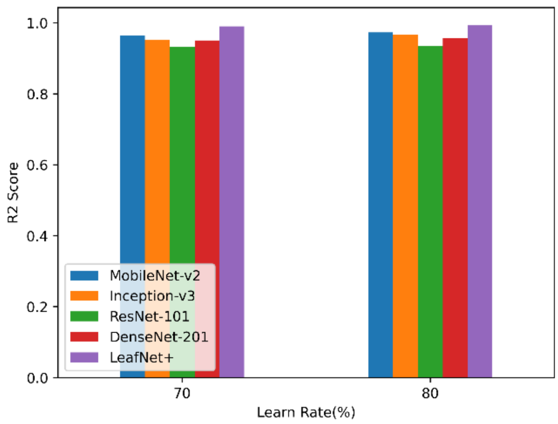

Figure.13: R2-score comparison

Another measure that emphasizes the success of the LeafNet+ model is the Matthews Correlation Coefficient (MCC). The suggested model in 70 learning rates outperformed MobileNet-v2 (0.981803), Inception-v3 (0.976353), U-Net (0.96565), and DenseNet-121 (0.973712) with an MCC of 0.994635. As In the 80-learning rate, the suggested model has an MCC of 0.996849, which is higher than that of MobileNet-v2 (0.985735), Inception-v3 (0.982712), U-Net (0.967341), and DenseNet-121 (0.978241). Figure.14: shows that MCC comparison

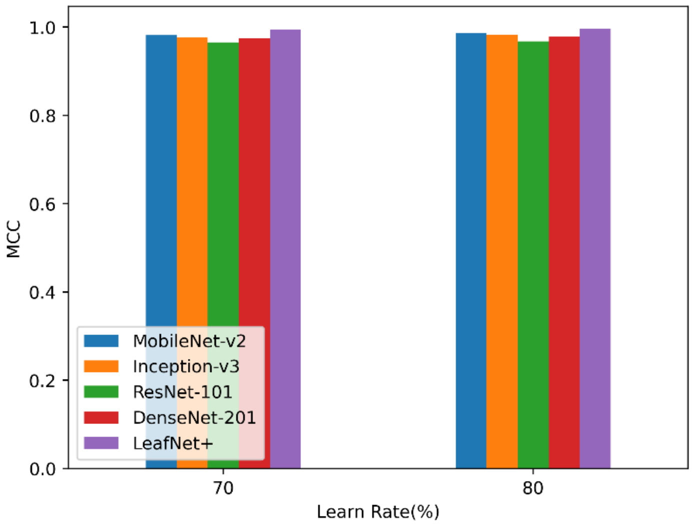

Figure.14: MCC comparison

TheLeafNet+model has the highest Jaccard Score, which calculates how similar the predicted and real labels in 70 learning rates, its score of 0.989945 is greater than that of Inception-v3 (0.956481), U-Net (0.937371), DenseNet-121 (0.951641), and MobileNet-v2 (0.966307). The suggested model outperformed MobileNet-v2 (0.973493), Inception-v3 (0.967959), U-Net (0.940287), and DenseNet-121 (0.959937) with a Jaccard Score of 0.994092 in 80 learning rates.

Figure.15: shows that Jaccard score comparison

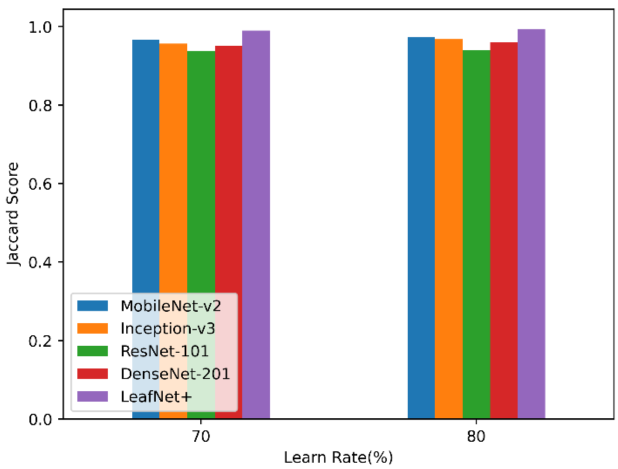

Figure.15: Jaccard score comparison

The LeafNet+model is also the top performer when measured by Cohen's Kappa statistic, which reflects the degree of agreement between the forecasted and actual labels. It outperformed MobileNet-v2 (0.981803), Inception-v3 (0.976352), U-Net (0.965649), and DenseNet-121 (0.973712) with a Kappa score of 0.994634 in 70 learning rates. The suggested model beat MobileNet-v2 (0.985735), Inception-v3 (0.982711), U-Net (0.967339), and DenseNet-121 (0.97824) with a score of 0.996849 in 80 learning rates. Figure.16: shows that Cohen's Kappa comparison

