ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

An Ensemble of Transfer Learning based InceptionV3 and VGG16 Models for Paddy Leaf Disease Classification

B. Sowmiya [1], K. Saminathan [2] and M. Chithra Devi [3]

ABSTRACT

Paddy is a crucial food crop providing essential nutrients and energy and serving more than half the global population. Diagnosing and preventing plant diseases at an early stage is crucial for the health and productivity of crops. Automated disease diagnosis eliminates the need for experts and delivers accurate outcomes. This research will diagnose paddy leaf diseases with Deep Learning technology. The diseases such as bacterial blight, blast, tungro, brown spot, and healthy leaf classes are diagnosed and classified in this study. The dataset contains 160 images from each class with 800 images. Our proposed model is an ensemble of transfer- learned InceptionV3 and VGG16 architectures, which utilizes the strength of individual models to improve overall performance. The use of transfer-learned ensemble deep learning architectures achieved impressive accuracy rates of 97.03%, 94.97%, and 98.87% for training, validation and testing respectively. The results indicating that model is not overfit and generalizes well to unseen data. The model's performance is evaluated with confusion matrix with the parameters like precision, recall, F1-score, and support. We also tested the model's performance against other proposed deep learning techniques with and without transfer learning techniques. Moreover, this research advances reliable automated disease detection systems, fostering sustainable agriculture and enhancing global food security.

Article information:

Keywords: Paddy Leaf Diseases, Classification, Transfer Learning, Ensemble, InceptionetV3, VGG16, Deep Learning

Article history:
Received: October 20, 2023
Revised: November 30, 2023
Accepted: February 1, 2024
Published: February 10, 2024 (Online)

DOI: 10.37936/ecti-cit.2024181.254501

1. INTRODUCTION

Rice is an important crop of the world, consumed everyday by half of the world's population. It also supplies 20% of dietary energy. It is one of the most strategic commodities worldwide. The key challenge in combating hunger and malnutrition is the fast growing population. It is expected that the population will reach 8.5 billion by 2030 [1]. India plays a prominent role in agriculture and ranks second in holding the largest agricultural land area. In addition, 55% of the Indian farmers contribute to the agricultural sector. It is predicted that the Indian agricultural sector will increase to US$ 24 billion by 2025, as of FY22, India has already captured nearly 50% of the world market for rice [2]. Biotic and abiotic factors can both significantly impact paddy plants, potentially resulting in detrimental consequences for their growth and survival. Plant pathogens, drought, over-irrigation, nutrient deficiencies, excessive fertilizers, and high salinity significantly decrease global crop yields by over 50%, posing a severe threat to sustainable agriculture [3]. Although symptoms are visible throughout the plant, leaf images are valuable for detecting plant diseases, nutrient deficiencies, and water scarcity, as they reveal crucial clues through changes in shape, color, and overall leaf health. This information aids farmers in making timely and informed decisions to enhance crop health and yield [4].

Despite the availability of advanced technologies, farmers generally rely on naked-eye observation for disease detection. However, this manual approach brings about several issues, including inaccurate pre-

[1,2] The authors are with P.G and Research Department of Computer Science,A.V.V.M. Sri Pushpam College (Autonomous), Affiliated to Bharathidasan University, Thanjavur, Tamil Nadu, India 613 503., Email: sowmiyabaskar@gmail.com and arksami@avvmspc.ac.in
[3]The author is in Queens College of Arts and Science for Women, Affiliated to Bharathidasan University, Pudukkottai, India., Email: m.chithradevi@gmail.com.
[1] Corresponding author: sowmiyabaskar@gmail.com

dictions, time-consuming processes, and increased costs [5]. Automated disease detection helps farmers and plant pathologists by providing reliable observations of leaf diseases overcoming complexities and misconceptions in optical diagnosis and enhancing agricultural practices [6]. The rapid global population growth and decline in rice farmers threaten this affordable staple food. Early and accurate diagnosis of diseases is crucial to prevent plant damage and ensure its usefulness, safeguarding the availability of this vital food source [7]. Plant disease identification benefits from computer vision technologies, such as Machine Learning (ML) [8], Convolutional Neural Networks (CNN) [9], and Deep Learning (DL) [10]. These advancements enable accurate and efficient detection of plant diseases. Researchers utilize DL for disease diagnosis when feature extraction is complex. DL, a subset of ML, automates feature extraction with enhanced accuracy and visualization [11]. Linear classifiers and other traditional ML algorithms may have difficulty effectively dealing with the non-linear complexities present in plant phenotyping tasks. DL models are skilled at automatically identifying complex patterns, resulting in improved representation and comprehension of plant attributes [12]. This study aims to create a deep learning system based on ensemble of deep learning models with a transfer-learning approach. The goal is to diagnose diseases accurately and cost-effectively, even without expert involvement. By ensembling models, we aim to develop a robust and dependable system that can provide accurate diagnoses, revolutionizing the field of disease detection and ultimately benefiting farmers. The contributions of this study include:

- Ensembling deep convolutional neural networks, incorporating pretrained architectures such as InceptionV3 and VGG16 with transfer-learning, to identify the paddy classes including bacterial blight, blast, brown spot, tungro and healthy leaf.
- Stacking of base models, later their predictions are combined using another model (called the meta-model or combiner) to make the final prediction.
- Evaluating the performance of the proposed model in contrast to other advanced proposed models such as MobileNet, ResNet, etc., both with and without transfer learning techniques.

The rest of the paper is organized as follows. Section 2 covers the related works. Section 3 depicts the materials and methods. Section 4 comprises experimental results and discussion. Section 5 concludes the paper.

2. LITERATURE REVIEW

This section discusses various techniques researchers to diagnose plant leaf diseases, with a strong emphasis on the Deep Learning methodology in the literature.

The authors investigated the prediction of four different diseases using ensemble learning paradigm based on pretrained architectures, including EfficientNetB0, InceptionV3, MobileNetV2, and VGG19. They employed bagging and weighted averaging strategies. The dataset was self-collected. Metrics like accuracy, recall, precision, and F1-score were evaluated [13]. Paymode et al. proposed a model for multi-crop leaf disease classification. The research used the PlantVillage dataset for training and self-collected images for testing. They fine-tuned the VGG model and reported accuracy of 98.40% for grapes and 95.71% for tomatoes, respectively [14]. The study introduced a robust approach for pneumonia detection by combining deep learning models, pre-activated ResNet with DenseNet169. The emphasis was on stability in performance curves, resulting in precision of 98%, AUC of 97%, and a loss of 0.23%. It highlighted the impact of data, optimizer choice, and simplified model design. The method outperformed prior research, handling limited data effectively and offering potential applications in COVID-19 data with integrated models [15]. Gautam et al. proposed a model for identifying three paddy leaf diseases such as blight, blast, brown spot, and a healthy class. They collected data from Kaggle and Mendeley. After preprocessing, semantic segmentation was done on the dataset and passed to pre-trained models, and they applied transfer learning to the models. The proposed method used InceptionV3, VGG16, ResNet, SqueezeNet, and VGG19. InceptionV3 outperformed with 96.43% accuracy. In addition, they checked the performance of the model with metrics such as F1-score, precision, and recall [16]. Saleem M. H et al. reviewed various DL models, discussing their parameters, pros and cons, different visualization techniques, improved DL architectures and performance metrics for various plant diseases. In addition, research gaps were identified [17]. Saleem M.H. et al. employed DL for plant disease classification by selecting the optimal model and optimizer for high accuracy on the PlantVillage dataset. The chosen Xception architecture was trained with the Adam optimizer and achieved a remarkable accuracy of 99.81% and an F1-score of 0.9978. These results highlighted the potential of transparent agriculture disease detection with this approach [18].

