Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19255-z

Progressive loss-aware fine-tuning stepwise learning with GAN augmentation for rice plant disease detection

Kamal Upreti[1] · Prashant Singh[2] · Dhyanendra Jain[3] · Amit Kumar Pandey[3] · Anjani Gupta[4] · Hare Ram Singh[5] · Santosh Kumar Srivastava[6] · Jay Shankar Prasad[5]

Received: 12 August 2022 / Revised: 28 February 2024 / Accepted: 14 April 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Modern technology like Artificial Intelligence (AI) must be used in the agricultural sector if sustainable agricultural output is to be achieved. One of the most convenient strategies for resolving current and future issues is data-driven agriculture. For this, disease prediction is a major task for precise farming. For predictive analysis and precise agriculture monitoring systems, with the application of AI, Machine Learning (ML) and Deep Learning (DL) play vital roles in building a more robust system. In this work, we will design a DL-integrated rice disease prediction system to be implemented for precise farming. Improvisation of the developed model to detect rice plant diseases & pest attacks with a high level of precision. In this work, the Progressive Loss-Aware Fine-Tuning Stepwise Learning (PLAFTSL) model is proposed for disease detection. For step-wise learning fine-tuned ResNet50 model is used with the introduction of freezing and unfreezing layers. This reduces the training parameters and thus computational complexity. The introduction of the step-wise and progressive loss-aware layer will result in fast convergence and improved training efficiency during information exchange among layers respectively. Our proposed work uses a dataset from two sources. The result analysis is presented with an ablation study. Additionally, the baseline model, ResNet50, is used to display the outcomes of the ablation. The results demonstrate that the fine-tuned model results in better performance as compared to the transfer learning model. The Conditional Generative Adversarial Network (cGAN) augmentation is also added to the designed model which will improve detection effectiveness and can also manage the imbalance in input data. The model has achieved approx. 98% accuracy and outperforms better with comparative state-of-art models.

Keywords Plant Disease · Rice Disease · Deep Learning · Transfer Learning · Augmentation

Extended author information available on the last page of the article

Published online: 24 April 2024

1 Introduction

The majority of nations' economies rely heavily on the agriculture sector. It is considered to be an important source of income in developing countries. According to reports presented by the WHO and Food and Agriculture Organization (FAO) to guarantee nutritious and healthy food, macroeconomic stability, and global food safety [1]. Correspondingly, "zero hunger" is among UNESCO's 17th sustainable development goals, which is projected to be accomplished by 2030 [2]. The consumption of food increases in proportion to the world's population. Prioritizing agricultural yields is required to deal with the issues. diseases, plants, animals, and bugs, combined contribute approximately between 20 and 40 percent of crop productivity losses and a staggering $220 billion loss yearly [3], which have a severe influence on agricultural yield quality and quantity worldwide. About 13% of crop productivity loss is responsible for plant diseases [3], which are considered the most contributing factor. However, firstly, it is required to comprehend the causes and influencing aspects of plant disease [4]. These factors are such as host, environment, and pathogens. In many diseases, the symptoms mainly appear from the bottom of the plant and affect to leaves. Therefore, it is needed to inspect and monitor regularly, disease-causing factors and prevent their spread [4]. In major regions of the world such as Asia, Africa, and America, rice is considered a low-cost and effective nutrient food. Therefore, its production is also significantly affected by plant diseases. Some of the examples of rice plant diseases are presented in the appendix. Most rice plant diseases appear on a leaf that needs a visual inspection to identify and distinguish them.

Approx. 50% of plant diseases are caused due to fungal infection [5]. Therefore, to better diagnose these, image processing and computer vision with an addon of ML has been used recently. It has been identified that most plant diseases are due to pathogens and are responsible for up to 50% of production loss. Therefore, researchers have contributed their efforts in image processing and computer vision applications along with ML and DL approaches in the field of precise farming or smart agriculture applications plant disease identification is quite necessary [6]. The traditional techniques for detecting plant diseases were extremely time-intensive and expensive. They also need the supervision of respective experts that involves manual inspection which contributes to delayed preventative measures and biased results that reduce the overall accuracy of the system. To overcome these issues, the paper investigates the application of image processing with ML or DL approaches using plant images [7].

Timely illness diagnosis can aid in the reduction of the usage of hazardous chemicals. Numerous computer vision or picture processing-based systems for illness detection and diagnosis have been created throughout the years. The use of cutting-edge AI approaches has also been recommended for disease diagnosis automated processes. The majority of these approaches rely on vision-based approaches such as image processing (IP), ML, and DL. High-performance AI techniques for detecting plant diseases have been constructed utilizing DL techniques in the past few years [10, 11].

Numerous ML and sophisticated DL techniques have been used in the past to identify plant diseases. Several research [12–14] in the realm of detecting plant diseases addresses this issue by employing step-by-step image processing approaches including picture capturing, picture pre-processing, picture segmentation, retrieval of features, and classification. The majority of these algorithms made use of hand-crafted characteristics and traditional approaches to ML. For example, K-Means clustering [15] can retrieve color information from plant images to detect disease. This color-based information is further classified using

ML algorithms such as SVM [16]. When contrasted to modern DL systems, traditional methods frequently need considerable picture pre-processing. Picture scaling, denoising with a Gaussian or other smoothing filter, and others are examples of pre-processing. These pre-processing processes add time to the identification of disease workflow. Traditional approaches remain useful when the information is limited and the characteristics are well-defined. However, this approach results in the issue of information accessibility. Whereas, convolution neural networks emerged as better as they can better discriminate image characteristics. CNN can retrieve and process higher-level data such as texture, shape, size, etc. The emergence of CNN models [10–14] has improved the performance of the DL approach for better prediction of plant diseases.

1.1 Research gaps and motivation

Even though several approaches have been proposed for detecting plant diseases, notably rice plant diseases, there are still some major challenges to be overcome. Some of them are listed below:

- Using conventional image processing methods to discover and extract relevant information from plant leaves (such as rice plants) to distinguish the traits of various diseases is rather difficult. There occurs variation in these features, thus it is necessary to thoroughly investigate their patterns using a variety of datasets.
- The nature of the manually chosen handcrafted features is the only factor that influences how well the ML-oriented algorithms operate. Therefore, feature extraction has to be automated to choose and learn the best collection of features for classifying purposes.
- In DL models, equal weightage is assigned to all features extracted at each level of the model. To enhance classification efficiency, feature weightage needs to be updated at each level. This leads to a significant learning process.
- In most of the works, transfer learning is used such as VGG16, ResNet50, GoogleNet, etc. These DL models need millions of learning parameters that will increase the computational complexity for real-time deployment.
- To achieve higher feature generalization, DL networks must be trained on large data samples.

1.2 Key contribution and paper organization

Major of the research contribution is focused on plant disease detection. However, very few research contributions are presented for specific plant and their type categorization. In this paper, the main focus is to detect and identify the type of rice plant disease to integrate with precise farming. Therefore, the novel and main contributions of the paper are such:

- In this paper, a stepwise fine-tuning learning model is proposed with a loss exchange block for rice disease classification. This step-wise fine-tuning learning will exchange the information about the area of interest and a specific progressive loss aware (PLA) block will exchange the loss information with the next layer which will result in better convergence of loss. The base model is ResNet50 but it is finetuned with PLA block.
- Before training data augmentation is applied with the cGAN algorithm which will result in a more robust and accurate system.

- The PLAFTSL integrated with cGAN data augmentation models will enhance efficiency and can also handle the imbalance in input data.
- The PLAFTSL model shows results with and without cGAN augmentation. Along with that ablation, results are also shown with baseline models i.e., scratch learning, transfer learning, and fine-tuned learning.
- The result analysis is also presented under environmental complexities such as noise, blur, and camera rotation.

The rest of the paper is divided into four sections. Section 2 presented a discussion of the material and methods used to design the proposed model. The experimental setup and ablation study are presented in Section 3 of the paper. Section 4 describes the result analysis of comparative state-of-art models. Section 5 gives the discussion on the future scope with a conclusion.

