International Conference on Machine Learning and Data Engineering

Custom Convolutional Neural Network for Detection and Classification of Rice Plant Diseases

Sanasam Premananda Singh[a,*], Keisham Pritamdas[a], Kharibam Jilenkumari Devi[a], Salam Devayani Devi[a]

National Institute of Technology Manipur, Imphal, 795004, India

Abstract

Agriculture and its allied sector provide the livelihoods of vast rural areas in India. It also plays a vital role in the growth of developing countries like India. Rice, one of the primary components of Indian agriculture plays a significant role in the food security of the country and the world. However, it is stricken by many diseases that affect the quality and quantity of rice yield. This paper proposes a custom CNN architecture for detecting and classifying common diseases found in rice plants by reducing the number of parameters associated with the network. The proposed CNN architecture has been trained using a dataset of four types of common rice plant diseases. In addition, 1400 on-field images of a healthy rice leaf image dataset have been introduced in the paper for the detection of disease-free plants as well. Independent experiments were carried out with and without the inclusion of the healthy leaf image dataset. The performance of the proposed model is evaluated with the use of Stochastic Gradient Descent with Momentum (SGDM) and Adaptive Moment Estimation (Adam) optimization techniques using several performance matrices. Experimental results from the dataset for the classification of four rice plant diseases show that the model using SGDM optimization gives maximum accuracy of 99.66% and the model using the Adam optimization gives maximum accuracy of 99.83% on the test set in the 7th epoch respectively. However, the model with Adam optimizer performed better than the model with SGDM optimizer when the healthy leaf image dataset is included giving a maximum accuracy of 99.66% and 97.61% in the 7th epoch respectively.

Keywords: Adaptive moment estimation; Convolutional neural networks; Rice leaf diseases; Stochastic gradient descent with momentum.

* Corresponding author. Tel.: 7005235474;
  E-mail address: sanasam299@gmail.com

1. Introduction

Agriculture plays a very important role in the Indian economy contributing 20.2% of the total GDP in the year 2020-21. Rice, the country's staple food, is the most planted crop, providing a livelihood for millions and playing a critical part in the food security of the country. Rice yields are significant not just for India's economy, but also for the rest of the globe, as India is the world's second-largest producer and largest exporter of rice [1].

With 70% of Asians consuming mostly rice, which is more than half of the world's population, rice is the most popular food grain worldwide [2]. Asian countries also account for 90% of the total global production of rice [3]. However, rice plants are stricken by several diseases caused by bacteria, fungi and viruses. The outbreak of these diseases causes a loss of about 10-15% of rice yield in India [4]. Early diagnosis and treatment of these diseases are very important as it improves the quantity and quality of yield, thereby reducing the use of pesticides. Detection of these diseases by manual observation may be a complicated, subjective, and tedious method. As a result, there is a need for automated systems that assist farmers in monitoring crops at every stage of growth in order to satisfy the demands of a growing population.

In this paper, a new custom CNN architecture is proposed for the detection and classification of rice plant diseases. Numerous experiments have been performed to modify the model's parameters for the accurate identification of the diseases that occur in rice plants. The custom CNN models were developed using a series of convolutional, batch normalization, activation, and pooling layers, followed by fully connected and softmax layers. The proposed CNN architecture significantly decreases the number of parameters used by choosing smaller kernel sizes and reducing the number of layers in the network. A dataset of four types (Bacterial Blight, Blast, Brownspot and Tungro) of diseased rice leaves of 5932 images collected from Sambalpur and Bargarh, Orissa, India [5] has been used to train the network. In addition,1400 healthy image samples collected from the Imphal East District of Manipur were also used in the study for the successful detection of diseases free plants. The healthy leaf image dataset collected is visually verified by the Indian Council of Agricultural Research, Lamphel, Imphal, India. The proposed model has the benefit of having a substantially smaller number of parameters and allowing it to be used on memory-constrained devices. Also, the proposed method could be a useful tool for farmers for detecting and classification of common diseases that occurs in rice plants enabling them to take preventive measures in time. Image samples of each class of the dataset are shown in Fig. 1 and the detailed description of the classes are as follows:

a)   Bacterial Blight

Bacterial blight is a deadly bacterial disease caused by *Xanthomonas oryzae pv. oryzae*. In severe epidemics, crop loss can be as excessive as 75%. The disease thrives in a warm and humid environment. It causes seedlings to wilt, as well as the yellowing and drying of leaves.

b)   Blast

Rice blast, caused by the fungus *Magnaporthe oryzae* (anamorph: *Pyricularia oryzae*), is often recognized as the world's most devastating rice disease due to its widespread distribution and destructiveness under ideal conditions. It occurs in locations with low moisture content in the soil, frequent extended periods of rain, and low daytime temperatures. It is also one of the most destructive diseases in Manipur, giving a loss of about 60 to 100% [6].

c)   Brownspot

Brown spot is caused by the fungus *Cochliobolus miyabeanus*. The primary source of the infection is the contaminated seed. The disease starts as a small brownspot, and then progresses to cylindrical, oval, and circular forms. It infects the leaves, coleoptile, panicle branches, leaf sheath, spikelets, and glumes. The disease thrives in a warm and humid environment.

d)   Tungro

Rice tungro disease is caused Rice tungro spherical virus (RTSV) and Rice tungro bacilliform virus (RTBV). The leaves turn yellow, and the discolouration begins at the tip of the leaf and extends to the bottom of the leaf. Tungro virus is transmitted from one plant to another by leafhoppers that feed on plants infected with the disease.

e)   Healthy

Healthy leaves are free from any kind of disease. Unlike the infected leaves, the healthy leaf has a uniform colour distribution. A dataset of healthy rice leaves has been included in the training of the models for the detection of disease-free plants.