Aggarwal M. *et al.* demonstrated pre-trained models to detect major diseases and highlighted that segmentation enhances results. The EfficientNet models with Extra Tree achieved 91% accuracy without segmentation and 94% with segmentation [19]. Sengupta S. *et al.* introduced an integrated method for predicting rice plant diseases within a big data context. They extracted attributes from images, applied rough set theory-based feature selection, employed ensemble machine learning classification for accurate disease prediction, and implemented the system within Hadoop's MapReduce framework for efficient large-scale data processing [20]. Ahmad A. *et al.* explored the capacity of deep learning models to detect plant diseases in diverse settings. It utilized five DNN architectures on corn disease images from various datasets. DenseNet169 excelled, achieving up to 81.60% accuracy by training on background-removed RGBA images. The combination of lab and field images improved generalization (up to 80.33%). Background removal and diverse data enhanced DL model generalization which is relevant for flexible disease management systems in different environments [21]. The authors introduced ensemble model combining EfficientNetB0 and MobileNetV2 for plant disease classification employing the PlantVillage Dataset with 54,305 images. The model achieved 99.77% accuracy and demonstrated the ensemble model's potential in enhancing disease classification [22]. In the study, the ensemble DEX model (combining Densenet121, EfficientNetB7, and XceptionNet) achieved the highest accuracy in rice disease classification among various models, surpassing both transfer learning and six individual CNN architectures [23]. Petchiammal A. *et al.* presented PaddyNet, a 17-layer model for accurate paddy leaf disease detection, validated on a dataset of 16,225 samples across 13 categories, including 12 diseases and a healthy class. The model has achieved 98.99% accuracy and outperforms other classifiers [24]. Incorporating both deep and traditional techniques, this research integrated the features from the ResNet50 and DenseNet121 architectures, resulted in an accuracy of 97.59% and F1-score 90.50% through data augmentation. Notably, the ResNet50-Logistic Regression model excelled with a recall of 93.08% and a peak F1-score of 93.80%, matching the former accuracy [25].

Turkoglu M. *et al.* have compared diverse methods for ensembling deep convolutional networks for plant disease and pest classification. Utilizing state-of-the-art networks (AlexNet, GoogleNet, ResNet18, ResNet50, ResNet101, and DenseNet201) with transfer learning and SVM classifiers yielded varied performances. To validate, they curated the Turkey-Plant Dataset with 4,447 images. The proposed PlantDiseaseNetMV and PlantDiseaseNetEF models have achieved 97.56% and 96.83% accuracy, respectively, surpassing state-of-the-art approaches. Future plans involve hybrid models of ML and DL, along with the implementation of feature reduction techniques to enhance overall efficiency [26]. Chen J. *et al.* approach involves two components: a pre-trained module for feature extraction and an auxiliary structure for accurate multi-scale feature map-based detection. The customized, fully connected Softmax layer at the top network forms the INC-VGGN model, effectively predicting plant diseases [27]. Dou S. *et al.* developed the CBAM-MobileNetV2 model with transfer learning for accurate citrus huanglongbing (HLB) diagnosis in southern China. It achieved 98.75% accuracy using attention and convolution modules [28]. Arshad F. *et al.* proposed a hybrid DL model called the PLDPNet model. It has combined features of VGG19 and Inception-V3, also incorporating vision transformers for better predictions, and produced 98.66% accuracy on a dataset of potato leaves. Validating the model on apple and tomato datasets underscores its potential practical application in agriculture [29]. Liu G. *et al.* introduced a lightweight network for precise plant disease classification with an accuracy of 99.28%, surpassing other models by almost 6%. Despite comparable FLOPs to ShuffleNetV2, superior results achieved. Utilizing deep transfer-learning through JAN and integrating SK blocks into MobileNet enhances micro-lesion feature recognition [30].

Nayak A. *et al.* used smartphones and advanced cameras for early plant disease detection via CNNs. They used smartphone images of rice plants and applied segmentation techniques such as foreground extraction. ResNet50 was suited for cloud usage and MobileNetV2 was suited for smartphones. The "Rice Disease Detector" app with MobileNetV2 demonstrated successful performance. Further research is required to evaluate its effectiveness across diverse smartphones [31]. Mahum R. *et al.* introduced a new method to categorize potato leaf diseases into five classes. It modified the DenseNet-201 design by including a transition layer, making the network more compact. It recognized four diseases and handled imbalanced data using a reweighted class-adjusted cross-entropy loss function. Tested on Plant Village Dataset and a custom dataset, it achieved 97.2% accuracy and surpassed existing methods [32]. Pavithra P. *et al.* highlighted a method using three optimization techniques to detect leaf diseases. It started with noise reduction using a Modified Wiener Filter. Improved Ant Colony Optimization extracted vital features. The approach, Hybrid Grasshopper Optimization with modified Artificial Bee Colony Algorithm, classified diseases in leaves. The research achieved 98.53% accuracy on the Plant Village dataset by utilizing evaluation metrics such as accuracy, precision, recall, false negative rate (FNR), negative predictive value (NPV), and Matthews correlation coefficient (MCC). [33]. Abd Algani Y. M. *et al.* used DL with Ant Colony Optimization – CNN (ACO-CNN) to diagnose plant leaf diseases accurately using leaf images. ACO utilized ants' indirect communication and path-finding behavior to achieve approximate optimization. This approach improved disease identification through color, texture, and leaf arrangement analysis [34]. Panchal A. V. *et al.* developed a DL model for classification tasks, aiding disease severity detection. Despite limited input data, training with diverse network architectures and fine-tuned popular models (InceptionV3, ResNet50, VGG19, and VGG16) improved model efficiency. The VGG model achieved 93.5% accuracy on validation data [35]. Thakur P. S. *et al.* presented 'VGG-ICNN,' a lightweight Convolutional Neural Network for identifying crop diseases from plant-leaf images. With about 6 million parameters, it was comparatively efficient. The evaluation covers five public datasets, encompassing diverse varieties of crops. It involves various datasets, including multi-crop sets such as PlantVillage and Embrapa (38 and 93 categories, respectively) and single-crop sets for Apple, Maize, and Rice and produced 99.16% accuracy on the PlantVillage dataset [36]. Deb S.D. *et al.* proposed ConvPlant-Net, a CNN-based plant disease detection system, which is a combination of Depth-Wise Separable Convolutional, 2D Transpose Layer and Convolutional Layer that efficiently learns high and low-level features. The model achieved 98.62%, 99.36%, and 99.60% accuracies on Tomato, Pepper Bell, and Potato crops from the PlantVillage dataset with trainable parameters. The classes such as target spot, early blight, late blight, healthy, and leaf mold were chosen [37]. Malathi V. *et al.* utilized various machine learning techniques, including SVM, KNN, Random Forest, AdaBoost, and neural networks for classifying paddy leaf diseases. Accuracy, recall, precision, F1 score, and ROC AUC (with different fold values) were used. The neural network achieved an AUC of 0.997 and a CA of 0.942 with ten folds. Its F1- score, precision, and recall also surpassed baseline classifiers. The study highlighted the superiority of the neural network and revealed that increasing the number of folds was associated with progressive improvement in the model's performance metrics [38].