2 Related work

The DL models are classified based on learning i.e., learning from scratch, learning from pre-trained models, and learning from the fine-tuned model. Some of the major contributions of scratch learning, transfer learning, and fine-tuned learning are presented in Table 1.

Berwal et al. [11] suggested a disease detection of tomatoes using the CNN model from scratch. The model is designed to classify 10 classes of diseases. The method has an accuracy of 90.0%. However, the model fails with data augmentation such as cropping. The model also degrades the quality of segmentation. Karthiks et al. [12] proposed a residual Convolution NN (R-CNN) model. The model was classified into four classes of tomato and cross-validation was also performed. Singh et al. [13] also proposed a multi-layer CNN model with scratch learning for the classification of mango leaf diseases. Similarly, Sambasivam and Opiyo [14] presented a CNN model for Cassava disease detection with an imbalanced dataset collected from Kaggle. Data augmentation was applied to handle data imbalance issues. However, the model was not efficient with low-resolution images. Garcia and Barbedo [10] designed a plant disease detection model using transfer learning. In this framework, the author used a pre-trained model i.e., GoogleNet for the prediction of plant disease level of infection. The model classified the disease severity into 4 classes. The model achieved an overall 94% accuracy. The benefit of using this model was that more samples of data would result in a better learning process for DL models. More homogeneous characteristics were obtained due to background removal before the learning process and it will result in a better detection rate. But still, it has some limitations such as the computational complexity of the model being quite high due to high dimensional data samples. Another issue with this model was that this model was unable to predict the categories of individual diseases. Chen et al. [19] used a MobileNetV2 framework which uses a pre-training technique integrated with a Classifier activation map (CAM) for the improvement of position visualization. But in this approach, twice training was performed increasing its computational complexities. The mobileNet-V2 pre-trained model was introduced with an attention mechanism by Chen et al. [23]. VGG-16 and GoogleNet pre-trained models were investigated by Yakkundimath et al. [25]. Ahmad et al. [17] proposed a fine-tuned model, MobileNet_V3 for plant disease detection. The model achieved good accuracy but resulted in quite high trainable parameters that needed to be focused on for a reduction in complexity. Jiang et al. [18] fine-tuned the VGG-Inception model to improve detection accuracy.

Table 1 Research Contribution for Plant Disease Detection

Ref	DT	DU	BM	MT	IA	CDC	FR	TL	TP	Accuracy
[10]	Plants	Self	GoogLeNet	Pre-trained	No	4	No	Yes	-	~94%
[11]	Tomato	Plant Village	CNN	Scratch	No	10	No	No	-	~98%
[12]	Tomato	Plant Village	Attention-ResNet	Scratch	Yes	4	No	No	-	~98%
[13]	Mango	Self	CNN	Scratch	No	4	No	No	-	~97%
[14]	Cassava	Kaggle	CNN	Scratch	Yes	5	No	No	-	~93%
[17]	Plants	Plant Village and Self	MobileNet	Fine-tuned	Yes	15	No	No	7.5 M	~99%
[18]	Apple	Apple leaf	VGG-INCEP	Fine-tuned	Yes	5	No	No	-	~78%
[19]	Plants	Plant Village	MobileNet-V2	Pre-trained	Yes	38	No	Yes	-	~99%
[20]	Rice	Self	CNN	Scratch	No	2	No	No	-	~95%
[21]	Rice	Self	CNN	Scratch	No	5	Yes	No	-	~98%
[22]	Rice	Kaggle	InceptionResNetV2	Fine-tuned	Yes	4	No	Yes	-	~95%
[23]	Rice	Self	MobileNet-V2	Pre-trained	Yes	38	No	Yes	-	~99%
[24]	Rice	Self	CNN	Scratch	No	4	No	No	-	~95%
[25]	Rice	Self	VGG16	Pre-trained	No	4	No	Yes	-	~92%
[26]	Rice	Self	VGG16	Fine-tuned	No	9	No	Yes	138 M	~97%
[27]	Plants	Plant Village	CNN	Fine-tuned	Yes	3	No	Yes	56 M	~96%
[33]	Rice	Kaggle	EfficientNet	Fine-tuned	No	3	No	Yes	-	~93%

DT = Disease Type, DU = Dataset Used, BM = Base Model, MT = Model Training, IA = Image Augmentation, CDC = Classification Disease Classes, FR = Feature Reduction, Transfer Learning, TP = Training Parameters.

But the model was object-oriented i.e., specific for apple disease detection. Narasimha et al. [22] fine-tuned the pre-trained model, InceptionResNetV2. The hyper-parameters were optimized to achieve better accuracy. Rahman et al. [26] performed a comparative analysis of transfer learning and fine-tuned learning models such as MobileNetv2, VGG16, InceptionV3, SqueezeNet, etc. It was observed that fine-tuned VGG16 outperforms better accuracy as compared to transfer learning. Singh et al. [27] introduced the fine-tuning of the CNN model with the replacement of a softmax classifier with an SVM and a random forest classification technique to enhance performance. This was done to reduce the trainable parameters. Therefore, optimization and reduction of the trainable parameter are also scopes for research. Aggarwal et al. [33] proposed a DL-based rice leaf disease classification model based on the EfficientNet model. Haridasan et al. [34] proposed a DL-based rice plant disease detection model and achieved 91% accuracy. According to these DL applications, it has been analyzed that learning from scratch requires more computational resources and increases the training time. Therefore, these are not suitable for deploying precise farming over cutting-edge technologies such as IoT, cloud, etc. It has been seen that transfer learning models give a good recognition rate and are more compatible with precise farming. But limitations of these pre-trained models are that they require high trainable parameters. Another issue created in transfer learning is negative learning. To overcome these issues, fine-tuned models are introduced that are more memory efficient and enable rapid development of precise or smart farming by reducing the training times and complexities. Another major issue created during the learning process is data imbalance issues, these issues can be handled by simply introducing feature engineering tools. Therefore, in this paper, we have presented a rice disease detection model for precise farming. The designed stepwise fine-tuned model interleaved with loss-aware blocks that can help in fast learning convergence with reduced complexities.

3 Materials and methods

In this paper, image processing tools and techniques are used to handle issues created in rice plant disease classification for precise farming. The methodologies suggested may be used for numerous industrial situations where efficient development of ML techniques is desired. Given that there is sufficient data for training, advanced CNN has previously done well on similar tasks. However, obtaining a large number of images for a given rice disease cannot always be feasible. Using data augmentation approaches might help CNNs understand the representative features of the various disease classes, we can overcome the problems of data shortage and class imbalance. Additionally, the data gathered in the field may have a variety of errors including background complexities and occlusion that lead to misclassifications. Additionally, because of their greater computational costs, high computational complexities, and complex architecture, large trainable parameters traditional CNNs are not suitable to support precise farming that is operated on cutting-edge technologies with limited resources.

3.1 Dataset description

The dataset was obtained from the two sources given in [26, 27, 32]. The first dataset was obtained from paddy fields maintained by the Bangladesh Rice Research Institute (BRRI) and includes 1426 images of rice diseases [26]. The next dataset was obtained from Kaggle (https://www.kaggle.com/datasets/minhhuy2810/rice-diseases-image-dataset/code.) and includes images of rice diseases. To collect as many images that were as relevant as was feasible, the image collection was done in a range of situations, including the winter season, summer season, and humid or rainy weather. The identities of the classes, together with the number of images collected for every classifier. Examples of rice diseases are presented in the appendix. Pests and diseases can be found in various sections of the rice plant. Temperature, humidity, rain, the species of rice plants, the season, nutrients, and other factors all have an impact on their occurrence. This research work has a total of twelve classifiers. The total number of images collected for each disease class is presented in the appendix.

3.2 Model overview

In this paper, we have presented a three-layered model, as presented in Fig. 1, for rice disease detection to support precise farming using the augmentation technique. Their layers are described below in sub-sections.