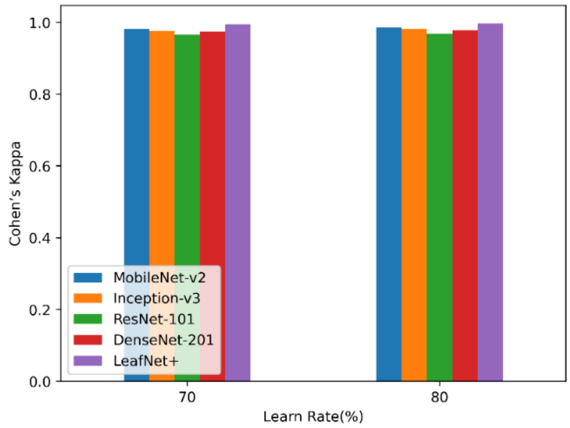

Figure.16: Cohen's Kappa comparison

The LeafNet+ model has the lowest Hamming Loss, an estimation of the average amount of misclassifications per sample. With a Hamming Loss of 0.005049 in 70 learning rates, the suggested model outperformed MobileNet-v2 (0.017124), Inception-v3 (0.022254), U-Net (0.032325), and DenseNet-121 (0.024738) by a large margin. In the 80 learning rates, the suggested model outperformed MobileNet-v2 (0.013424), Inception-v3 (0.016269), U-Net (0.030735), and DenseNet-121 (0.020477) in terms of Hamming Loss, which further dropped to 0.002965. Figure.17: shows that Hamming loss comparison

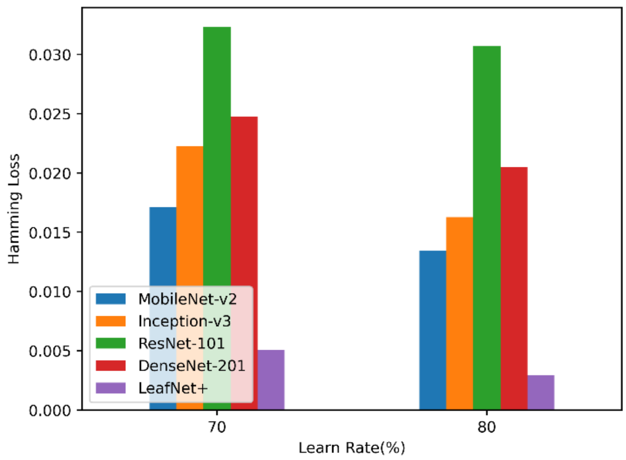

Figure.17: Hamming loss comparison

The suggested model performs better than the other techniques on all criteria in both tables, as can be seen by its persistent superiority.

5.3 Discussion

The LeafNet+ model shows excellent performance in all the parameters with a learning rate of 70 and 80 surpassing MobileNet-v2, Inception-v3, U-Net, and DenseNet-121. It yields the highest accuracy, precision, recall, F1 Score, R2 Score, MCC, and Cohen's Kappa while having the lowest Hamming Loss and the highest Jaccard Score. These findings further support the model's performance in terms of instance classification accuracy, precision-recall trade-off, and misclassification costs. The LeafNet+ model has highefficiency and high reliability, which makes it significantly superior to other models. The classified report of the designed technique with two learning rates is shown in Figure.18 below.