3. MATERIALS AND METHODS

The proposed system attempts to create a reliable technique for identifying diseases in paddy plants by ensembling pre-trained Deep Convolution Neural Networks (DCNNs) with transfer learning. Figure 1 depicts the proposed workflow.
This section provides an in-depth explanation of the suggested approach, drawing upon transfer learning principles and the architecture of the proposed model.

3.1 Materials

The research includes 160 image samples with an equal distribution of five classes and 800 images. The images in each class (160 images) are split into three subsets for classification. The training set comprises 60% of the total dataset (96 images), while both the validation and the test sets consist of 20% each (32 images). Figure 2 depicts the sample images of all five classes of paddy images.

Table 1 represents the description of the dataset, URL of the image, and their accessibility information.

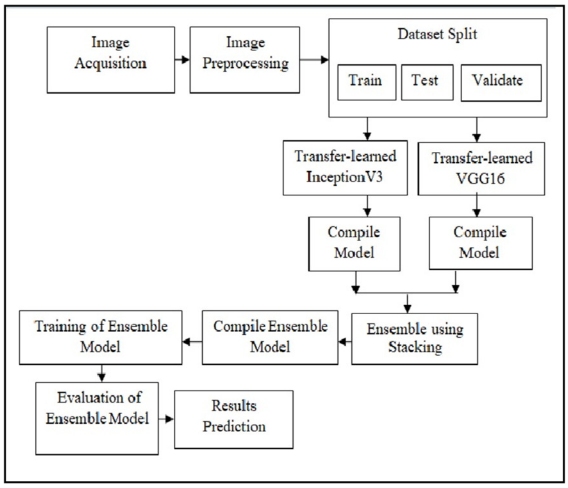

Fig.1: *The Proposed Workflow.*

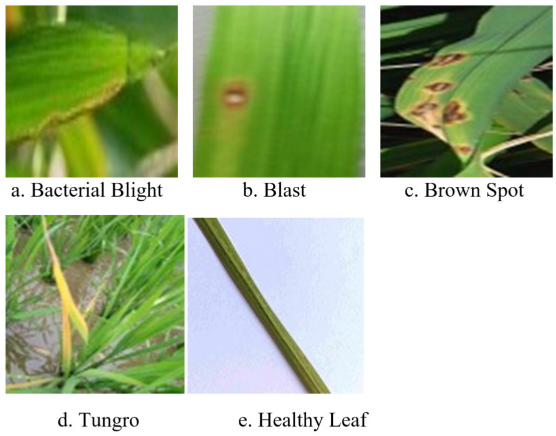

Fig.2: *Sample paddy leaves.*

Table 1: *Dataset description.*

Class	Dataset	URL	Accessibility
Bacterial blight	P.K.Sethy [38]	http://dx.doi.org/10.17632/fwcj7stb8r.1	Publicly accessible.
Blast			
Brown spot			
Tungro			
Healthy	Kaggle	https://www.kaggle.com/datasets/minhhuy2810/rice-diseases-image-dataset	

Table 2 represents dataset partition details, including division for training, testing, and validation images.

Table 2: *Dataset partition.*

Class name	Total images	Train	Validate	Test
Bacterial Blight	160	96	32	32
Brown Spot	160	96	32	32
Blast	160	96	32	32
Tungro	160	96	32	32
Healthy	160	96	32	32
Total	800	480	160	160

Table 3 represents the model's hyperparameters, such as epochs, L2_weight, loss function, optimizer, etc. The choice of hyperparameter has a high impact on the results of a model.

Table 3: *Model's hyperparameters.*

Hyperparameter	Description	Value
input_shape	The shape of the input images	(224, 224, 3)
num_classes	Number of classes in the classification task	5
Epochs	Number of training epochs	30
dropout_rate	Dropout rate for regularization	0.5
L2_weight	Weight of L2 regularization	0.0001
Optimizer	Optimizer used for model compilation	Adam
Loss	Loss function used for model compilation	Categorical Cross-Entropy

The effectiveness of the proposed model is assessed through a metrics such as accuracy, precision, recall, F1-score, and AUC (Area Under the Curve). These metrics are derived using Formulas 1 through 5. Table 4 represents classification metrics.

$$\text{Accuracy} = (\text{TP} + \text{TN}) \ / \ (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (1)$$

$$\text{Recall} = TP/(TP + FN) \quad (2)$$

$$\text{Precision} = TP/(TP + FP) \quad (3)$$

$$\text{F1-score} = 2 * ((\text{precision} * \text{recall}) \ / \ (\text{precision} + \text{recall})) \quad (4)$$

$$\text{AUC} = \int [0,1]\text{ROC(fpr)dfpr} \quad (5)$$

Table 4: *Classification metrics.*

Abbreviation	Full Form	Definition
TP	True Positive	Correctly identified positive instances
FN	False Negative	Incorrectly labeled as negative when positive
FP	False Positive	Incorrectly labeled as positive when negative
TN	True Negative	Correctly identified negative instances
AUC	Area Under the Curve	Distinguishing between positive and negative instances.

3.2 Materials

The overall architecture of the proposed model's approach is depicted in Figure 3.