The rest of this work is structured as follows. The second section describes relevant research in the field of plant disease detection. The proposed methodology for detecting and classifying various diseases in rice plants is provided

in section 3. Section 4 goes over the experimental setup. Section 5 presents experimental results. Finally, section 6 concludes the work.
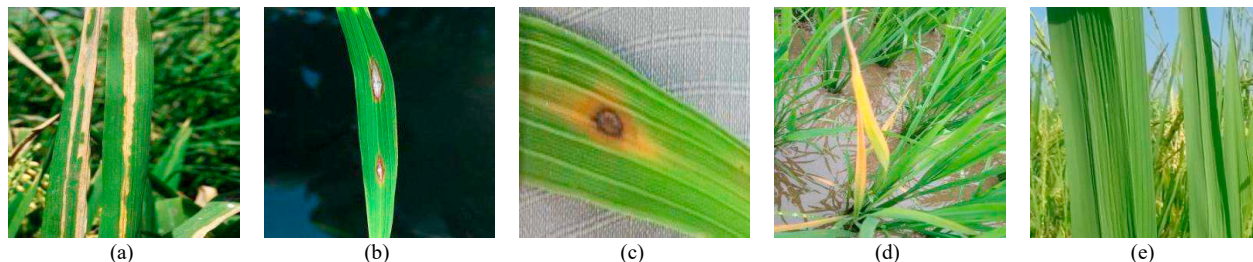

Fig. 1. Image samples of all the rice plant diseases and healthy plants in the dataset (a) Bacterial Blight (b) Blast (c) Brownspot (d) Tungro (e) Healthy leaves

2. Literature Review

Many studies on the automatic diagnosis of rice disease based on image processing and machine learning have been published. Pugoy et al. [7] design a system that can identify areas containing anomalies using histogram intersections and threshold-based clustering algorithms to isolate outlier areas into separate clusters where each cluster contains colours related to one another. Orillo et al. [8] developed a program to identify diseases present in rice leaves by image processing combined with backpropagation neural networks. Also, typical lesions of diseased leaves were successfully identified with 100% accuracy for disease identification. Pavithra et al. [9] extracted colour texture features using K-means segmentation, which was then fed into SVM and ANN classifiers. The results reveal that the SVM classifier outperforms the ANN classifier. Prajapati et al. [10] the background is removed by applying a threshold mask to the saturation component of the original image in HSV colour space. K means clustering has been used for disease segmentation. A total of 88 features were extracted and three different models with different numbers of features were examined. Diseases were classified using SVM, achieving 93.33% training accuracy and 73.33% test accuracy.

Islam et al. [11] described a new method for detecting and diagnosing rice diseases that relies solely on the percentage of RGB values in the affected area. Furthermore, the naive Bayes classification method has simplified the technique. Chawal et al. [12] prototyped a system to classify rice leaf diseases using K means clustering for segmentation and the Twin Support Vector Machine (TSVM) for classification. The disease detection system provides 95% accuracy in the test set. Ahmed et al. [13] presented a comparative study of four machine learning algorithms in the field of rice leaf disease detection, including ANN, decision tree, logistic regression, and Naive Bayes. The decision tree outperforms the other algorithms with an accuracy of 97% on the test data. Sethy et al. [14] compare the performance of 13 pre-existing CNN models using transfer learning and a deep feature plus SVM approach. The resnet50 plus SVM technique yields the greatest F1 score of 0.9838, however, the transfer learning technique fails to produce a satisfactory result. While machine learning approaches have achieved significant advances in picture classification, they do have some limitations, such as restricted data processing capabilities and the need for segmentation and feature extraction [15]. Deep learning techniques solve these restrictions by eliminating the need for pre-requisite operations like segmentation and feature extraction. In addition, the deep learning-based model often outperforms contemporary machine learning techniques by achieving higher accuracies [16].

In recent years, CNNs have been used in a variety of areas such as object detection, image classification, video classification, and document processing [17-19]. Aside from rice plant disease detection, it is also used to classify diseases in other major crops such as maize, wheat, potato, and cotton [20-23], as well as various fruits and vegetables such as apple, mango, cucumber, and tomato [24-27]. CNN is also utilized in medical diagnosis, including pulmonary nodule detection on computed tomography images [28], classification of skin diseases [29] and acute leukaemia is diagnosed using microscopic pictures of a peripheral blood smear [30].

3. Materials and methodology

This section gives the details of the dataset and the methodology of the proposed CNN architecture in the study.

3.1. Dataset

The first dataset includes 5932 images of infected rice leaves, including brownspots, tungro, blast, and bacterial blight. The images were taken in the rice fields in western Orissa with the Nikon DSLRD5600. With an 18-55mm high-resolution lens [17]. The second dataset contains 1400 images of healthy rice leaf that has been collected from the Imphal East District, Manipur. The images were captured using SONY HDR PJ540 with a 26.8 mm wide-angle (G Lens). The healthy leaf images captured are taken to the Indian Council of Agricultural Research, Lamphel, Imphal and visually verified.

All the training images are augmented with simple image rotation and image flip operations (90-degree rotation, 180-degree rotation, 270-degree rotation, vertical flipping, horizontal flipping) to all images in the training set before the training begins. Therefore, the number of images increased by a factor of 6, including those original images as shown in Table 1.

Table 1. Dataset

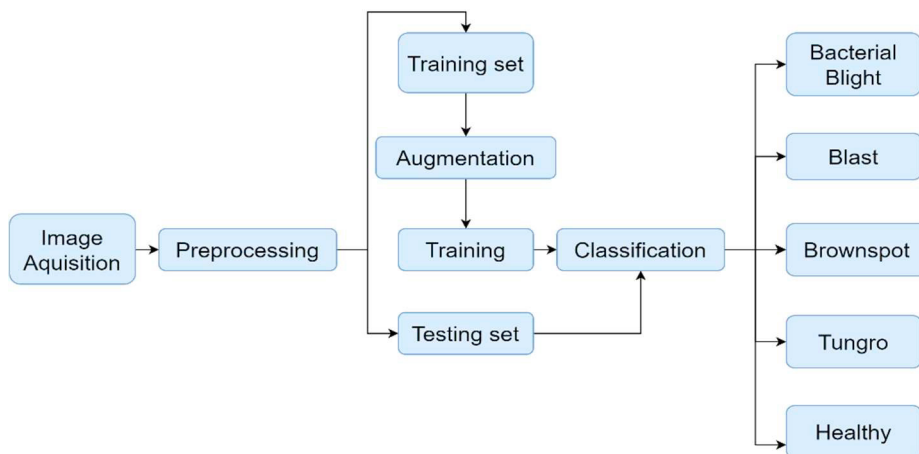| Classes	Original Images	Training Set	Augmentation	Testing Set
Bacterial Blight	1584	1267	7602	317
Blast	1440	1152	6912	288
Brownspot	1600	1280	7680	320
Tungro	1308	1046	6276	262
Healthy	**1400**	**1120**	**6720**	**280**
Total (Original)	5932	4745	28470	1187
New Total	**7332**	**5865**	**35190**	**1467**

3.2. Methodology

Fig. 2. Proposed Methodology.