3.2.1 Pre-processing

The input rice disease images are resized in size 224 × 224 × 3. Each image in each disease type contains different samples. Then z-normalization is performed over all input image data. Normalized images are combined and divided into training and testing sets in the ratio of 70:30. To avoid overfitting of combined data during the learning process augmentation is performed over them.

3.2.2 Data augmentation

cGAN can be employed as a data augmentation method to improve the size of the dataset to save the system from overfitting. GAN uses conventional CNN layers to create an image

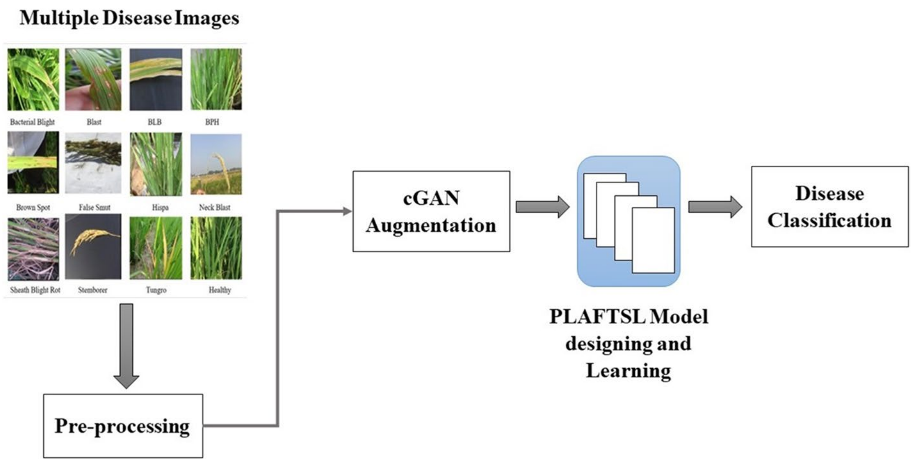


Fig. 1 System Architecture

matrix from noise. A Discriminator Model (DM) and a Generator Model (GM) constitute a Generative Adversarial Network. The objective of the generator is to create fake images, and the task of the DM is to determine which images are fake and which are real. The training of both the models is continuous and concurrent and they try to overcome one another. The DM ensures that the images produced by the GM are as similar as possible to the real ones. As discussed above this model has two adversarial models i.e. Discriminator and Generator. In which the first DM has eight layers along with four Conv2D. The first one is the Input Layer with [128 × 128 × 3], Second is the Embedded layer with [0, 3, 25] filter size followed by the Dense layer having a sigmoid function. The next layers are reshaped and concatenated layers having [64 × 64 × 3] filter size. Then this layer is followed by four CNN layers. In which every CNN layer ([32 × 32], [16 × 16], [8 × 8], [4 × 4]) is followed by the sixth layer i.e. Leaky ReLU, after the last sixth layer two layers flatten and dropout layer.

Similarly, the second model is the GM which has six layers with four convolutional layers. The first one is the input layer with [4 × 4 × 3] filter size followed by a dense layer having sigmoid activation. The third layer is the Embedded layer followed by four convolutional layers. Each layer has Leaky RELU an activation function with Reshape layer and Concatenate layer. cGAN uses generators which generate fake images which are compared by the discriminator.

Here, data, d with respective label l are used as discriminative input function with generator distribution of I_{gen} [28]. The GAN architecture is embedded with a discriminator layer to improvise the respective label assigning probability with a GM that generates the fake or synthetic images, f, as logarithmic function, $log\ D(d|l)$. Concerning the discriminator, the generator reduces the image dissimilarity loss using the logarithmic function, $\log(1 - \log D(G(f|l)))$. The objective function adopted in this case is the min–max function. In the suggested method, we created synthetic images of various diseases on rice plants using cGAN. We initially trained the cGAN model on rice plant images before using it to create synthetic images. Let's take the rice plant disease dataset $\mathcal{LL} = \{(d^{(n)}, l^{(n)})\}_{n=1}^{N}$, where $d(n)$ stands for a specific image and $l(n)$ for its related label $l(n) \in \{0, 1, \ldots, 9\}$. For cGAN training, the discriminator model gets a real rice plant image $l(n)$ and associated label $l(n)$, while the GM also receives a noise input and label $l(n)$. The GM then creates a false rice plant image. The discriminator model additionally receives the fake image that is currently being created. Fake or augmented images are compared with real images by discriminators. In this manner, the rice plant images were used to train the cGAN, and augmentation was applied over it. The fake or synthetic images are termed augmented images and can be used for further training models for disease diagnosis. Therefore, the primary reason for using cGANs for data augmentation in the paper is their ability to generate high-quality, targeted synthetic data. cGANs can produce specific and diverse samples by conditioning the generation process on certain inputs, such as class labels.

3.2.3 Progressive loss-aware fine-tuned stepwise learning (PLAFTSL)

In this paper, a progressive loss-aware stepwise fine-tuned learning model interleaved with a loss-aware stepping block is presented. Below in Fig. 2, the proposed model architecture is presented.

As illustrated in Fig. 2, step-wise learning of a fine-tuned model is presented in which some layers gradually freezeout with the convergence of the learning rate. The entire,

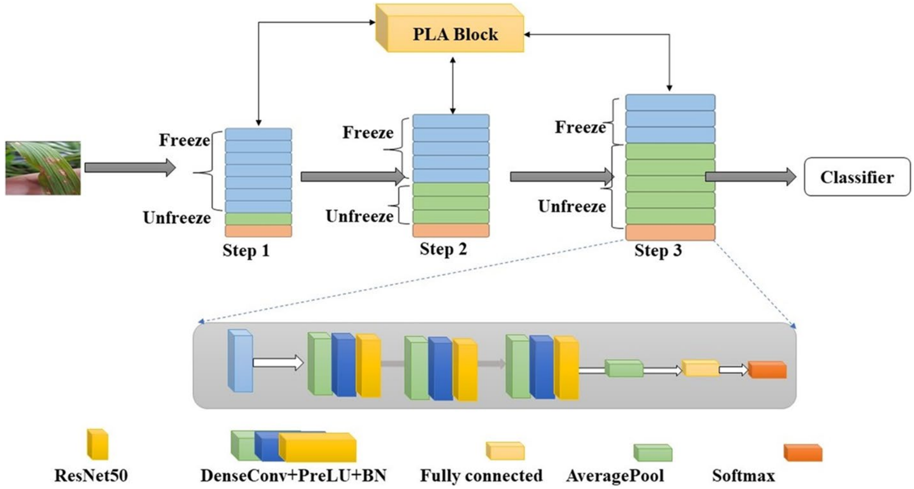


Fig. 2 The architecture of PLAFTSL Model

progressive loss aware fine-tuned stepwise fine-tuned learning (PLAFTSL) is divided into three steps. At each step, the model is trained and some pre-trained layers are frozen. In step 1, all layers of the pre-trained ResNet50 model are frozen and the last layer is left unfreeze. In the last layer, a fine-tuned dense layer is added as illustrated in Fig. 2. Then these layers are trained and then information is exchanged to the next step. The loss function is calculated at this step and its weight is passed to the PLA block. In step 2, some more layers are also unfrozen, and fine-tuning is again performed with this step, and the model is retrained and further information is also exchanged to the next step, and loss information is also shared with the PLA block. And finally, in step 3 some more layers are also unfreeze and fine-tuned layers are also added and the model is retrained again. And finally, the classifier is added to this layer that classifies the features to the respective class of disease. In PLAFTSL, the basic idea of freezeout layers was adopted from [31]. Training with frozen layers away from the classifier layers rather than initiating with all the layers set to trainable as in Freezeout [31]. Another point of distinction is that Freezeout focuses on faster training, while the proposed work seeks to identify the best strategy for exchanging information stepwise. Therefore, the network is progressive due to the presence of the PLA block. Stepwise transfer learning results in faster convergence and thus results in lower training time. By adding up the trainable parameters at all training steps and dividing them by the total number of steps, the accumulative trainable parameters can be calculated. However, stepwise fine-tuned learning will carry forward the pre-trained weights through freeze layers and only retraining is performed over unfreeze layers.