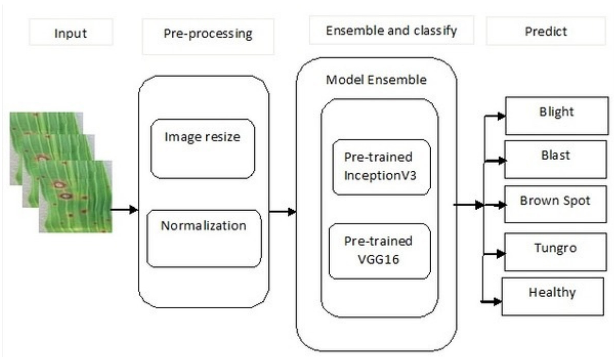

Fig.3: *The Proposed model's approach.*

3.2.1 Preprocessing

Preprocessing is essential for enhancing image quality and is a crucial step for neural networks to achieve effective learning and generalization. All the images have been resized to 224*224 pixels. The dataset is normalized to a range of 0 to 1 before being input into the neural network model. The normalization of the dataset is achieved by dividing the values by 255. Both the training and the testing sets are preprocessed using the same method. Together, the processes of resizing and normalization will modify the visual attributes and qualities of the images to prepare them for input into a neural network for training and evaluation purposes. Eq. 6 and Eq. 7 represents the formula for resizing and normalization.

$$Y_{i,j} = \frac{224}{\text{origunal\_width}} * X_{i,j} \quad (6)$$

Where $Y_{i,j}$ is the pixel value at position (i,j) in the resized image, and $X_{i,j}$ is the pixel value at the corresponding position in the original image.

$$Z_{i,j} = \frac{Y_{i,j}}{255} \tag{7}$$

Where $Z_{i,j}$ represents the normalization of image, which has undergone resizing.

3.2.2 Classification using Proposed Transfer learning based ensemble InceptionV3 and VGG16

Ensemble learning is a machine learning strategy where multiple base models are combined to tackle a problem. Various techniques such as Bagging, Boosting, and Stacking exist for ensemble modeling [13]. In this research, we have focused on Stacking. In stacking, multiple base models are trained individually on the same dataset, then their predictions are combined using another model (called the meta-model or combiner) to make the final prediction. Figure 4 depicts the proposed ensemble model steps.

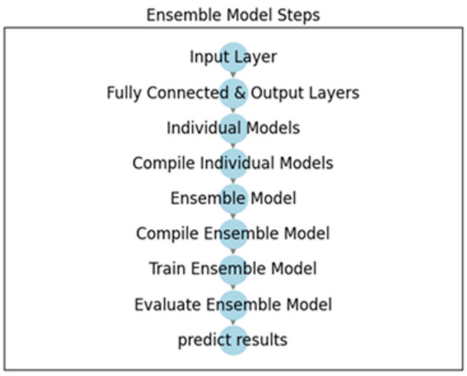

Fig.4: *Steps of the Proposed Ensemble Model.*

Stacking enhances predictions by using different models together, which helps understand complex patterns better. Transfer learning for training deep learning models on small datasets lead to enhanced accuracy and faster training times compared to training models from the scratch [40].

The algorithm of the proposed ensemble model using Inception V3 and VGG16:

Step 1: Define the input layer for the ensemble model.

Step 2: Define the fully connected layers and output layers for each base model.

Step 3: Create individual models for each base model.

Step 4: Compile the individual models.

Step 5: Define the ensemble model by averaging the outputs of the individual models.

Step 6: Compile the ensemble model.

Step 7: Train the ensemble model using the training data.

Step 8: Evaluate the ensemble model on the test data.

Our ensemble DCNN incorporates transfer learned architectures of **InceptionV3** and **VGG16**. InceptionV3 is a DCNN architecture designed for image classification. It features inception modules that enable efficient multi-level feature extraction. With 48 convolutional layers and 24 million parameters, InceptionV3 is pretrained on large-scale datasets such as ImageNet for transfer learning. It has achieved remarkable performance on image classification benchmarks, demonstrating its ability to handle complex visual patterns [41]. Due to its robust feature extraction capabilities and high accuracy in visual recognition tasks, InceptionV3 is widely used as a base architecture in various computer vision tasks, including object detection and image segmentation. Moreover, the InceptionV3 architecture proved to be more efficient when compared to the VGGNet family of architectures [21].

Let $N = \{InceptionV3, VGG16\}$ is the set of the transfer learned CNNs architectures. Each network $n \in N$ was trained on five different classes in the dataset (Mi, Si) where M is the number of pictures. Each image is resized to 224 $\times$ 224 $\times$ 3 and normalized in the [0, 1] interval, and S is the relative symptoms, that are the labels of images, $S = \{$Bacterial Blight, Blast, Brown spot, Tungro, healthy$\}$.

Using weighted average method, the final prediction is done. Each model n in the set N gets a fixed weight, multiplying its prediction from the classifier. To obtain the final prediction, we average these weighted predictions. Equation 8 defines the formula for weighted average method.

$$P' = \sum\nolimits_{i=1}^{n} x_i \cdot y' \tag{8}$$

Where P' represents the resulting value, n is the total number of instances of ensemble CNN, x_i denotes the weight assigned to the i^{th} instance, and y' stands for the predicted value of an instance.

The proposed model creates an ensemble model by choosing pre-trained InceptionV3 model and VGG16 as the base models. Their predictions are averaged. The top layers of base models are frozen to retain their learned features and prevent them from being trained during the ensemble model training. We add GlobalAveragePooling2D layer on top of each base model, to convert the 2D feature maps into a 1D vector, which serve as fully connected layer. We add a dense layer with 128 units and a ReLU activation function to each base model. These layers help capture high-level features and patterns from the base model's outputs. Finally, we add a dense layer with softmax activation to produce class predictions. We compile the base models separately using the Adam optimizer and categorical cross-entropy loss function. The base and ensemble models are trained using the training data and validated using the validation data. We store the training history in the 'history' variable.

We evaluate the ensemble model on the test dataset. The test loss and accuracy are calculated and printed. We plot the training loss, validation loss, training accuracy, and validation accuracy using Matplotlib. During the training, learning rate scheduling adjusts the learning rate and controls gradient descent step sizes. Its goal is to enhance model convergence by gradually lowering the learning rate over time [22]. Table 5 shows the architecture of the proposed ensemble model.

Table 5: *Architecture of the proposed Ensemble model.*

Layer (type)	Output Shape	Param♯	Connected to
input_4 (InputLayer)	[(None, 224, 224, 3)]	0	[]
model (Functional)	(None, 5)	22065701	['input$_4$[0][0]']
Model$_1$ (Functional)	(None, 5)	14780997	['input$_4$[0][0]']
average (Average)	(None, 5)	0	['model[0][0]', 'model$_1$[0][0]']
Total params: 36846698 Trainable-params: 329226 Non-trainable params: 36517472			

Figure 5 depicts the visualization of the proposed ensemble model, depicting the testing and averaging of the individual models, as well as clearly illlustrating the stacking of the ensemble model.