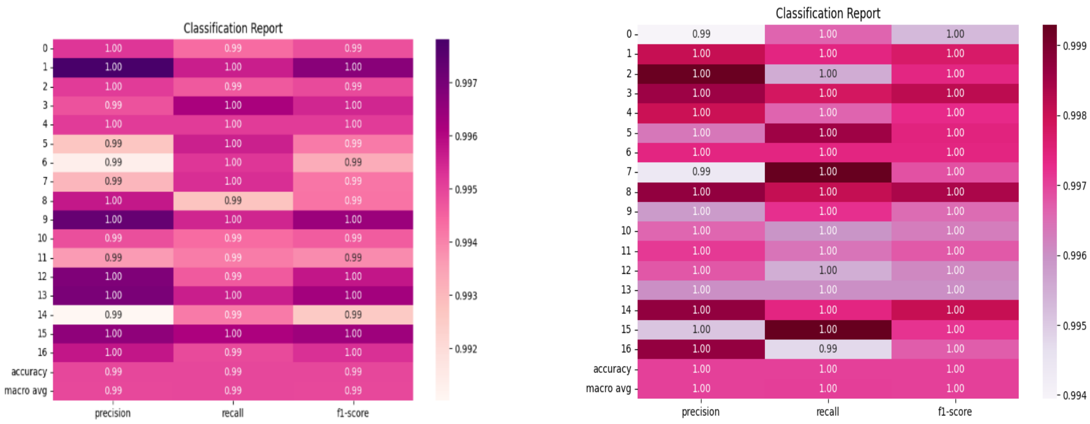

Figure.18: Classified report of 70 and 80 learning rates

The LeafNet+ model shows the highest efficiency and reliability according to the indicators calculated, it can be claimed that this model achieves a high level of accuracy. The accuracy is 0.997035% and the F1 Score is 0. 997037 at a learning rate set at 80, the model shows high efficiency in classifying instances. The precision and recall of the system are 0. 997027 and 0.997049. The accuracy of the LeafNet+ model is evident with the total number of positive instances identified as 0.997049. The overall R2 Score is 0.993113 suggests good ability in capturing data patterns and the low Hamming Loss of 0.002965 reflects minimal misclassifications. Also, the Jaccard Score of 0.994092 and MCC of 0. The model's performance and accuracy are evident in the number 0.996849. Altogether, these metrics confirm the efficiency and reliability of the LeafNet+ model in applied practice.

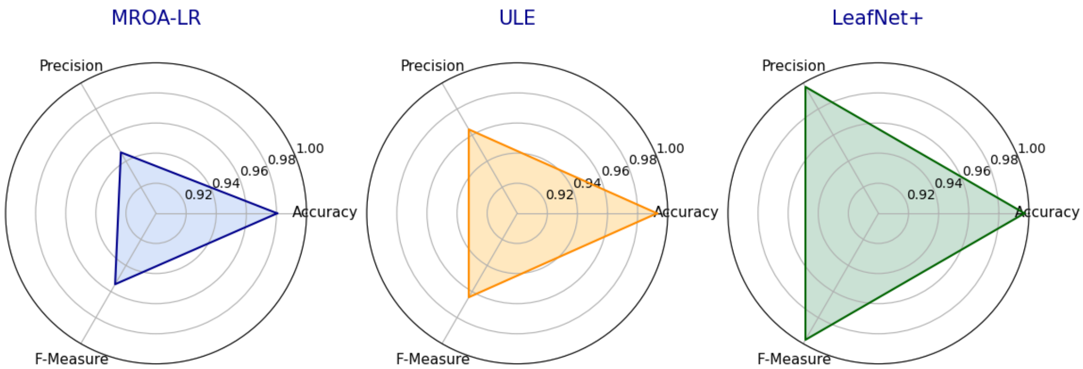

Figure.19: Comparison with first, second, and third papers

The analysis of the three methods shows a clear improvement in the models' performance, shown in Figure.19. The Method 1 (MROA-LR) yields an accuracy of 0. 980512, Specifically, the Precision is 0. 944, Recall is 0. 946754, and F-Measure is 0. 954411. Method 2 (ULE) has better results than these metrics with an accuracy of 0.992879 and a precision of 0. 964397, however, F-Measure remains the same as precision of 0. 964397. Compared to the previous papers, the accuracy of Method 3 (LeafNet+) is significantly higher to 0. 997035. The accuracy of the LeafNet+ method is 99. 70%, Precision is 0. 997027, and F-Measure is 0. 997037. Paper 3 (LeafNet+) gained a significant improvement from the previous methods it balances precision and recall therefore the best and most reliable performance.

6. CONCLUSION

The LeafNet+ system shows that advanced techniques are integrated in a comprehensive way to overcome the existing problem in a better way, that is leaf disease detection and insecticide recommendation. High accuracy and reliability are achieved due to the use of the most advanced preprocessing methods, the segmentation with DeepLabV3++E-AHC, and a diverse feature extraction strategy. Hybrid feature selection models like SRHO are applied to select the most relevant features from the refined set of features. The disease detection system with multi-backbone structures and self-attention mechanisms is accurate in classification and has higher performance. RL improves the system's practical applications and offers flexible insecticide suggestions based on the existing circumstances. The performance is quite impressive on all the evaluated parameters; the accuracy of the model is 0. 997035, and the precision is 0. 997027. The precision achieved is 0. 997049 and the F1 score is 0. 997037. The Hamming Loss is even lower and stood at 0. 002965 and the Jaccard Score increased to 0. 994092 when the learning rate is 80. The LeafNet+ model is more accurate, and precise, and has a higher recall rate than the other models being compared. This integrated approach provides great improvements in agricultural technology to enhance disease control and pest control.