Therefore, concisely, it can be concluded that the PLAFTSL model presents the hierarchical, stepwise approach to fine-tuning a pre-trained deep neural network. It incorporates a Progressive Loss-Aware (PLA) mechanism that intelligently manages the training process across different layers of the network. The fine-tuning process of PLAFTSL is divided into three main steps, each with a specific focus on the layers being fine-tuned and the strategy for freezing and unfreezing layers:

Initialization Step:

- Freeze all layers of the ResNet50 model except the last layer.
- Add a dense layer to the last unfrozen layer for fine-tuning.
- Train these layers and calculate the loss, which is then communicated to the PLA block.

Intermediate Step:

- Unfreeze additional layers based on the PLA strategy.
- Perform further fine-tuning and training, with loss information exchanged with the PLA block.

Final Step:

- Unfreeze additional layers, incorporating fine-tuned layers as needed.
- Retrain the model, culminating in the addition of a classifier layer for disease classification.

The loss function employed in PLAFTSL is presented in Eq. (1) that is designed to address class imbalance and is applied progressively across the fine-tuning steps:

$$Loss_{Step} = \frac{1}{N} \sum_{i=1}^{I} \sum_{j=1}^{J} FC_{ij} \tag{1}$$

where, j = Number of loss functions, i = Number of layers in the model and FC_{ij} = Focal loss.

The Eq. (1) presents the progressive nature of the learning process, emphasizing the importance of managing loss at each step to mitigate the effects of class imbalance and to enhance the model's learning efficiency.

The pre-trained ResNet50 [29] is used for the fine-tuning model used for progressive stepwise learning. A Residual Neural Network (ResNet) is a type of ANN that builds networks by stacking residual blocks on top of one another and. ResNet is an abbreviation for Residual Network [29]. The Resnet 50 architecture includes a convolution with a kernel size of 7 × 7 and 64 distinct kernels, each with a stride size of 2, yielding one layer. Max pooling is next, with a stride size of 2. There is a 1 × 1,64 kernel in the next convolution, followed by a 3 × 3,64 kernel, and finally a 1 × 1,256 kernel. These three layers are repeated three times in total, giving us nine layers in this phase. After that, we observe a kernel of 1 × 1, 128 followed by a kernel of 3 × 3,128 and finally a kernel of 1 × 1,512. This phase was done four times, giving us a total of 12 layers. Then there's a 1 × 1,256 kernel, followed by 3 × 3, 256, and 1 × 1, 1024 kernels, which are repeated six times for a total of 18 layers. Then a 1 × 1,512 kernel was added, followed by two additional 3 × 3, 512, and 1 × 1, 2048 kernels, for a total of nine layers. Afterward when, we conduct an average pool and finish with a fully linked layer with 1000 nodes, followed by a softmax function, giving us one layer.

The output of ResNet-50's last convolution block is cascaded with layers of denseconv that include batch normalization with parametric ReLU [30] activation. For the classification of the data used, the network was retrained, and features were taken from the suggested fine-tuned framework and then further flattened. Using a progressive layer-wise loss function and parametric ReLU activation layer, fine-tuning was carried out. This loss function is developed to address the issue of class imbalance. Parametric ReLU is used for fine-tuning models since it does not have a vanishing gradient problem and allows for fine-tuning of the learning parameters on the learning rate. The loss function is described below:

3.3 Training details

The PLAFTSL network is designed for rice disease detection for multiple disease classification to support precise farming. The input image size is 224×224 which is used to train the model. In this paper, images collected from different sources [26, 27, 32] are used for training and purposes. For residual learning, the loss function is used to train the model, termed FC_{loss}. The minimum batch size of the training module was 64. Adam optimizer for training the models with a learning rate of 10^{-4}. The training is done for the 100 epochs. The designed framework is implemented in Python using Google Colab. The implementation is done in Keras, using TensorFlow as the backend. All networks are trained on the Tesla P100-PCIE GPU. To handle training overfitting, cGAN data augmentation is employed. Below in Table 2, learning hyper-parameters are presented.

4 Results and discussions

The validation of simulation experiments for the suggested model is presented in this section. This section offers a summary of the ablation study findings that focus on the performance of the suggested work. The advantages of each module employed in the proposed approach are demonstrated using the ablation study. As a result, we evaluated and compared the suggested PLAFTSL with and without augmentation in this section. Furthermore, baseline models are contrasted with the suggested PLAFTSL. Below in Eq. (1) to Eq. (5), performance parameters are discussed:

$$Accuracy = \frac{TrP + TrN}{TrP + TrN + FlP + FlN} \tag{2}$$

$$Precision = \frac{TrP}{TrP + FlP} \tag{3}$$

$$Recall = \frac{TrP}{TrP + FlN} \tag{4}$$

Table 2 Learning Hyperparameters

Parameters	Values
Input images	224×224
Batch size	64
Learning rate	10^{-4}
Epoch	100
Activation	Parametric ReLU
Environmental Conditions for Testing	Noise, blur, camera rotation

$$F1_{Score} = \frac{2 * precision * Recall}{precision + Recall} \quad (5)$$

where, TrP, FlP, TrN, FlN are True Positive, False Positive, True Negative, False Negative respectively.

4.1 Comparison of baseline models

In this sub-section, we have presented the result analysis of baseline models without the cGAN augmentation module. The testing results are compared in terms of accuracy, precision, recall, and f1_score. Three baseline models i.e., scratch model, transfer learning, and fined-tuned learning model are discussed here.

Baseline Model 1: The CNN residual learning is used from scratch initially having the combination of Convolution layer, batch normalization, and ReLU activation function. The loss function adopted for the baseline model is categorical cross entropy only at the final layer of the model.

Baseline Model 2: In this model, a pre-trained transfer learning model, such as the ResNet50 model is used. The loss function adopted for baseline model 2 is also categorical cross-entropy.

Baseline Model 3: In this model, fine-tuning of ResNet50 is performed. The output of ResNet-50's last convolution block is cascaded with layers of denseconv that include batch normalization with parametric ReLU activation function.

The results of these 3 baseline models are presented in Table 3. For the scratch CNN model, Accuracy is 85.7%, Precision is 87%, Recall is 86% and F-1 Score is 85%. For the Transfer learning model accuracy is 91.4%, precision is 93%, recall is 91% and F-1 Score is 92%. Similarly for the Fine-tuned model, these performance parameter is evaluated and it has a maximum percentage in terms of all the parameters. The fine-tuned model achieved an accuracy of 98.3%, 98% precision 97% recall, and 98% of F1_Score. The trainable parameters for three of the models are 9, 24, and 1.2 million respectively. From the table, it is concluded that the fine-tuned model achieved better accuracy with lower trainable parameters. Figure 3 shows the training accuracy and training loss for the different baseline models i.e. scratch model transfer learning and fined-tuned learning model. It is clear from the graph that the fine-tuned learning model has better accuracy than the remaining two models the fine-tuned learning accuracy is 98%. Similarly, all the model loss is also evaluated. From the graph, it is clear that in comparison with these three baseline models transfer learning model has the maximum loss while the fine-tuned learning model has the minimum loss.