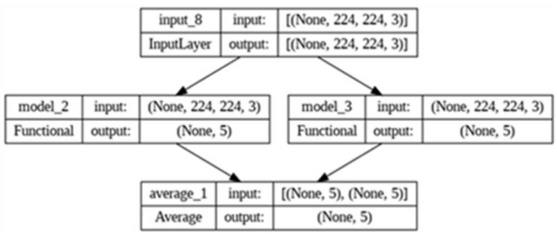

Fig.5: *Visualization of stacked ensemble models.*

4. EXPERIMENTAL RESULTS AND DISCUSSION

We represent the outcomes of the experiments according to the fundamental structures of the original individual neural networks, the application of transfer learning, and the utilization of ensemble methods. The achieved outcomes aim to address the subsequent inquiries: *Does the incorporation of transfer learning contribute to an enhancement in accuracy? Can the implementation of ensemble techniques further enhance accuracy?*

4.1 Experimental Setup

The proposed system utilizes an Intel(R) Core(TM) i3-5005U CPU with 4GB of RAM and a 64-bit processor. We develop the proposed model with Google Colab, a cloud platform. It provides rich support with pre-built packages like Keras, Tensorflow, etc. It has a rich set of tools for the development and implementation.

4.2 Experimental Results

After fine-tuning through several runs with varying epochs, dropout rates, L2_weights, optimizers, and loss functions, we adjusted the hyperparameters as detailed in Table 3. The code was run multiple times, each time for a specific number of epochs. The model was trained and evaluated in each run to calculate accuracy and loss. We regularly recorded metrics to track progress. From epochs 1 to 30, the model steadily improved accuracy and decreased loss.

Training Accuracy

This metric shows how accurately the model predicts the training data labels. It begins at 96.88% and fluctuates between 88.91% and 99.22% over epochs. A higher value is better, indicating the model fits well to the training data.

Training Loss

The training loss quantifies the difference between predicted values and actual in the training dataset. It starts at 0.2233 and varies between 0.0888 and 0.4633. Smaller values imply a better fit to the training data.

Validation Accuracy

Reflecting the model's performance on unseen data, it starts at 87.42% and ranges between 84.91% and 96.86%. An upward trend is positive, showing the model generalizing better to new data.

Validation Loss

The validation loss measures the difference between model predictions and actual values in the validation dataset. It initiates at 0.431 and fluctuates between 0.1641 and 0.5561. Smaller values indicate better generalization ability. Figure 6 and 7 depict the loss and accuracy curves on the proposed ensemble model for 30 epochs.

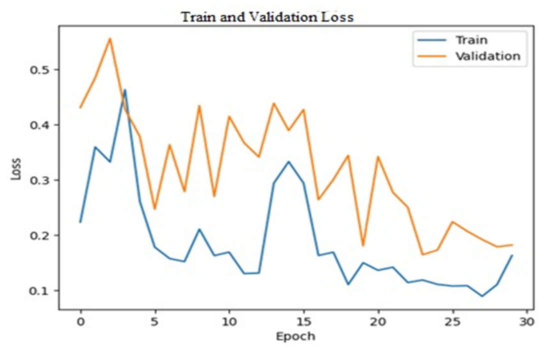

Fig.6: *The Proposed model's loss.*

Figure 8 shows the ROC curve of the confusion matrix, which represents the performance of the classification model. The ROC curve is a graphical representation of the performance of a model. We plot

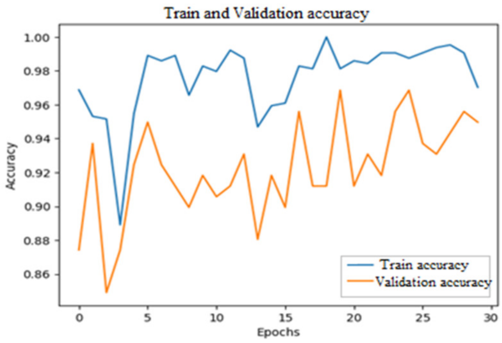

Fig.7: *The Proposed model's accuracy.*

True positive rate and false positive rate for diverse threshold values.

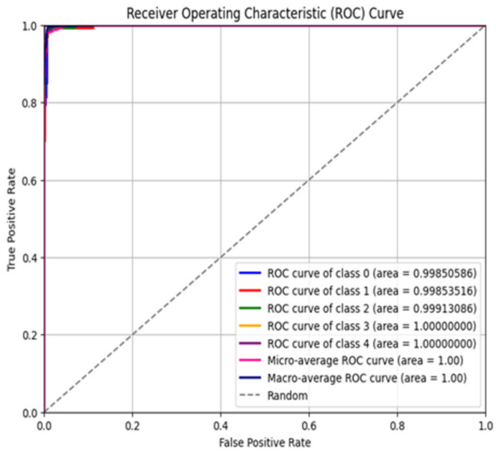

Fig.8: *The Proposed model's ROC Curve.*

Figure 9 depicts the confusion matrix of the proposed method with correctly classified instances in the range of 155-160.

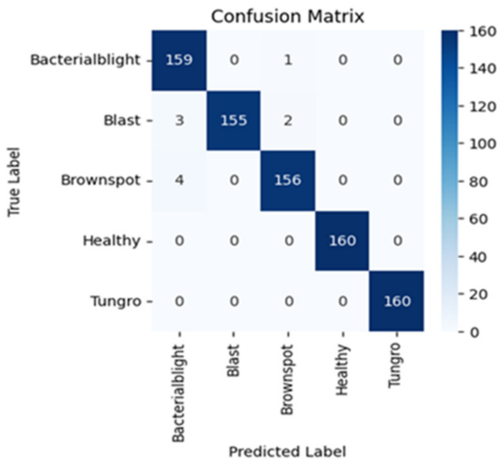

Fig.9: *Confusion matrix.*

Table 6 provides the classification report of the proposed model.

We analyze the effectiveness of a classification model by analyzing a confusion matrix, which compares the predicted outcomes of the ensemble model with the actual desired outcomes. Table 7 shows the results of the proposed model in comparison with other models.