DECLARATIONS

DATA AVAILABILITY STATEMENT

All the data is collected from the simulation reports of the software and tools used by the authors. Authors are working on implementing the same using real world data with appropriate permissions.

FUNDING

No fund received for this project

CONFLICTS OF INTEREST

The authors declare that they have no conflict of interest.

ETHICAL APPROVAL AND HUMAN PARTICIPATION

No ethics approval is required.

REFERENCES

[1] Kaur, P., Mishra, A. M., Goyal, N., Gupta, S. K., Shankar, A., &Viriyasitavat, W. (2024). A novel hybrid CNN methodology for automated leaf disease detection and classification. Expert Systems, e13543.

[2] Das, S., Pattanayak, S., & Behera, P. R. (2022). Application of machine learning: a recent advancement in plant diseases detection. Journal of Plant Protection Research, 122-135.

[3] Lorenzini, G., & Nali, C. (2025). Plant protection, the Cinderella of the one health strategy?. *One Health*, 101080.

[4] Jeger, M., Beresford, R., Bock, C., Brown, N., Fox, A., Newton, A., ... & Yuen, J. (2021). 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, 2(1), 1-18.

[5] Koul, B. (2022). Transgenics and Crop Improvement. In Cisgenics and Transgenics: Strategies for Sustainable Crop Development and Food Security (pp. 131-347). Singapore: Springer Nature Singapore.

[6] Ristaino, J. B., Anderson, P. K., Bebber, D. P., Brauman, K. A., Cunniffe, N. J., Fedoroff, N. V., ... & Wei, Q. (2021). The persistent threat of emerging plant disease pandemics to global food security. Proceedings of the National Academy of Sciences, 118(23), e2022239118.

[7] Wani, J. A., Sharma, S., Muzamil, M., Ahmed, S., Sharma, S., & Singh, S. (2022). Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges. Archives of Computational methods in Engineering, 29(1), 641-677.

[8] ADEKOLA, F. A. (2025). Multidimensional applications and potential health implications of nanocomposites. *Journal of Water and Health*.

[9] Buja, I., Sabella, E., Monteduro, A. G., Chiriacò, M. S., De Bellis, L., Luvisi, A., &Maruccio, G. (2021). Advances in plant disease detection and monitoring: From traditional assays to in-field diagnostics. Sensors, 21(6), 2129.

[10] Moreira, R., Moreira, L. F. R., Munhoz, P. L. A., Lopes, E. A., & Ruas, R. A. A. (2022). AgroLens: A low-cost and green-friendly Smart Farm Architecture to support real-time leaf disease diagnostics. Internet of Things, 19, 100570.

[11] Kyriakarakos, G. (2025). Artificial Intelligence and the Energy Transition. *Sustainability*, *17*(3), 1140.

[12] Abbas, A., Zhang, Z., Zheng, H., Alami, M. M., Alrefaei, A. F., Abbas, Q., ... & Zhou, L. (2023). Drones in plant disease assessment, efficient monitoring, and detection: a way forward to smart agriculture. Agronomy, 13(6), 1524.

[13] Jacquet, F., Jeuffroy, M. H., Jouan, J., Le Cadre, E., Litrico, I., Malausa, T., ... & Huyghe, C. (2022). Pesticide-free agriculture as a new paradigm for research. Agronomy for Sustainable Development, 42(1), 8.

[14] Ali, S., Ullah, M. I., Sajjad, A., Shakeel, Q., & Hussain, A. (2021). Environmental and health effects of pesticide residues. Sustainable Agriculture Reviews 48: Pesticide Occurrence, Analysis and Remediation Vol. 2 Analysis, 311-336.

[15] Jones, R. A. (2021). Global plant virus disease pandemics and epidemics. Plants, 10(2), 233.

[16] Demilie, W. B. (2024). Plant disease detection and classification techniques: a comparative study of the performances. Journal of Big Data, 11(1), 5.

[17] Kotwal, J., Kashyap, R., & Pathan, S. (2023). Agricultural plant diseases identification: From traditional approach to deep learning. Materials Today: Proceedings, 80, 344-356.