A Convolutional Neural Network-based architecture is being developed to detect and classify rice plant diseases efficiently. The block diagram of the proposed methodology can be seen in Fig. 2. The pre-processing step of the methodology includes cropping the images to remove unwanted backgrounds and resizing all the images to 256X256 pixels. Following the pre-processing step, each class of the dataset is randomly split into the 80:20 proportion for training and testing. Then, the training samples are augmented to increase the number of images before the training process.
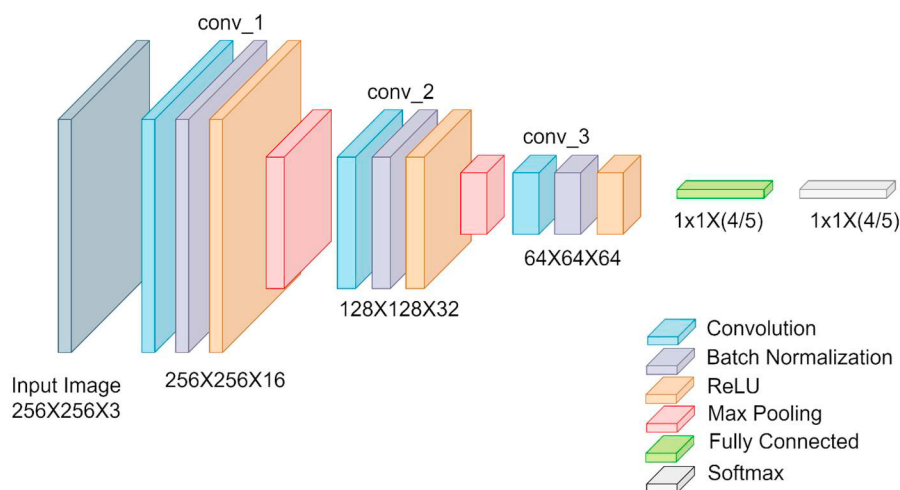
Fig. 3. Proposed custom CNN architecture.

Table 2. Network analysis (kernel size: 3X3, with the inclusion of healthy leaf image dataset (5 classes)).

S. No.	Name	Activations	Learnables	Total Learnables
1	imageinput	256X256X3	-	0
2	conv_1	256X256X16	Weights 3X3X3X16 Bias 1X1X16	448
3	batchnorm_1	256X256X16	Offset 1X1X16 Scale 1X1X16	32
4	relu_1	256X256X16	-	0
5	maxpooling_1	128X128X16	-	0
6	conv_2	128X128X32	Weights 3X3X16X32 Bias 1X1X32	4640
7	batchnorm_2	128X128X32	Offset 1X1X32 Scale 1X1X32	64
8	relu_2	128X128X32	-	0
9	maxpooling_2	128X128X32		0
10	conv_3	64X64X64	Weights 3X3X32X64 Bias 1X1X64	18496
11	batchnorm_3	64X64X64	Offset 1X1X64 Scale 1X1X64	128
12	relu_3	64X64X64	-	0
13	fc	1X1X5	Weights 5X262144 Bias 1X5	1310725
14	softmax	1X1X5	-	0
15	Classoutput	1X1X5	-	0
	Total parameters			1334533

Convolutional Neural Networks are multilayered network architectures widely used to solve various image classification problems. The combination of properties of CNN such as parameter sharing and sparsity of connection has significantly reduced the number of parameters as compared to the traditional deep neural network. The proposed CNN architecture consists of a series of convolutional layers, batch normalization, pooling, activation and followed by a fully connected and softmax layer. The proposed custom CNN architecture and the number of parameters associated with the layers are shown in Fig. 3 and Table 2 respectively. The detailed description of the layers and their functions are as follows:

1.   Convolution Layer
This is the first layer of the neural network which is the core building block of the architecture. A convolutional layer has a multiple feature detector known as kernel or filters which will be dragged by the stride across the entire image to check for the presence of the particular feature. An example of a convolution operation without padding is

shown in Fig. 4. The proposed architecture uses three convolution layers namely conv_1, conv_2, and conv_3 with 16, 32 and 64 kernels respectively.
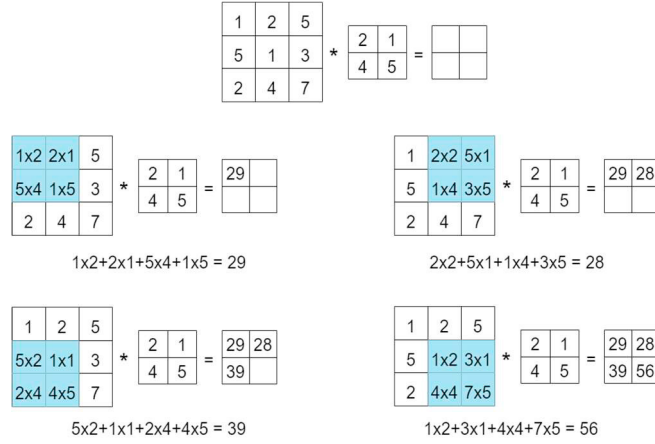

Fig. 4. An example of convolution operation without padding.

2. Batch Normalization Layer

Batch normalization is an approach for training very deep neural networks that standardize the inputs to layers for each mini-batch. As a result, the learning process is stabilized and deep network training requires a much smaller number of training epochs. [31].

Consider a mini-batch $B = \{x_{1...m}\}$. The elements x_i are normalized by calculating the mini-batch mean μ_B and variance σ_B^2. The normalized activation is calculated as follows

$$\hat{x} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon_B}} \tag{1}$$

ϵ_B is added to the mini-batch variance for numerical stability. A default value of ϵ_B is 10^{-5}. The batch normalization operation further shifts and scales the activation using the transformation

$$y_i = \gamma_B \hat{x}_i + \beta_B \tag{2}$$

where the scale γ_B and offset β_B are trainable parameters. The proposed architecture uses three batch normalization layers with 32, 64, and 128 parameters respectively as shown in Table 2.