Table 3 Ablation Study on Learning Models

Models	Accuracy	Precision	Recall	F1_score	Trainable Parameters
Scratch CNN	0.857	0.87	0.86	0.85	9 Million
Transfer Learning	0.914	0.93	0.91	0.92	24 Million
Fine-tuned Learning	0.983	0.98	0.97	0.98	1.2 Million

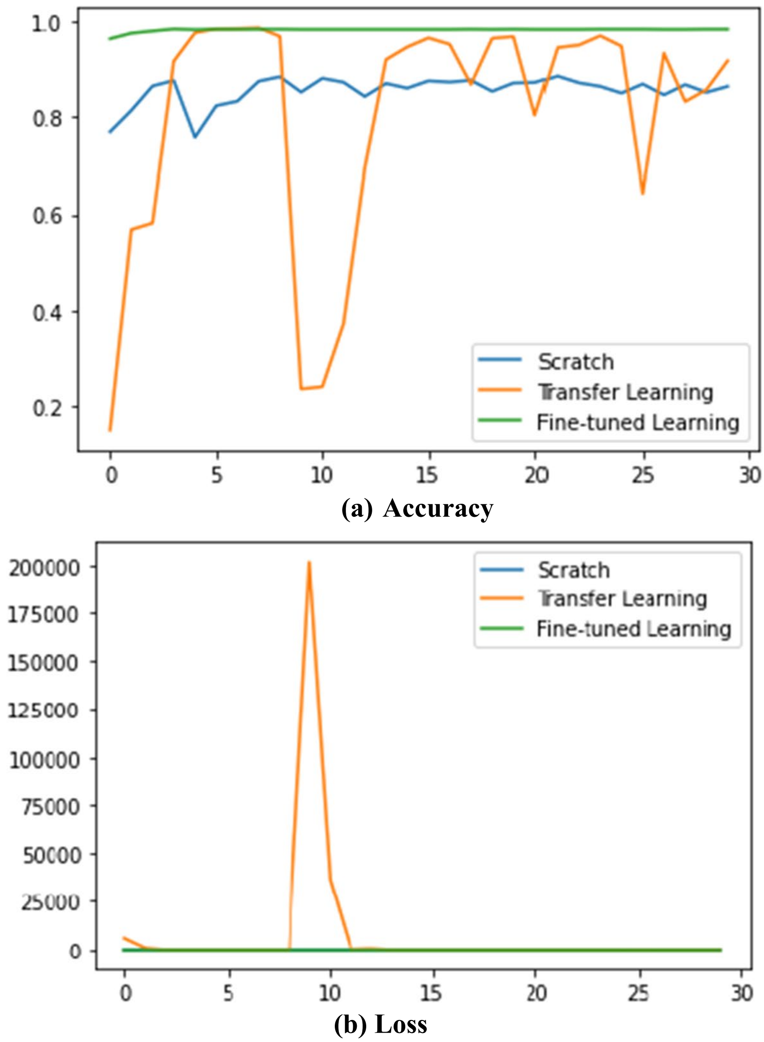


Fig. 3 Training Accuracy and Training Loss for Baseline Models

4.2 Comparison with augmentation

In this sub-section, we have presented the result analysis of baseline models with and without a cGAN augmentation module with a fine-tuned baseline model. The testing results are compared in terms of accuracy, precision, recall, and f1_score. Table 4 shows the model performance evaluation with and without augmentation for the designed model. Without augmentation, accuracy is 97.9%. The results are improved with augmentation for all parameters. With augmentation, accuracy is 98.4% (Table 5).

The training accuracy and training loss with and without augmentation are presented in Fig. 4. It is clear from the graph that with cGAN augmentation the system has better accuracy than without Augmentation. The accuracy is between 98 to 95% without cGAN and this is due to the presence of unequal count of different diseases as the dataset is

Table 4 Performance of Model with and Without Augmentation

Models	With cGAN	Without cGAN
Accuracy	0.984	0.979
Precision	0.98.5	0.98
Recall	0.98	0.98
F1_score	0.98	0.98

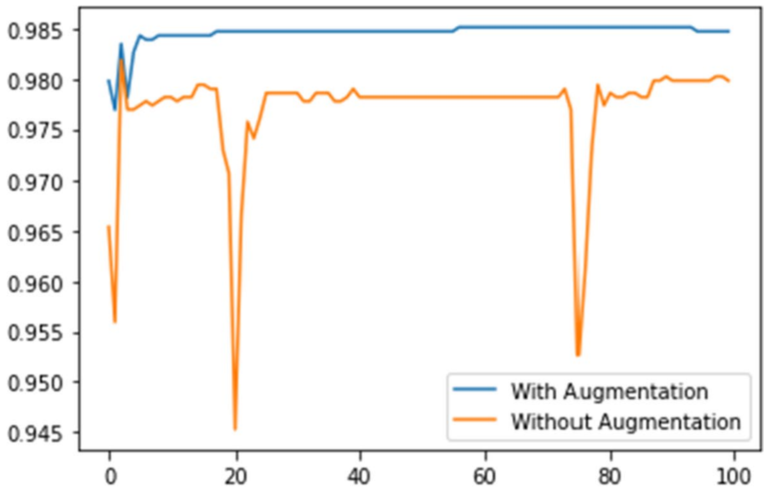
Table 5 Classification Performance with Disease Type

Disease	Accuracy	Precision	Recall	F1-score
BLB	98.30%	84.00%	91.00%	87.00%
BPH		95.00%	87.00%	91.00%
Brown_Spot		99.00%	99.00%	99.00%
False_Smut		88.00%	88.00%	88.00%
Healthy		92.00%	99.00%	95.00%
Hispa		91.00%	68.00%	78.00%
Neck_Blast		98.00%	100.00%	99.00%
Sheath_Blight_Rot		94.00%	91.00%	93.00%
Stemborer		99.00%	94.00%	96.00%
Tungro		100.00%	100.00%	100.00%
Blast		100.00%	99.00%	99.00%
Bacterialblight		98.00%	100.00%	99.00%

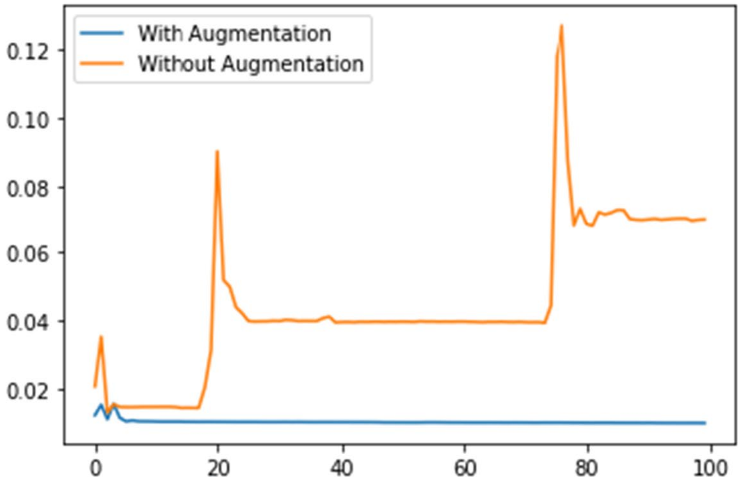
collectively prepared from different sources. Therefore, after augmentation, it is showing a smooth trendline for the learning process. Accuracy is 98.5% with cGAN augmentation. Similarly, for all loss functions, as presented in Fig. 4b, the model has almost constant loss and is also lowest with cGAN augmentation.

4.3 Result analysis and comparative state-of-art

The network performance of the proposed PLAFTSL model is presented in this section. Rice image datasets are used for testing the network performances. Below in Fig. 5a, the training and validation graph of the model is presented. Similarly, in Fig. 5b, the training and validation loss graph of the designed PLAFTSL model is presented. From the graph, it is observed that the PLAFTSL model has achieved approx. 98% of accuracy. The loss graph shows better convergence towards minimum loss due to the presence of a PLA block in the model. In Fig. 6a and b, we have presented the ROC curve and confusion matrix respectively for illustration of classification performance. The confusion matrix for the rice disease classification model across 12 classes shows that the model generally performs well, with a strong diagonal indicating correct classifications. However, some misclassifications are evident, as seen in the off-diagonal cells where the model confused one disease for another. However the high numbers on the matrix's diagonal suggest a relatively robust performance in disease identification. In Table 6, we have shown the result of testing samples in terms of accuracy, precision, recall, and f1_score. The results show that the model has achieved approx. 98% of accuracy. Figure 7 shows the feature visualization sample of the testing image by PLAFTSL model.