[18] Gajjar, R., Gajjar, N., Thakor, V. J., Patel, N. P., & Ruparelia, S. (2022). Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. The Visual Computer, 1-16.

[19] Pechuho, N., Khan, Q., & Kalwar, S. (2020). Cotton crop disease detection using machine learning via tensorflow. Pakistan Journal of Engineering and Technology, 3(2), 126-130.

[20] Khan, A. I., Quadri, S. M. K., Banday, S., & Shah, J. L. (2022). Deep diagnosis: A real-time apple leaf disease detection system based on deep learning. computers and Electronics in Agriculture, 198, 107093.

[21] Storey, G., Meng, Q., & Li, B. (2022). Leaf disease segmentation and detection in apple orchards for precise smart spraying in sustainable agriculture. Sustainability, 14(3), 1458.

[22] Thorat, T., Patle, B. K., & Kashyap, S. K. (2023). Intelligent insecticide and fertilizer recommendation system based on TPF-CNN for smart farming. Smart Agricultural Technology, 3, 100114.

[23] Panchal, A. V., Patel, S. C., Bagyalakshmi, K., Kumar, P., Khan, I. R., & Soni, M. (2023). Image-based plant diseases detection using deep learning. Materials Today: Proceedings, 80, 3500-3506.

[24] Harakannanavar, S. S., Rudagi, J. M., Puranikmath, V. I., Siddiqua, A., &Pramodhini, R. (2022). Plant leaf disease detection using computer vision and machine learning algorithms. Global Transitions Proceedings, 3(1), 305-310.

[25] Tudi, M., Daniel Ruan, H., Wang, L., Lyu, J., Sadler, R., Connell, D., ... & Phung, D. T. (2021). Agriculture development, pesticide application and its impact on the environment. International journal of environmental research and public health, 18(3), 1112.

[26] Sharma, M., Rastogi, M., Srivastava, P., & Saraswat, M. (2022, May). Intelligent pesticide recommendation system based on plant leaf disease and pests. In *International Conference on Advancements in Interdisciplinary Research* (pp. 352-361). Cham: Springer Nature Switzerland.

[27] Rajeshram, V., Rithish, B., Karthikeyan, S., & Prathab, S. (2023, March). Leaf diseases prediction pest detection and pesticides recommendation using deep learning techniques. In *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)* (pp. 1633-1639). IEEE.

[28] Rani, S., Das, K., Aminuzzaman, F. M., Ayim, B. Y., &Borodynko-Filas, N. (2023). Harnessing the future: cutting-edge technologies for plant disease control. Journal of Plant Protection Research, 387-398.

[29] Singh, P., & Shankar, A. (2021). A novel optical image denoising technique using convolutional neural network and anisotropic diffusion for real-time surveillance applications. Journal of Real-Time Image Processing, 18(5), 1711-1728.

[30] Albelwi, S. A. (2022). Deep architecture based on DenseNet-121 model for weather image recognition. International Journal of Advanced Computer Science and Applications, 13(10).

[31] Rachmad, A., Hapsari, R. K., Setiawan, W., Indriyani, T., Rochman, E. M. S., &Satoto, B. D. (2023, May). Classification of Tobacco Leaf Quality Using Feature Extraction of Gray Level Co-occurrence Matrix (GLCM) and K-Nearest Neighbor (K-NN). In 1st International Conference on Neural Networks and Machine Learning 2022 (ICONNSMAL 2022) (pp. 30-38). Atlantis Press.

[32] Nadimi-Shahraki, M. H., Asghari Varzaneh, Z., Zamani, H., &Mirjalili, S. (2022). Binary starling murmuration optimizer algorithm to select effective features from medical data. Applied Sciences, 13(1), 564.

[33] Zhao, W., Wang, L., &Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. Computer Methods in Applied Mechanics and Engineering, 388, 114194.

[34] Indu, Baghel, A. S., Bhardwaj, A., & Ibrahim, W. (2022). Optimization of pesticides spray on crops in agriculture using machine learning. Computational Intelligence and Neuroscience, 2022(1), 9408535.

[35] Chen, J., Zhang, D., Suzauddola, M., & Zeb, A. (2021). Identifying crop diseases using attention embedded MobileNet-V2 model. Applied Soft Computing, 113, 107901.

[36] Chugh, G., Sharma, A., Choudhary, P., & Khanna, R. (2020). Potato leaf disease detection using inception V3. Int. Res. J. Eng. Technol, 7(11), 1363-1366.
https://www.kaggle.com/datasets/tushar5harma/plant-village-dataset-updated