3. Max Pooling Layer

The primary function of the pooling layer is to reduce the dimension of the output of the given convolutional layer which reduces the computational complexity in the further layer. Max pooling is by far the most popular technique used for pooling. It divides the image into sub-region rectangles defined by the kernel and only returns the maximum value of each kernel. The proposed architecture uses a max-pooling operation with kernel size 2X2 and a stride of 2. An example of the mentioned max-pooling operation can be seen in Fig. 5.
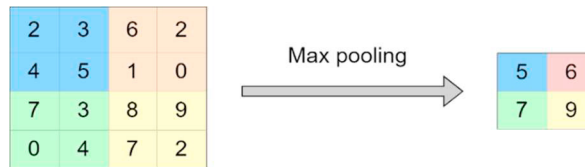

Fig. 5. An example of max-pooling operation.

4. Activation Function Layer

Following the pooling layer is the activation function layer. This layer introduces non-linearity into the neural network. Using non-linearity, the generated output of the previous layer can be modified or cut off. This is used to limit the output produced. Recent works in CNN often use Rectified Linear Unit (ReLU) as the activation function for the following reasons:

$$g(z) = max\ (0, z) \qquad (3)$$

$$\frac{d}{dz}g(z) = \begin{cases} 1, & if\ z > 0 \\ 0, & otherwise \end{cases} \qquad (4)$$

    a) Simplicity in both function and derivative as shown in (3) and (4).
    b) Reduced the likelihood of vanishing gradient occurring.
    c) ReLU produces a sparser representation. This is due to the fact that zero in the gradient results in a complete zero. [32].

5. Fully Connected Layer

The fully-connected layer is structured similarly to how neurons are organized in classic neural networks. As a result, every neuron in the completely linked layer is directly coupled to every neuron in the previous and subsequent layers. The fully connected layer's main disadvantage is that it has a large number of parameters that require extensive computation during training [32]. Hence, in the proposed model only one fully connected layer has been used. The fully connected layer has a total of 1310725 parameters in the model that includes the healthy image dataset as can be seen in Table 2. Finally, the neural network output is sent to the softmax activation function, which turns the real-valued output of the fully connected layer into a probability distribution.
.

3.3. Training the proposed custom CNN models

When training a CNN model, the weights within the model are getting updated. Arbitrary weights are assigned to each connection between the neurons. These weights are constantly updated during training session to achieve optimal values. The optimization of the weights depends on the optimization algorithm chosen for the model. The Stochastic Gradient Descent (SGD) with momentum and Adaptive Moment Estimation (Adam) optimizers are two of the most extensively used optimizers. Both optimizers are based on the exponentially weighted moving average.

1. Stochastic Gradient Descent with Momentum (SGDM)

Momentum is a way to help accelerate the SGD in the right direction and dampen oscillations by calculating the exponentially weighted average of the gradients and then updating the weights using that gradient instead [33]. It is accomplished by adding a fraction γ of the update vector of the preceding step to the current update vector. The weight(w) update rule for stochastic gradient descent with momentum is calculated using (5)

$$w_{t+1} = w_t - \alpha\nabla J(w_t) + \gamma(w_t - w_{t-1}) \qquad (5)$$

where t is the iteration number, α is the learning rate and $\nabla J(w_t)$ is the derivative of the cost function. The momentum term γ is commonly set to 0.9 for various deep learning applications.

2. Adaptive Moment Estimation (Adam)

Adaptive Moment Estimation (Adam) [34] algorithm updates the gradient and squared gradient of the exponentially moving average with the hyper-parameters $\beta_1, \beta_2 \in [0,1)$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla J(w_t) \qquad (6)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)[\nabla J(w_t)]^2 \tag{7}$$

where $\nabla J(w_t)$ is the derivative of the cost function. The moment estimates of the moving average are skewed towards zero since they are initialized as vectors of zeros. The author of Adam counteracts these biases by computing bias-corrected estimates $\hat{m}$ and $\hat{v}$.

$$\hat{m} = \frac{m_t}{1 - \beta_1^t} \tag{8}$$

$$\hat{v} = \frac{v_t}{1 - \beta_2^t} \tag{9}$$

Then, the weights are updated using the following update rule in (10)

$$w_{t+1} = w_t - \frac{\alpha.\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{10}$$

4. Experimental Setup

In this study, experiments were conducted on LENOVO 80XL with an Intel(R) Core (TM) i5-7200U CPU @ 2.50GHz (4 CPUs), running Windows 10 Pro 64-bit. The MATLAB 2021b deep learning toolbox was used to carry out the experiments. The performance of the model is measured in accuracy, recall, precision, F1 score and elapsed time.

The proposed CNN architecture uses 3X3 and 5X5 convolution filters with 16, 32, and 64 filters for conv_1, conv_2, and conv_3, respectively. Max pooling of 2X2 and stride 2 is performed. The experiment used a mini-batch size = 128 and epoch = 7. Both the SGDM and Adam optimizer has been used with $\beta_1 = 0.9, \beta_2 = 0.999, \alpha = 0.01$ and $\varepsilon = 10^{-8}$.

5. Results and discussions

5.1. Evaluation on the dataset of four classes of rice plant diseases

Experiments were conducted using the dataset of four classes of rice plant diseases. A total of four experiments were carried out by varying the kernel size with the SGDM optimizer and Adam optimizer. The comparison of validation accuracy during the training process of SGDM-based and Adam-based models in Fig. 6 shows that it is faster to train the Adam-based model as it achieves higher validation accuracy in the earlier epoch. The confusion matrices in Fig. 7 and Fig. 8 shows the performance of the models on the test set for various experiments conducted on the dataset of four rice leaf diseases. Then, the performance of the model is expressed in Table 5 as