Table 6: *Classification report of proposed model.*

	Precision	Recall	F1-score	Support
Bacterial blight	0.96	0.99	0.98	160
Blast	1	0.97	0.98	160
Brown spot	0.98	0.97	0.98	160
Tungro	1	1	1	160
Healthy	1	1	1	160
Accuracy		0.99	800	
macro avg	0.99	0.99	0.99	800
weighted avg	0.99	0.99	0.99	800

From Table 7, we discuss the findings. In essence, the model begins with high accuracy on training data and gradually improves. During training, the loss decreases, implying better alignment with training data. Similarly, the model learns to generalize better as validation accuracy improves and validation loss reduces, though there are fluctuations. The overall goal is to lower the loss and improve accuracy for both the training and validation datasets. The result of model performance varies across different approaches. *The "Ensemble InceptionV3 and VGG16" achieve the highest testing accuracy, indicating strong generalization.* "TL + MobileNetV2" and "TL + InceptionV3" show high training accuracy but slightly lower validation and testing performance. The "TL + Ensemble ResNet50 + VGG16" maintains balanced accuracy. *Notably, the "TL + Ensemble ResNet50 + MobileNetV2" model has high training but lower validation/testing accuracy, suggesting overfitting.* "Ensembling InceptionV3+Vgg16 (No TL)" overfits significantly, while "TL + ResNet50" lags. The proposed ensemble model achieves 97.03% of training accuracy, 94.97% of validation, and 98.87% of testing accuracy. Figure 10 depicts the training accuracy comparison with other models.

Figure 11 depicts the validation accuracy comparison with other models.

Figure 12 depicts the testing accuracy comparison with other models.

Table 8 depicts metrics such as precision, recall, F1-score for comparison with other models. We compare the proposed ensemble model with other deep learning classifiers, both with and without transfer learning techniques. The proposed ensemble architecture outperforms the other DL architectures used for comparison.

Table 7: *Accuracy Comparison of the Proposed Ensemble Model with Other DL Models.*

Model	**Train Accuracy**	**Validation Accuracy**	**Testing Accuracy**	**Total params**	**Trainable params**	**Non-trainable params**
TL+InceptionV3	99.06	88.68	98.25	2,20,65,701	2,62,917	2,18,02,784
TL+ResNet50	71.72	56.25	70.63	2,38,50,629	2,62,917	2,35,87,712
TL+Ensemble ResNet50 + VGG16	95.16	90.62	95.13	3,86,31,626	3,29,226	3,83,02,400
TL+MobileNetV2	100	94.38	98.75	3,86,31,626	3,29,226	3,83,02,400
TL+Ensemble ResNet50+ mobileNetV2	99.53	95.63	98.87	3,65,17,472	3,68,46,698	3,29,226
Ensemble InceptionV3+ VGG16 (No TL)	98.75	74.37	74.87	3,68,46,698	3,68,12,266	34,432
Proposed TL+ Ensemble InceptionV3 and VGG16	97.03	94.97	98.87	3,68,46,698	3,29,226	36517472

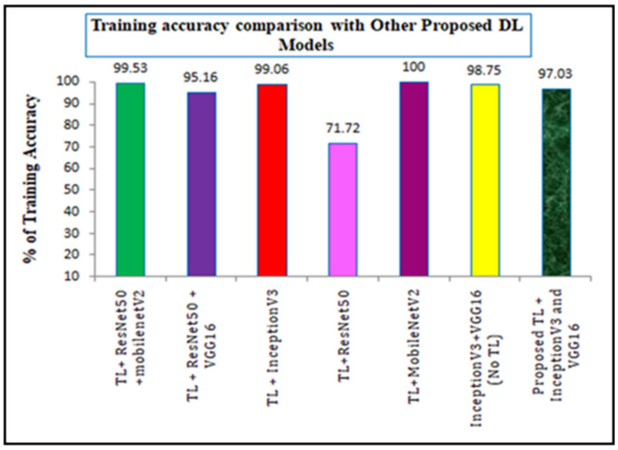

Fig.10: *Training Accuracy comparison with other DL models.*

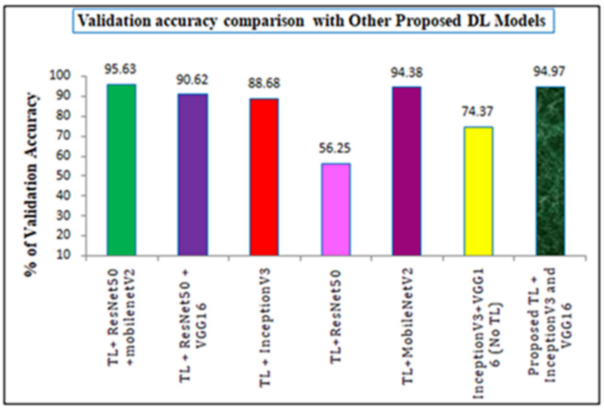

Fig.11: *Validation Accuracy comparison with other DL models.*

Table 8: *Performance Metrics of Proposed Ensemble Model with other Architectures.*

Method	**Class**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Ensemble InceptionV3 +VGG16 (No TL)	0	0.53	0.85	0.66
	1	0.67	0.04	0.07
	2	0.78	0.88	0.83
	3	0.94	1	0.97
	4	0.84	0.97	0.9
Proposed TL +Ensemble InceptionV3 and VGG16	0	0.96	0.99	0.98
	1	1	0.97	0.98
	2	0.98	0.97	0.98
	3	1	1	1
	4	1	1	1
TL+InceptionV3	0	0.92	1	0.96
	1	0.99	0.97	0.98
	2	1	0.94	0.97
	3	1	1	1
	4	1	1	1
TL+Ensemble ResNet50 and VGG16	0	0.98	0.99	0.98
	1	0.9	0.87	0.88
	2	0.89	0.9	0.89
	3	1	1	1
	4	0.99	1	1
TL+ResNet50	0	0.52	0.64	0.58
	1	0.38	0.64	0.48
	2	0.76	0.17	0.28
	3	0.79	0.86	0.83
	4	0.7	0.54	0.61
TL+MobileNetV2	0	0.99	0.99	0.99
	1	0.95	0.99	0.97
	2	1	0.95	0.97
	3	1	1	1
	4	1	1	1
TL+Ensemble ResNet50 and MobileNetV2	0	1	0.98	0.99
	1	0.98	0.98	0.98
	2	0.97	0.98	0.98
	3	1	1	1
	4	1	1	1

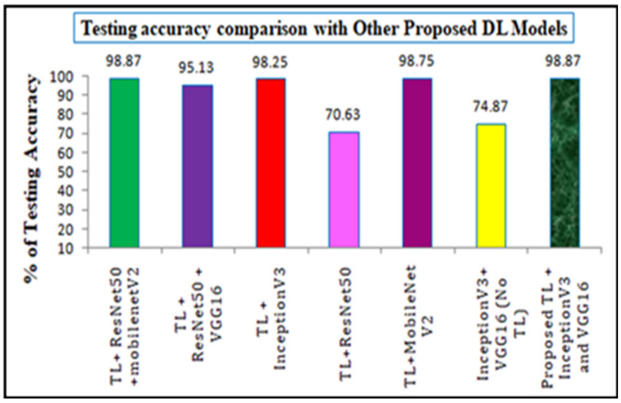

Fig.12: *Testing Accuracy comparison with other DL models.*

Finally, we evaluate our achieved accuracy in comparison to accuracy levels conducted by various researchers in the field. The results indicate that our proposed model outperforms others in terms of accuracy. ***The answers to research questions discussed in the beginning of this section (section 4) are yes. The incorporation of transfer learning contributes to an enhancement in accuracy, and the implementation of ensemble techniques further enhances the accuracy.*** Table 9 displays a comparison of the accuracy of the proposed model with that of other models.