(a) Accuracy



(b) Loss

Fig. 4 Training Accuracy and Training Loss with and Without Augmentation

The classification performance of PLAFTSL on different disease types is presented in Table 5. The table presents an average accuracy of 98.30%. Figure 8 presents the testing outcome of some images by PLAFTSL. Table 6 presents the result analysis for disease detection under different environmental conditions. The result shows that the tested models exhibit the highest resilience to noise and are least effective in handling camera motion. Overall, these results underline the models' strengths and weaknesses in dealing with various environmental disturbances, highlighting particular robustness to noise and a need for improvement in handling camera motion.

Further, comparative state-of-art is also presented with some existing models i.e., MDFF [24], FT-VGG16 [26], and DL-RF [27]. MDFF [24] designed a hybrid CNN fusion model and achieved good efficiency but suffered from an overfitting problem

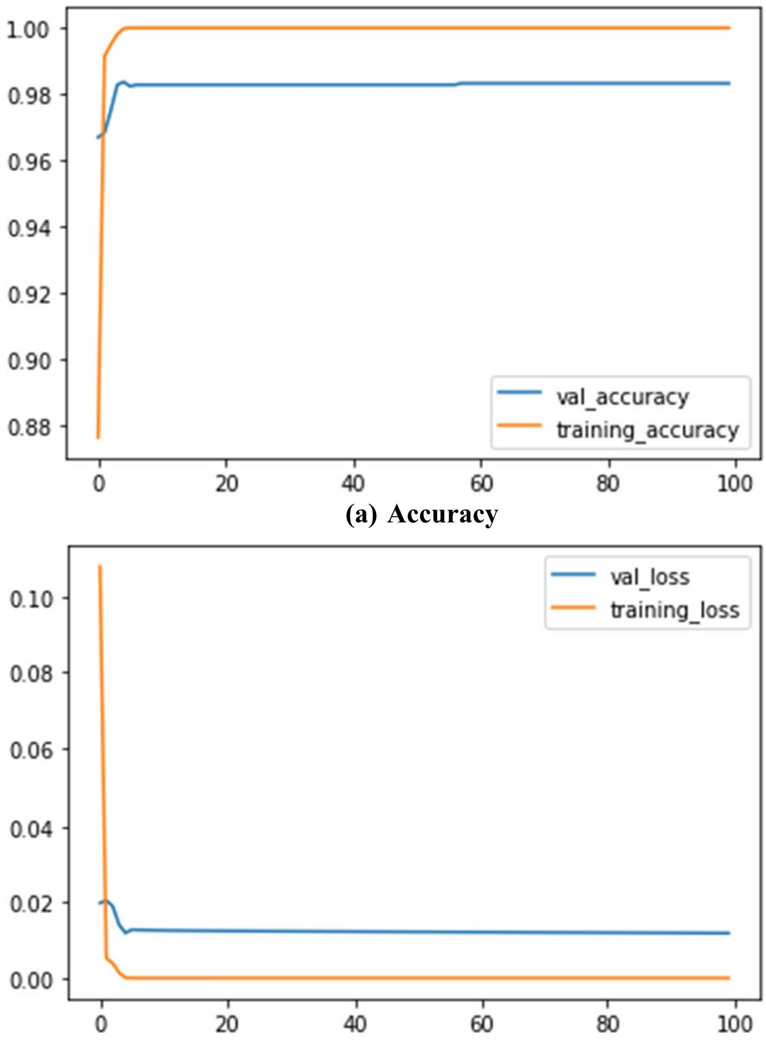
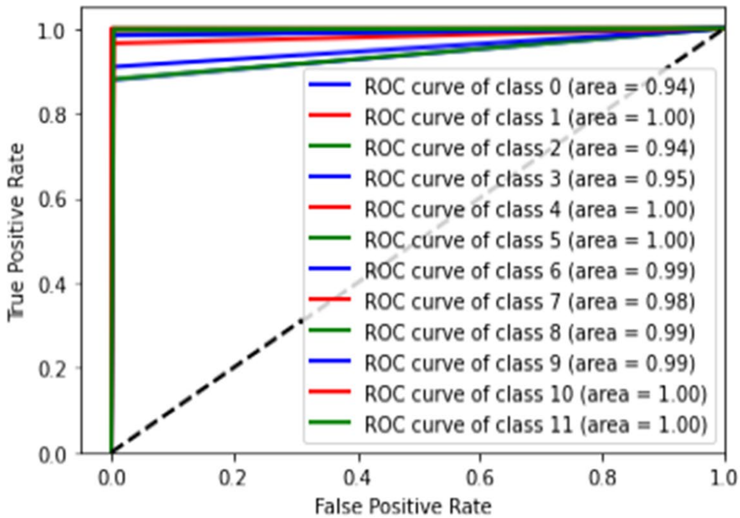
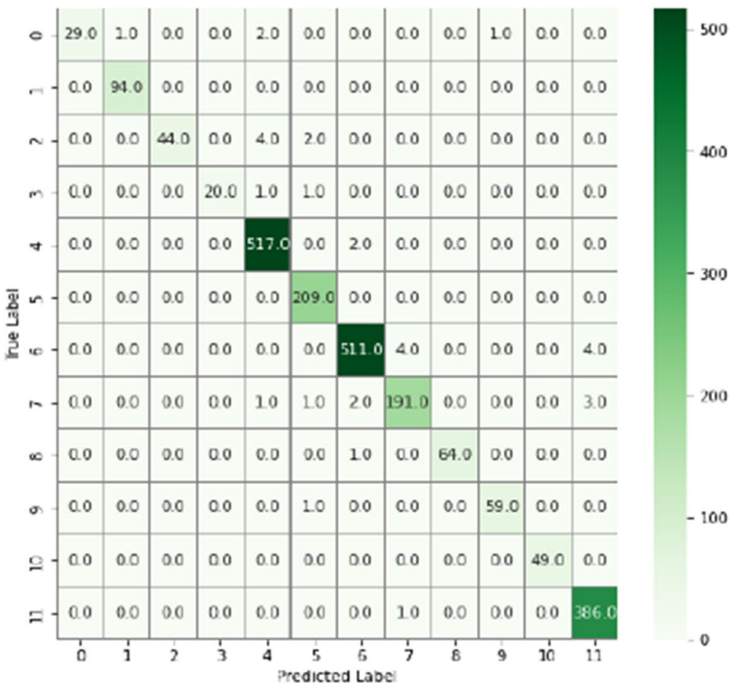


Fig. 5 Training and Validation Performance of PLAFTSL

during the training phase. FT-VGG16 [26] was a fine-tuned VGG16 model for rice disease detection. The model achieved an accuracy of about 97% but the dataset is considered very small, and images of different rice diseases look very similar, deep neural networks cannot be trained from scratch since the data is very small, and shallow feature extraction is produced around 80%. The trainable parameter was approximately 138 million. This results in the high computational cost of the model. In [27], the DL model was modified by changing the classifier layer of the model. Plant Village Dataset was used in this work and scratch learning was adopted. In this hybrid model, the voting strategy was used from five different models such as LeNet, AlexNet, MobileNet, ShuffleNet, and EffNet. The combination of these five models requires 56 million trainable parameters which makes its computational complexity very high. In [22], the Inception



(a) ROC



(b) Confusion Matrix

Fig. 6 Training and Validation Performance of PLAFTSL

CNN model was proposed and this model achieved approx. 95% of testing accuracy but the limitation of this model was that this model was designed for four types of rice plant diseases only taken from [22]. Whereas the computational complexity of this model was quite high.