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

where, $TN = true\ negative, TP = true\ positive, FP = false\ positive$ and $FN = false\ negative$.
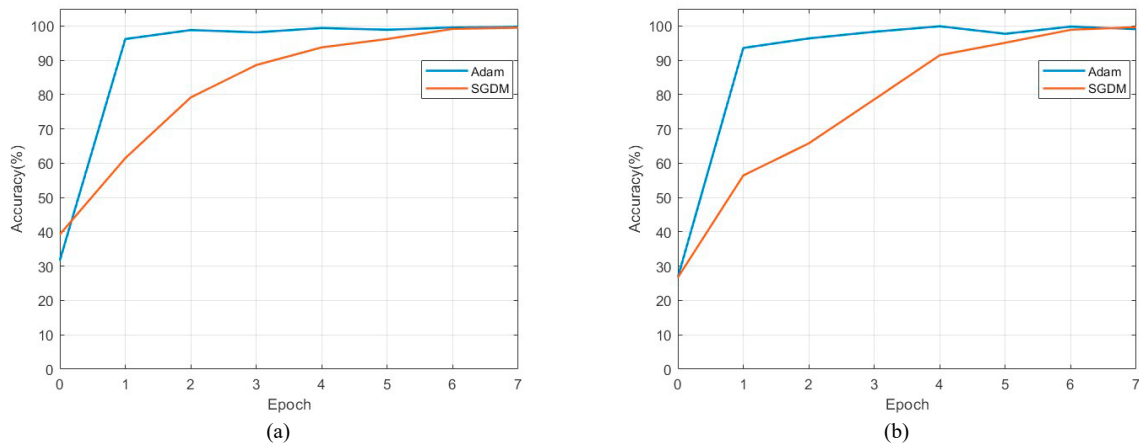
Fig. 6. Validation accuracy comparison of SGDM and Adam optimization with variation in the kernel size during the training process for (a) kernel size: 3X3 (b) kernel size: 5X5.
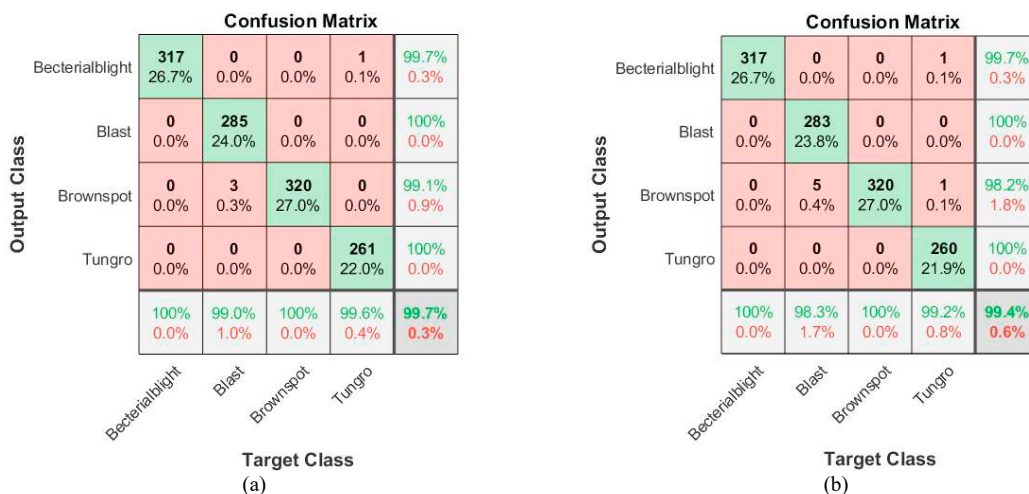

Fig. 7. Confusion matrices experimental results on the test set with SGDM optimization (a) kernel size: 3X3 (b) kernel size: 5X5

From the confusion matrix in Figure 7(a) it can be observed that, out of the 288 test images of blast, 3 images are misclassified as brownspot and 1 image out of the 262 test images of tungro is misclassified as bacterial blight while no case of misclassification in the case of bacterial blight and brownspot test images. In Figure 7(b), 5 test images of blast is misclassified as brownspot and 2 test images of tungro are misclassified as bacterial blight and tungro respectively. Similar cases of misclassification can be observed in the confusion matrices in Figure 8. The misclassification is caused by color resemblance between the classes.
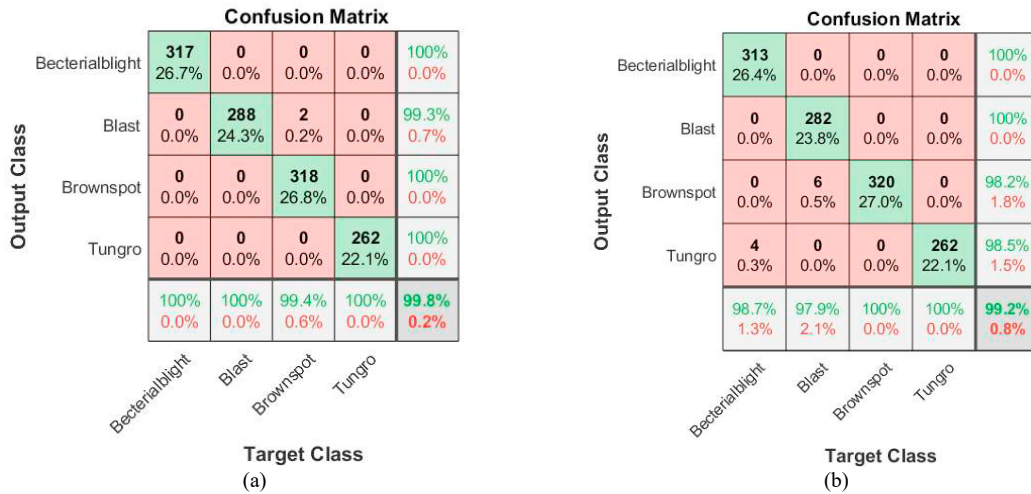
Fig. 8. Confusion matrices experimental results on the test set with Adam optimization (a) kernel size: 3X3 (b) kernel size: 5X5

Table 3. Experimental results of the proposed model on the dataset of four classes of rice plant diseases

Ex. No.	Kernel size	Optimizer	Accuracy (%)	Precision	Recall	F1 Score	Elapsed Time
1	3X3	SGDM	99.66	0.9969	0.9964	0.9967	4 hr 47 min 57 sec
2	5X5	SGDM	99.41	0.9946	0.9938	0.9942	6 hr 25 min 19 sec
3	3X3	Adam	99.83	0.9983	0.9984	0.9984	4 hr 53 min 0 sec
4	5X5	Adam	99.16	0.9916	0.9916	0.9916	6 hr 20 min 52 sec