Table 9: *Comparison of the proposed model with other models.*

Reference	Accuracy	Architecture Used
Fenu, G. *et al.*	91.14%	EfficientNetB0+ InceptionV3
Arshad F *et al.*	98.66%	Inception+VGG19
Qi. H *et al.* [25]	97.59%	ResNet50 and Logistic Regression (LR)-97.59%,
Our proposed Ensemble model	***98.87%***	***Fine-tuned InceptionV3+VGG16***

5. CONCLUSION

Plant disease diagnosis plays a pivotal role in agriculture. The proposed ensemble model with pre trained InceptionV3 and VGG16 leverages the strengths of multiple models to improve the overall performance and generalization. By combining the predictions of the individual models, better accuracy and robustness can often be achieved compared to a single model. The suggested ensemble model delivers remarkable outcomes, boasting a training accuracy of 97.03%, a validation accuracy of 94.97%, and an impressive testing accuracy of 98.87%. This investigation revolves around five distinct disease categories: bacterial blight, blast, tungro, brown spot, and healthy leaves. The primary advantage of utilizing DL techniques is their capability to eliminate the need for labor-intensive manual feature extraction. Utilizing automatic methods for diagnosing these diseases can benefit farmers by increasing yields and reducing unnecessary pesticide use, which can harm the soil. Overall, such an automated disease detection system can be a valuable tool for farmers, empowering them to make informed decisions and enhance their productivity and livelihoods. The futuristic work is to implement with an IOT (Internet-of-Things) setup and develop a mobile application. It will include sensor networks, image-capturing devices, and real-time data transmission. We will utilize the cloud storage for scalability and efficient storage of large datasets. Such an integrated technology will revolutionize the plant disease diagnosis process.

References

[1] A. R. Bin Rahman and J. Zhang, "Trends in rice research: 2030 and beyond," *Food and Energy Security*, vol. 12, no. 2, e390, 2023.

[2] `https://www.ibef.org/industry/agriculture-india.`

[3] B. S. Anami, N. N. Malvade and S. Palaiah, "Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images," *Artificial intelligence in agriculture*, vol. 4, pp. 12-20, 2020.

[4] J. A. Wani, S. Sharma, M. Muzamil, S. Ahmed, S. Sharma and S. Singh, "Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges," *Archives of Computational methods in Engineering*, vol. 29, no. 1, pp. 641-677, 2022.

[5] S. K. Upadhyay and A. Kumar, "Early-Stage Brown Spot Disease Recognition in Paddy Using Image Processing and Deep Learning Techniques," *Traitement du Signal*, vol. 38, no. 6, pp. 1755-1766, 2021.

[6] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and electronics in agriculture*, vol. 145, pp. 311-318, 2018.

[7] S. Jain, R. Sahni, T. Khargonkar, H. Gupta, O. P. Verma, T. K. Sharma and H. Kim, "Automatic rice disease detection and assistance framework using deep learning and a Chatbot," *Electronics*, vol. 11, no. 14, pp. 2110, 2022.

[8] M. Sharma, C. J. Kumar and A. Deka, "Early diagnosis of rice plant disease using machine learning techniques," *Archives of Phytopathology and Plant Protection*, vol. 55, no. 3, pp. 259-283, 2022.

[9] S. P. Singh, K. Pritamdas, K. J. Devi and S. D. Devi, “Custom Convolutional Neural Network for Detection and Classification of Rice Plant Diseases,” *Procedia Computer Science*, vol. 218, pp. 2026-2040, 2023.

[10] S. H. Lee, C. S. Chan, P. Wilkin and P. Remagnino, “Deep-plant: Plant identification with convolutional neural networks,” *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, QC, Canada, pp. 452-456, 2015.

[11] A. R. T. Z. A. I. Picon Ruiz, A. I. T. O. R. Alvarez Gila, U. Irusta and J. O. N. E. Echazarra Huguet, “Why deep learning performs better than classical machine learning?,” *Dyna Ingenieria E Industria*, 2020.

[12] J. R. Ubbens and I. Stavness, “Deep plant phenomics: a deep learning platform for complex plant phenotyping tasks,” *Frontiers in plant science*, vol. 8, no. 1190, pp.1-11, 2017.

[13] G. Fenu and F. M. Malloci, “Classification of Pear Leaf Diseases Based on Ensemble Convolutional Neural Networks,” *AgriEngineering*, vol. 5, no. 1, pp. 141-152, 2023.

[14] A. S. Paymode and V. B. Malode, “Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG,” *Artificial Intelligence in Agriculture*, vol. 6, pp. 23-33, 2022.

[15] A. T. Al-Taani and I. T. Al-Dagamseh, “Automatic detection of pneumonia using concatenated convolutional neural network,” *Jordanian Journal of Computers and Information Technology (JJCIT)*, vol. 9, no. 2, pp. 118-136, 2023.

[16] V. Gautam, N. K. Trivedi, A. Singh, H. G. Mohamed, I. D. Noya, P. Kaur and N. Goyal, “A transfer learning-based artificial intelligence model for leaf disease assessment,” *Sustainability*, vol. 14, no. 20:13610, 2022.

[17] M. H. Saleem, J. Potgieter and K. M. Arif, “Plant disease detection and classification by deep learning,” *Plants*, vol. 8, no.11:468, 2019.

[18] M. H. Saleem, J. Potgieter and K. M. Arif, “Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers,” *Plants*, vol. 9, no. 10:1319, 2020.

[19] M. Aggarwal, V. Khullar, N. Goyal, A. Singh, A. Tolba, E. B. Thompson and S. Kumar, “Pre-Trained Deep Neural Network-Based Features Selection Supported Machine Learning for Rice Leaf Disease Classification,” *Agriculture*, vol. 13, no. 5:936, 2023.

[20] S. Sengupta, A. Dutta, S. A. Abdelmohsen, H. A. Alyousef and M. Rahimi-Gorji, “Development of a Rice Plant Disease Classification Model in Big Data Environment,” *Bioengineering*, vol. 9, no. 12:758, 2022.

[21] A. Ahmad, A. E. Gamal and D. Saraswat, “Toward Generalization of Deep Learning-Based Plant Disease Identification Under Controlled and Field Conditions,” in *IEEE Access*, vol. 11, pp. 9042-9057, 2023.