Table 6 Testing Performance Under Environmental Conditions

Models	Noise	Camera Motion	Blur
Accuracy	97	92	96
Precision	94	89	95
Recall	92	88	89
F1_score	93	88	92

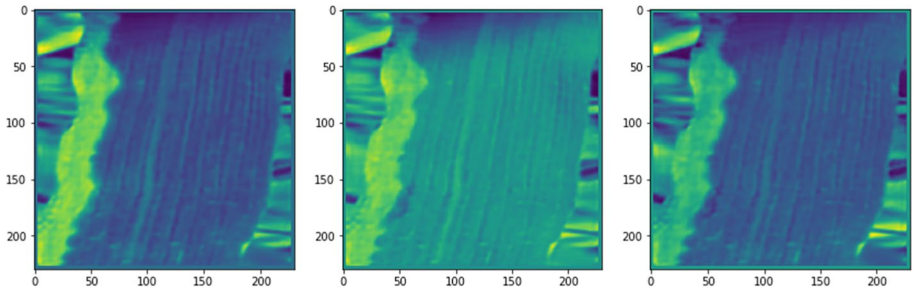


Fig. 7 Feature Visualization in PLAFTSL

Fig. 8 Testing Outcome Visualization of PLAFTSL

Table 7 shows the model performance evaluation for different benchmark techniques. For MDFF [24], accuracy is 95.31%, precision is 94%, recall is 96% and F1_score is 95%. For FT-VGG16 [26], accuracy is evaluated which is 97.12%. Similarly for DL-RF [27] model, accuracy is 96.1%, precision is 95.9%, recall is 88.6% and f1_score is 92.1%. For our proposed work, all these four performance parameters are evaluated and it has a

Table 7 Comparative State-of-art Performance

Models	Accuracy	Precision	Recall	F1_score	Trainable Parameters
TL [22]	95.64%	96.5%	96.5%	96.5%	64 million
MDFF [24]	95.31%	94%	96%	95%	-
FT-VGG16 [26]	97.12%	-	-	-	138 million
DL-RF [27]	96.1%	95.9%	88.6%	92.1%	56 million
PLAFTSL	98.32%	98%	96%	97%	1.2 million

maximum percentage in terms of all the parameters. The proposed model has achieved the highest accuracy of 98% of all. The results are maximum improved by 2.69% for accuracy, 5% for precision, 3% for recall, and 4% for f1_score. Therefore the PLAFTSL model's combination of high performance and low parameter count suggests a highly efficient model architecture that is easier to train and deploy, making it particularly appealing for real-world applications where computational resources might be limited. This analysis reveals that while high accuracy is crucial, it's equally important to consider the balance of precision and recall, the efficiency of the model (in terms of trainable parameters), and how these factors align with the specific needs and constraints of real-world applications. The PLAFTSL model, with its high accuracy, efficiency, and balanced performance, appears to be a strong candidate for practical deployment in detecting rice diseases, assuming its performance is consistent across diverse and unseen data.

5 Limitations of work

In this paper, the PLAFTSL model is presented for rice disease classification and achieved good accuracy. However this work can have the following limitations.

- As a large number of images are processed while training, therefore feature reduction needs to be applied.
- Another limitation of PLAFTSL is that with increasing environmental complexity, its efficiency is decreasing which needs to be focused on in the future.

6 Conclusions

In addressing the critical challenge of mitigating the adverse effects of rice diseases on crop yields, this study emphasizes the necessity of effective disease prevention and management strategies in rice agriculture. The basis of such strategies is the timely and accurate diagnosis of diseases, which facilitates the immediate deployment of targeted pesticide treatments. Traditional approaches to rice disease diagnosis have predominantly relied on manual observation of disease symptoms or the application of computer vision, alongside machine learning (ML) and deep learning (DL) techniques. Innovatively advancing the field, the authors of this paper propose a novel loss-aware stepwise learning approach specifically designed for the accurate diagnosis of rice diseases. This approach is anchored by an enhanced version of the ResNet50 model, which has been meticulously fine-tuned through a strategic process of layer freezing and unfreezing. This methodological refinement significantly streamlines the computational process and reduces the number of parameters requiring training, thereby optimizing the system's efficiency. The efficacy and reliability of the proposed approach are rigorously validated through extensive testing and training on a dataset collected from a variety of sources, demonstrating its superior capability in the accurate detection of rice diseases. Furthermore, the study comprehensively assesses the model's performance under diverse environmental conditions, underscoring its robustness and applicability in real-world scenarios. In the future, the designed model will be extended to real-time capturing of data and also find the factors that cause different types of rice diseases.

Appendix

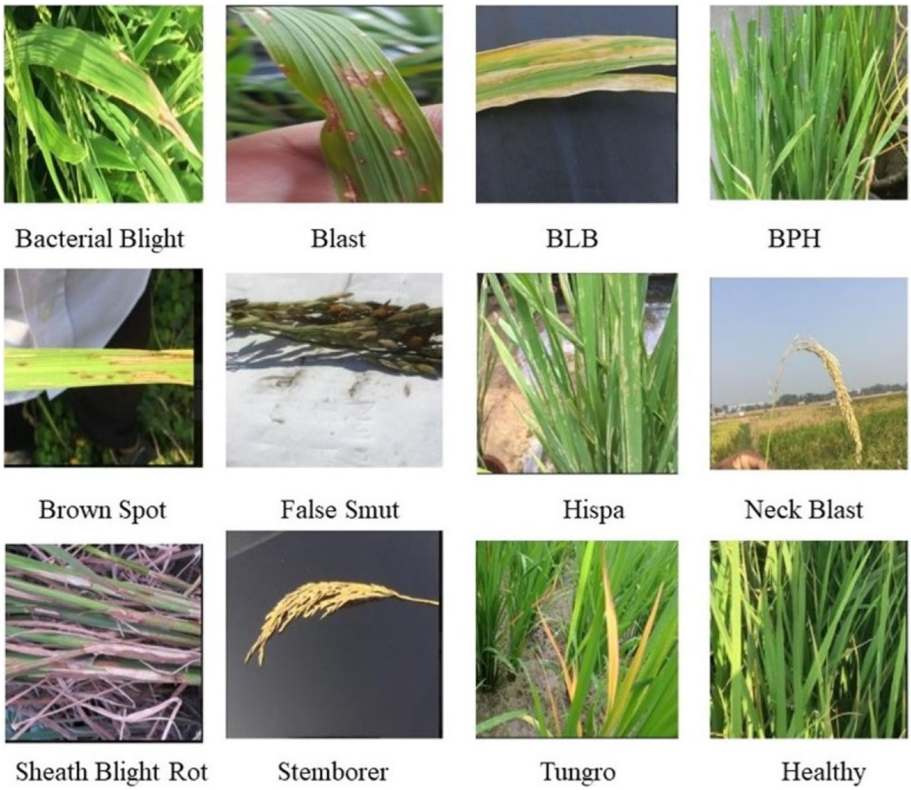


Fig. 9 Collected Rice Plant Disease Samples [26, 27, 32]

Table 8 Plant Disease Types and their Symptoms [8, 9]

Disease	Occurrence	Symptoms	Spread
Rice Blast	Pathogens	Diamond shaped lesions	Leaves, Collars, Seeds
Bacterial Blight	Pathogens	Small yellow or brown spots	Leaves
Brown Spots	Pathogens	Brown lesion	Leaves
False Smut	Pathogens	Chalkiness of grains (velvety smut ball)	Grains or seeds
Node Blast or Neck blast	Pathogens	Blackish blast spores	Ground part (leaf, collar, node, neck)
Stemborer	Insects	Tiny holes in tiller or stem	Stems
Hispa	Insects	Whitish leaves, Irregular white patches parallel to a mid vein in leaves	Leaves
Leaf Blast	Pathogens	White or gray lesion spots	Ground part (leaf, collar, node, neck)
Sheath Blight	Pathogens	Oval-shaped gray or green spots	Upper part (leaves)

Table 9 Dataset Description [26, 27, 32]

S.No	Class Name	No. of images
1	Brown Spot	634
2	Hispa	638
3	Leaf Blast	779
4	Healthy	1722
5	False Smut	93
6	BPH	71
7	BLB	138
8	Neck Blast	286
9	Stemborer	201
10	Sheath Blight and/or Sheath Rot	219
11	Tungro	1308
12	Bacterialblight	1584
	Total	**7673**

Abbreviations AI: Artificial Intelligence; ML: Machine Learning; DL: Deep Learning; PLAFTSL: Progressive Loss-Aware Fine-Tuning Stepwise Learning; cGAN: Conditional Generative Adversarial Network; FAO: Food and Agriculture Organization; WHO: World Health Organization; IP: Image processing; PLA: Progressive loss aware; R-CNN: Residual Convolution NN; CAM: Classifier activation map; BRRI: Bangladesh Rice Research Institute; DM: Discriminator Model; GM: Generator Model; ResNet: Residual Neural Network

Funding There is no financial support for this article.