The model's accuracy ranges from 99.16% to 99.83% as can be seen in Table 3. Also, changing the kernel from 3X3 to 5X5 has no noticeable results, however, the elapsed time increases owing to the increased number of parameters. The bar graph in Fig. 9 is an illustration of the performance matrix comparison of the SGDM-based and Adam-based custom CNN model for classifying four classes of diseased rice plants. In the comparison, the SGDM-based models perform better with a 3X3 kernel size, and Adam-based models perform better with a 5X5 kernel size, but with minimal margins in both cases.
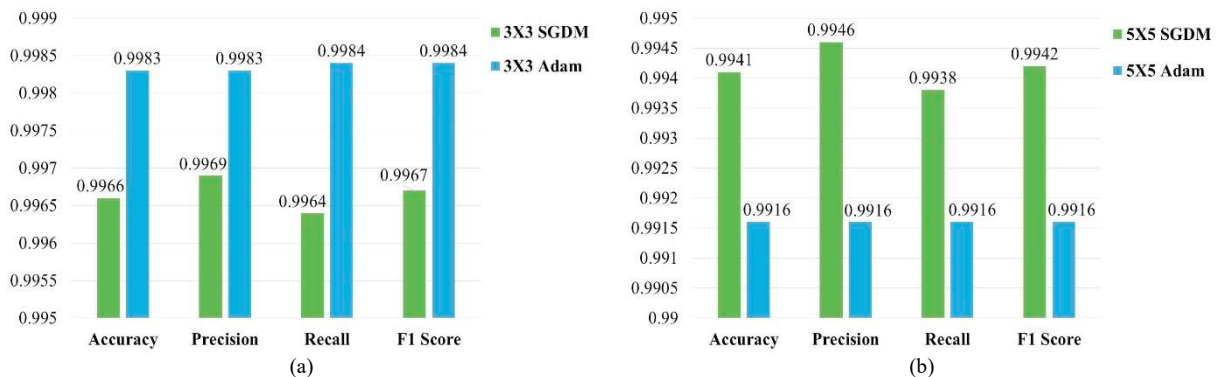

Fig. 9. Performance matrix comparison of the custom CNN model for SGDM-based and Adam-based model for (a) kernel size: 3X3, (b) kernel size: 5X5.

5.2. Evaluation with the inclusion of healthy leaf image dataset to the four classes of rice plant diseases.

The same architectures are trained and the performances are evaluated with the inclusion of the 1400 healthy rice leaf image dataset to the four classes of rice plant diseases keeping all the experimental set-up equal. The comparison of the validation accuracy during the training process of the SGDM-based and Adam-based models in Fig. 10 reaffirms that it is faster to train the Adam-based model as the validation accuracy improves more rapidly as

compared to the SGDM-based model. The confusion matrices of the model's performances on the test set can be observed in Fig. 11 and Fig. 12. From the confusion matrices, it can be observed that most of the misclassification occurs in the healthy plant class. This is because of the fact that the majority of the healthy leaves collected are real-time images, which may contain background noise like dry leaves, weeds, and so on which have close colour proximity with the diseased leaf images.
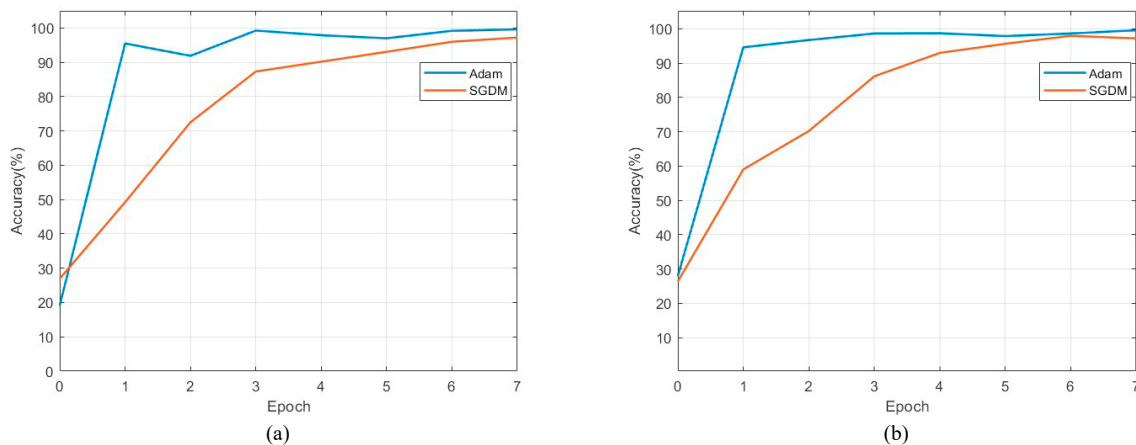

Fig. 10. Validation accuracy comparison of SGDM and Adam optimization with variation in the kernel size during the training process for (a) kernel size: 3X3 (b) kernel size: 5X5.

Fig. 11. Confusion matrices experimental results on the test set with SGDM optimization (a) kernel size: 3X3 (b) kernel size: 5X5.
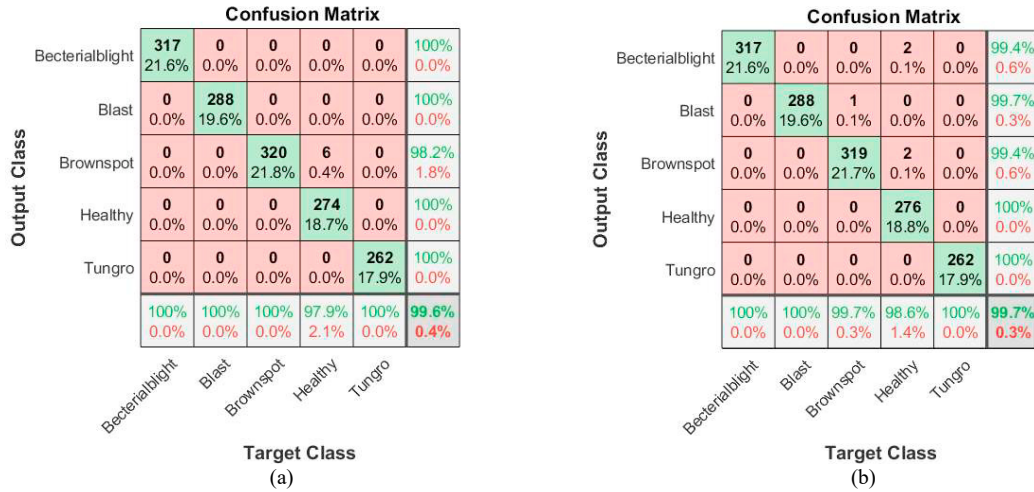
Fig. 12. Confusion matrices experimental results on the test set with SGDM optimization (a) kernel size: 3X3 (b) kernel size: 5X5.