[22] H. T. Vo, L. D. Quach and T. N. Hoang, “Ensemble of Deep Learning Models for Multi-plant Disease Classification in Smart Farming,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023.

[23] M. T. Ahad, Y. Li, B. Song and T. Bhuiyan, “Comparison of CNN-based deep learning architectures for rice diseases classification,” *Artificial Intelligence in Agriculture*, vol. 9, pp. 22-35, 2023.

[24] A. Petchiammal, D. Murugan and K. S. Briskline, “PaddyNet: An Improved Deep Convolutional Neural Network for Automated Disease Identification on Visual Paddy Leaf Images,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023.

[25] H. Qi, Y. Liang, Q. Ding and J. Zou, “Automatic identification of peanut-leaf diseases based on stack ensemble,” *Applied Sciences*, vol. 11, no. 4:1950, 2021.

[26] M. Turkoglu, B. Yanikoglu and H D. anbay, “PlantDiseaseNet: Convolutional neural network ensemble for plant disease and pest detection, *Signal, Image and Video Processing*, vol. 16. no. 2, pp. 301-309, 2022.

[27] J. Chen, J. Chen, D. Zhang, Y. Sun and Y. A. Nanehkaran, “Using deep transfer learning for image-based plant disease identification,” *Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020.

[28] S. Dou, L. Wang, D. Fan, L. Miao, J. Yan and H. He, “Classification of Citrus Huanglongbing Degree Based on CBAM-MobileNetV2 and Transfer Learning,” *Sensors*, vol. 23, no. 12, p. 5587, 2023.

[29] F. Arshad, M. Mateen, S. Hayat, M. Wardah, Z. Al-Huda, Y. H. Gu and M. A. Al-antari, “PLDPNet: End-to-end hybrid deep learning framework for potato leaf disease prediction,” *Alexandria Engineering Journal*, vol. 78, pp. 406-418, 2023.

[30] G. Liu, J. Peng and A. A. A. El-Latif, “SK-MobileNet: A Lightweight Adaptive Network Based on Complex Deep Transfer Learning for Plant Disease Recognition,” *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 1661-1675, 2023.

[31] A. Nayak, S. Chakraborty and D. K. Swain, “Application of smartphone-image processing and transfer learning for rice disease and nutrient deficiency detection,” *Smart Agricultural Technology*, vol. 4, p. 100195, 2023.

[32] R. Mahum, H. Munir, Z. U. N. Mughal, M.

Awais, F. Sher Khan, M. Saqlain and I. Tlili, "A novel framework for potato leaf disease detection using an efficient deep learning model," *Human and Ecological Risk Assessment: An International Journal*, vol. 29, no. 2, pp. 303-326, 2023.

[33] P. Pavithra and P. Aishwarya, "Plant leaf disease detection using hybrid grasshopper optimization with modified artificial bee colony algorithm," *Multimedia Tools and Applications*, pp. 1-23, 2023.

[34] Y. M. Abd Algani, O. J. M. Caro, L. M. R. Bravo, C. Kaur, M. S. Al Ansari and B. K. Bala, "Leaf disease identification and classification using optimized deep learning," *Measurement: Sensors*, vol. 25, p.100643, 2023.

[35] A. V. Panchal, S. C. Patel, K. Bagyalakshmi, P. Kumar, I. R. Khan and M. Soni, "Image-based plant diseases detection using deep learning," *Materials Today: Proceedings*, vol. 80, pp. 3500-3506, 2023.

[36] P. S. Thakur, T. Sheorey and A. Ojha, "VGG-ICNN: A Lightweight CNN model for crop disease identification," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 497-520, 2023.

[37] S. D. Deb, R. K. Jha and S. Kumar, "ConvPlant-Net: A Convolutional Neural Network based Architecture for Leaf Disease Detection in Smart Agriculture," *2023 National Conference on Communications (NCC)*, Guwahati, India, pp. 1-6, 2023.

[38] V. Malathi, M. P. Gopinath, M. Kumar, S. Bhushan and S. Jayaprakash, "Enhancing the Paddy Disease Classification by Using Cross-Validation Strategy for Artificial Neural Network over Baseline Classifiers," *Journal of Sensors*, vol. 2023, no. 1576960, 2023.

[39] P. K. Sethy, "Rice Leaf Disease Image Samples," Mendeley Data, V1, 2020. http://dx.doi.org/10.17632/fwcj7stb8r.1.

[40] P. Alirezazadeh, M. Schirrmann and F. Stolzenburg, "Improving Deep Learning-based Plant Disease Classification with Attention Mechanism," *Gesunde Pflanzen*, vol. 75, no. 1, pp. 49-59, 2023.

[41] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and electronics in agriculture*, vol. 147, pp. 70-90, 2018.

B. Sowmiya is working as an Assistant Professor in the department of Computer Science, S.I.V.E.T. College, Chennai. She was University topper at Bharathidasan University, Trichy in Master of Science in Computer in 2006. She has qualified both SET conducted by Government of Tamilnadu and NET conducted by Government of India. Currently, She is pursuing Ph.D. in Computer Science with specialization in Image Processing at A.V.V.M Sri Pushpam College (autonomous) affiliated to Bharathidasan University, Trichy. She has authored few books and book chapters. Her research areas of interests include Digital Image processing, Machine Learning and Deep Learning. She has published research papers in UGC care listed journals, International conferences and Scopus indexed journals. She is a reviewer of scopus indexed journals and board member of peer reviewed journal.

K. Saminathan is working as an Assistant Professor in the department of Computer Science, A.V.V.M Sri Pushpam College (Autonomous), Poondi. He has completed his Post Graduate degree in Computer Science at Periyar E.V.R College, Trichy in 2006. Master of Philosophy in Computer Science at A.V.V.M. Sri Pushpam College, Poondi in 2007. He has completed his M.Tech in Computer Science at PRIST University, Thanjavur in 2011and Ph.D. in Computer Science at Bharathidasan University and degree awarded in the year of 2017. He has published more than 10 research articles in various domains such as Image Processing, Computer Networks and Machine learning in peer reviewed International Journals. He is a research supervisor in Bharathidasan University, Trichy and currently guiding four research scholars.

M. Chithra Devi is working as Principal at Queens College of Arts and Science for Women, Gandarvakottai, Thanjavur (Dt), Tamil Nadu, India. She has completed her M.E in Computer Science at Periyar Maniammai University, Vallam, Thanjavur, Ph.D. in Computer Science at Bharathiar University, Coimbatore. She has published more than 10 research articles in various domains such as Image Processing and Machine learning in peer reviewed International Journals. She is a research supervisor in Bharathidasan University, Trichy and currently guiding three research scholars.