Data Availability All data are made available in the manuscript.

Declarations

Conflict of Interest There are no conflicts of interest in this article.

References

1. FAO (2021) The impact of disasters and crises on agriculture and food security. https://doi.org/10.4060/cb3673en
2. FAO (2021) FAO - News Article: new standards to curb the global spread of plant pests and diseases. https://www.fao.org/news/story/en/item/1187738/icode/. Accessed 30 Jul 2022
3. Oerke EC, Dehne HW (2004) Safeguarding production—losses in major crops and the role of crop protection. Crop Prot 23:275–285. https://doi.org/10.1016/J.CROPRO.2003.10.001
4. DeChant C, Wiesner-Hanks T, Chen S et al (2017) Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. Phytopathology 107:1426–1432. https://doi.org/10.1094/PHYTO-11-16-0417-R
5. Picon A, Alvarez-Gila A, Seitz M et al (2019) Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. Comput Electron Agric 161:280–290. https://doi.org/10.1016/J.COMPAG.2018.04.002
6. Kim WS, Lee DH, Kim YJ (2020) Machine vision-based automatic disease symptom detection of onion downy mildew. Comput Electron Agric 168:105099. https://doi.org/10.1016/J.COMPAG.2019.105099
7. Chen J, Chen J, Zhang D et al (2020) Using deep transfer learning for image-based plant disease identification. Comput Electron Agric 173:105393. https://doi.org/10.1016/J.COMPAG.2020.105393
8. Tunio MH, Jianping L, Butt MHF, Memon I (2021) Identification and classification of rice plant disease using hybrid transfer learning. 2021 18th Int Comput Conf Wavelet Act Media Technol Inf Process ICCWAMTIP 2021 525–529. https://doi.org/10.1109/ICCWAMTIP53232.2021.9674124
9. Home - IRRI Rice Knowledge Bank. http://www.knowledgebank.irri.org/. Accessed 3 Aug 2022
10. Arnal Barbedo JG (2019) Plant disease identification from individual lesions and spots using deep learning. Biosyst Eng 180:96–107. https://doi.org/10.1016/j.biosystemseng.2019.02.002
11. Sharma P, Berwal YPS, Ghai W (2020) Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. Inf Process Agric 7:566–574. https://doi.org/10.1016/j.inpa.2019.11.001
12. Karthik R, Hariharan M, Anand S et al (2020) Attention embedded residual CNN for disease detection in tomato leaves. Appl Soft Comput J 86:105933. https://doi.org/10.1016/j.asoc.2019.105933
13. Singh UP, Chouhan SS, Jain S, Jain S (2019) Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease. IEEE Access 7:43721–43729. https://doi.org/10.1109/ACCESS.2019.2907383
14. Sambasivam G, Opiyo GD (2021) A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. Egypt Informatics J 22:27–34. https://doi.org/10.1016/j.eij.2020.02.007
15. Sannakki SS, Rajpurohit VS, Nargund VB (2013) SVM-DSD: SVM based diagnostic system for the detection of pomegranate leaf diseases. Adv Intell Syst Comput 174 AISC:715–720. https://doi.org/10.1007/978-81-322-0740-5_85/COVER

16. Jenifa A, Ramalakshmi R, Ramachandran V (2019) Classification of cotton leaf disease using multi-support vector machine. IEEE Int Conf Intell Tech Control Optim Signal Process INCOS 2019. https://doi.org/10.1109/INCOS45849.2019.8951356
17. Ahmad M, Abdullah M, Moon H, Han D (2021) Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning. IEEE Access 9:140565–140580. https://doi.org/10.1109/ACCESS.2021.3119655
18. Jiang P, Chen Y, Liu B et al (2019) Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. IEEE Access 7:59069–59080. https://doi.org/10.1109/ACCESS.2019.2914929
19. Chen J, Zhang D (2020) Nanehkaran YA (2020) Identifying plant diseases using deep transfer learning and enhanced lightweight network. Multimed Tools Appl 7941(79):31497–31515. https://doi.org/10.1007/S11042-020-09669-W
20. Lu Y, Yi S, Zeng N et al (2017) Identification of rice diseases using deep convolutional neural networks. Neurocomputing 267:378–384. https://doi.org/10.1016/J.NEUCOM.2017.06.023
21. Ramesh S, Vydeki D (2020) Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm. Inf Process Agric 7:249–260. https://doi.org/10.1016/J.INPA.2019.09.002
22. Narasimha NK, Prasad LV, Pavan Kumar CS et al (2021) Rice leaf diseases prediction using deep neural networks with transfer learning. Environ Res 198:111275. https://doi.org/10.1016/J.ENVRES.2021.111275
23. Chen J, Zhang D, Zeb A, Nanehkaran YA (2021) Identification of rice plant diseases using lightweight attention networks. Expert Syst Appl 169:114514. https://doi.org/10.1016/J.ESWA.2020.114514
24. Patil RR, Kumar S (2022) Rice-fusion: a multimodality data fusion framework for rice disease diagnosis. IEEE Access 10:5207–5222. https://doi.org/10.1109/ACCESS.2022.3140815
25. Yakkundimath R, Saunshi G, Anami B (2022) Palaiah S (2022) Classification of rice diseases using convolutional neural network models. J Inst Eng Ser B 1034(103):1047–1059. https://doi.org/10.1007/S40031-021-00704-4
26. Rahman CR, Arko PS, Ali ME et al (2020) Identification and recognition of rice diseases and pests using convolutional neural networks. Biosyst Eng 194:112–120. https://doi.org/10.1016/J.BIOSYSTEMSENG.2020.03.020
27. Singh AK, Sreenivasu SVN, Mahalaxmi USBK, et al. (2022) Hybrid feature-based disease detection in plant leaf using convolutional neural network, bayesian optimized SVM, and random forest classifier. J Food Qual 2022:. https://doi.org/10.1155/2022/2845320
28. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784
29. Sharma N, Jain V, Mishra A (2018) An analysis of convolutional neural networks for image classification. Procedia Comput Sci 132:377–384. https://doi.org/10.1016/J.PROCS.2018.05.198
30. Bingham G, Miikkulainen R (2022) Discovering parametric activation functions. Neural Netw 148:48–65. https://doi.org/10.1016/J.NEUNET.2022.01.001
31. Brock A, Lim T, Ritchie JM, Weston N (2017) Freezeout: accelerate training by progressively freezing layers. https://doi.org/10.48550/arxiv.1706.04983
32. Prajapati HB, Shah JP, Dabhi VK (2017) Detection and classification of rice plant diseases. Intell Decision Technol 11(3):357–373
33. Aggarwal Meenakshi et al (2023) Pre-trained deep neural network-based features selection supported machine learning for rice leaf disease classification. Agriculture 13.5:936
34. Haridasan Amritha, Thomas Jeena, Raj EbinDeni (2023) Deep learning system for paddy plant disease detection and classification. Environ Monit Assess 195.1:120

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Kamal Upreti[1] · Prashant Singh[2] · Dhyanendra Jain[3] · Amit Kumar Pandey[3] · Anjani Gupta[4] · Hare Ram Singh[5] · Santosh Kumar Srivastava[6] · Jay Shankar Prasad[5]

Kamal Upreti
kamalupreti1989@gmail.com

1 CHRIST(Deemed to be University), Delhi NCR Campus, Delhi, India

2 JSS Academy of Technical Education, Noida, India

3 ABES Engineering College, Ghaziabad, India

4 Department of CSE (AI & ML), PIET Engineering College Samalkha, Haryana, India

5 Greater Noida Institute of Technology, Greater Noida, India

6 Department of Applied Computational Science & Engineering, GL BAJAJ Institute of Technology & Management, Greater Noida, India