The performance of the SGDM-based model reduces giving a maximum accuracy of 97.61%, while the Adam-based model gives a maximum accuracy of 99.66% on the test set respectively as can be observed from Table 4. The performance matrix comparison in Fig. 13 for the model trained with the inclusion of the healthy rice leaf image dataset to the four classes of rice plant diseases shows that the Adam-based CNN model clearly outperformed the SGDM-based CNN model.

Table 4. Experimental results of the proposed model with the inclusion of the healthy leaf image dataset to the dataset of four classes of rice plant diseases.

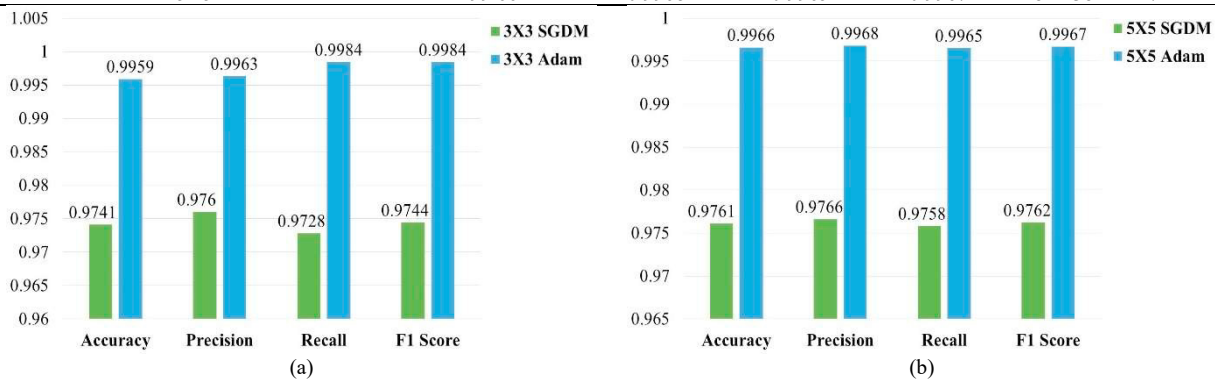| Ex. No.	Kernel size	Optimizer	Accuracy (%)	Precision	Recall	F1 Score	Elapsed Time
1	3X3	SGDM	97.41	0.9760	0.9728	0.9744	6 hr 28 min 23 sec
2	5X5	SGDM	97.61	0.9766	0.9758	0.9762	8 hr 24 min 3 sec
3	3X3	Adam	99.59	0.9963	0.9984	0.9984	6 hr 58 min 27 sec
4	5X5	Adam	99.66	0.9968	0.9965	0.9967	8 hr 38 min 27 sec

Fig. 13. Performance matrix comparison of the custom CNN model for SGDM-based and Adam-based model for (a) kernel size: 3X3, (b) kernel size: 5X5.

Table 5 shows a further evaluation of the performance of the proposed models by comparing it with the existing state-of-the-art models for the detection and classification of rice plant diseases The majority of the models in the comparative research are built using a deep learning methodology. From the comparison, the performance of the proposed models is found to be satisfactory.

Table 5. Performance comparison with other state-of-art models for rice plant disease detection and classification.

Works	Dataset	Methodology	Accuracy
Sethy et al. [14]	5932 images samples (4 classes of diseased rice leaf)	CNN+SVM (ResNet50), Transfer learning (AlexNet)	98.38%, 79.58%
Hossain et al. [36]	Diseased images rice of plants from the International Rice Research Institute (IRRI) and Bangladesh Rice Research Institute (BRRI)	Custom CNN	97.82%
Malathy et al. [38]	3549 paddy pest images	ResNet50	95.01%
Bari et al. [37]	Dataset collected from field and Kaggle	R-CNN	98.84%
Patidar et al. [35]	Dataset from the UCI Machine Learning Repository	Residual Neural Network	95.83%
Sharma et al. [39]	Rice disease dataset containing 1108 images	InceptionResNetV2	98.9%
Upadhyay et al. [40]	12000 images of three rice leaf diseases and 4000 healthy rice leaves taken from Kaggle	CNN, Otsu's global thresholding technique + CNN	99.1%, 99.7%
Proposed model	5932 images samples of diseased rice plants	Custom CNN (4 Classes)	99.83%
Proposed model	5932 images samples of diseased rice plants + 1400 healthy rice leaf images	Custom CNN (5 classes)	99.66%

6. Conclusions

In this paper, a custom CNN-based architecture for the detection and classification of rice plant diseases is presented. Initially, experiments were carried out on the dataset of four classes of diseased rice plants. Statistical analysis of the models' performances on the test set shows no significant difference between the SGDM-based and Adam-based models as the models give accuracies of over 99%. Then, the same architecture is trained and tested with the inclusion of the healthy leaf dataset. The model's performance on the test set shows that the Adam-based CNN model outperforms the SGDM-based CNN model giving maximum accuracies of 99.66% and 97.61% respectively. With the inclusion of this healthy leaf image dataset, the model can distinguish between healthy and diseased leaves hence the detection of disease-free plants is made possible. The proposed architecture has the advantage of notably decreasing the number of parameters compared to the existing architectures. This approach will enable the farmer to diagnose various rice diseases and take preventive measures in time. In the future, the reliability and robustness of the models could be increased by extending the work to a variety of deep learning architectures and different training methods.

References

[1] Bandumula, N. (2018). Rice Production in Asia: Key to Global Food Security. *Proc. Natl. Acad. Sci., India, Sect. B Biol. Sci.* 88, 1323–1328 (2018). https://doi.org/10.1007/s40011-017-0867-7

[2] Importance of Rice (2020). http://www.knowledgebank.irri.org/ericeproduction/ Importance of Rice.htm. Accessed 20 September 2021

[3] Muthayya, S., Sugimoto, J. D., Montgomery, S., & Maberly, G. F. (2014). An overview of global rice production, supply, trade, and consumption. *Annals of the new york Academy of Sciences*, *1324*(1), 7-14.

[4] Gianessi, L. P. (2014). Importance of pesticides for growing rice in South and South East Asia. *International pesticide benefit case study*, *108*, 30-33.

[5] Sethy, Prabira Kumar (2020), "Rice Leaf Disease Image Samples", Mendeley Data, V1, doi: 10.17632/fwcj7stb8r.1 Accessed 30 September 2021

[6] Subhalakshmi, T., & Devi, I. S. (2017). Blast of rice in Manipur and its biocontrol by Pseudomonas fluorescens and Trichoderma sp. *Int J Curr Microbiol App Sci*, *6*(6), 1619-1634.

[7] Pugoy RA, Mariano VY. Automated rice leaf disease detection using color image analysis. InThird international conference on digital image processing (ICDIP 2011) 2011 Jul 8 (Vol. 8009, pp. 93-99). SPIE.OECD.

[8] Orillo, J. W., Cruz, J. D., Agapito, L., Satimbre, P. J., & Valenzuela, I. (2014, November). Identification of diseases in rice plant (oryza sativa) using back propagation Artificial Neural Network. In *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)* (pp. 1-6). IEEE.

[9] Pavithra, S., Priyadharshini, A., Praveena, V., & Monika, T. (2015). Paddy leaf disease detection using SVM classifier. *International Journal of communication and computer Technologies*, *3*(1), 16-20.

[10] Prajapati, H. B., Shah, J. P., & Dabhi, V. K. (2017). Detection and Classification of Rice Plant Diseases.

Intelligent Decision Technologies, 11, 357-373.

[11] Islam, T., Sah, M., Baral, S., & Choudhury, R. R. (2018). A faster technique on rice disease detection using image processing of affected area in agro-field. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 62-66). IEEE.

[12] Chawal, Bikash & Panday, Sanjeev. (2019). Rice Plant Disease Detection using Twin Support Vector Machine (TSVM). Journal of Science and Engineering. 7. 61-69. 10.3126/jsce.v7i0.26794.

[13] Ahmed, K., Shahidi, T. R., Alam, S. M. I., & Momen, S. (2019, December). Rice leaf disease detection using machine learning techniques. In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-5). IEEE.

[14] Sethy, P. K., Barpanda, N. K., Rath, A. K., & Behera, S. K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, *175*, 105527.

[15] Chen, L., & Yuan, Y. (2018). Agricultural disease image dataset for disease identification based on machine learning. *International Conference on Big Scientific Data Management*. Springer, Cham, pp. 263–274.

[16] Singh, V., Asari, V. K., & Rajasekaran, R. (2022). A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease. *Diagnostics*, *12*(1), 116.

[17] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

[18] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[19] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1725-1732).

[20] Zhang, L. N., & Yang, B. (2014). Research on recognition of maize disease based on mobile internet and support vector machine technique. In *Advanced Materials Research* (Vol. 905, pp. 659-662). Trans Tech Publications Ltd.

[21] Gaikwad, V.P., & Musande, V.B. (2017). Wheat disease detection using image processing. *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 110-112.

[22] Oppenheim, D., & Shani, G. (2017). Potato disease classification using convolution neural networks. *Advances in Animal Biosciences*, *8*(2), 244-249.

[23] Jenifa, A., Ramalakshmi, R., & Ramachandran, V. (2019). Cotton leaf disease classification using deep convolution neural network for sustainable cotton production. In *2019 IEEE international conference on clean energy and energy efficient electronics circuit for sustainable development (INCCES)* (pp. 1-3). IEEE.

[24] Liu, B., Zhang, Y., He, D., & Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, *10*(1), 11.

[25] Arivazhagan, S., & Ligi, S. V. (2018). Mango leaf diseases identification using convolutional neural network. *International Journal of Pure and Applied Mathematics*, *120*(6), 11067-11079.

[26] Zhang, S., Zhang, S., Zhang, C., Wang, X., & Shi, Y. (2019). Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Computers and Electronics in Agriculture*, *162*, 422-430.

[27] Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., & Menaka, R. (2020). Attention embedded residual CNN for disease detection in tomato leaves. *Applied Soft Computing*, *86*, 105933.

[28] Ding, J., Li, A., Hu, Z., & Wang, L. (2017). Accurate pulmonary nodule detection in computed tomography images using deep convolutional neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 559-567). Springer, Cham.

[29] Rathod, J., Waghmode, V., Sodha, A., & Bhavathankar, P. (2018). Diagnosis of skin diseases using Convolutional Neural Networks. In *2018 second international conference on electronics, communication and aerospace technology (ICECA)* (pp. 1048-1051). IEEE.

[30] Rastogi, P., Khanna, K., & Singh, V. (2022). LeuFeatx: Deep learning-based feature extractor for the diagnosis of acute leukemia from microscopic images of peripheral blood smear. *Computers in Biology and Medicine*, *142*, 105236.

[31] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456). PMLR.

[32] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)* (pp. 1-6). Ieee.

[33] Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

[34] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[35] Patidar, S., Pandey, A., Shirish, B. A., & Sriram, A. (2020). Rice plant disease detection and classification using deep residual learning. In *International Conference on Machine Learning, Image Processing, Network Security and Data Sciences* (pp. 278-293). Springer, Singapore.

[36] Hossain, S. M., Tanjil, M., Morhsed, M., Ali, M. A. B., Islam, M. Z., Islam, M., ... & Islam, S. M. (2020, November). Rice leaf diseases recognition using convolutional neural networks. In *International Conference on Advanced Data Mining and Applications* (pp. 299-314). Springer, Cham.

[37] Bari, B. S., Islam, M. N., Rashid, M., Hasan, M. J., Razman, M. A. M., Musa, R. M., ... & Majeed, A. P. A. (2021). A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ Computer Science*, *7*, e432.

[38] Malathi, V., & Gopinath, M. P. (2021). Classification of pest detection in paddy crop based on transfer learning approach. *Acta Agriculturae Scandinavica, Section B—Soil & Plant Science*, *71*(7), 552-559.

[39] Sharma, M., Kumar, C. J., & Deka, A. (2022). Early diagnosis of rice plant disease using machine learning techniques. *Archives of Phytopathology and Plant Protection*, *55*(3), 259-283.

[40] Upadhyay, S. K., & Kumar, A. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. *International Journal of Information Technology*, *14*(1), 185-199.