

# A classification model based on depthwise separable convolutional neural network to identify rice plant diseases

Md. Sazzadul Islam Prottasha, Sayed Mohsin Salim Reza

Department of Information and Communication Technology, Bangladesh University of Professionals, Dhaka, Bangladesh

## Article Info

### Article history:

Received Jan 17, 2021

Revised Dec 19, 2021

Accepted Jan 25, 2022

### Keywords:

Agriculture

Convolutional neural network

Deep learning

Image processing

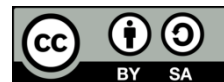
Plant disease

Rice plant diseases

## ABSTRACT

Every year a number of rice diseases cause major damage to crop around the world. Early and accurate prediction of various rice plant diseases has been a major challenge for farmers and researchers. Recent developments in the convolutional neural networks (CNNs) have made image processing techniques more convenient and precise. Motivated from that in this research, a depthwise separable convolutional neural network based classification model has been proposed for identifying 12 types of rice plant diseases. Also, 8 different state-of-the-art convolution neural network model has been fine-tuned specifically for identifying the rice plant diseases and their performance has been evaluated. The proposed model performs considerably well in contrast to existing state-of-the-art CNN architectures. The experimental analysis indicates that the proposed model can correctly diagnose rice plant diseases with a validation and testing accuracy of 96.5% and 95.3% respectively while having a substantially smaller model size.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Md. Sazzadul Islam Prottasha

Department of Information and Communication Technology, Bangladesh University of Professionals

Mirpur Cantonment, Dhaka-1216, Bangladesh

Email: 19541026@bup.edu.bd

## 1. INTRODUCTION

Agricultural science has an enormous effect on the food production system around the world, hence this field is emerging day by day. Technologies have brought a new dimension to this field. Researchers are implementing different methodologies and invented different types of seeds, treatments and weeds to improve the overall crop production. Developments of recent deep learning based image processing methods have improved the disease classification accuracy significantly. Inspired by that, our research is primarily focused on the categorization of various rice plant diseases using a deep learning approach. Contributing to this field has developed a profound interest in us.

There are more than 40 different types of rice plant diseases that can be fatal to the rice plants as described by Ou [1]. Diseases like rice blast, smut and leaf blight can cause severe damage to rice production. There exists some other diseases that can be lethal unless necessary measurements are taken early. Researchers have come up with numerous methodologies and models for the detection of rice plant diseases over the years. Different kinds of segmentation and feature extraction methods have been implemented. A multistage convolutional neural network architecture has been presented by Lu *et al.* [2] that can identify 10 different rice plant diseases. A total of 500 images has been considered including healthy and diseased images and trained using the convolutional neural network (CNN) model. The result reported in the paper shows an accuracy of 95.38% while diagnosing the rice plant diseases. The work was conducted on 10 types of rice plant diseases, however there are only 500 training images meaning only 50 images per disease class. Based on the minimal quantity of training pictures, it seems doubtful that this model will be viable in

the real life scenarios. The work by Liu *et al.* [3] proposed a machine learning based along with a CNN based model for identifying rice false smut images. The proposed CNN architecture inspired from AlexNet and VGGNet-16 architecture comprises a total of 10 layers which includes 7 convolution layers and 3 fully connected layers. Experimental result suggests that, the proposed method performed better than AlexNet in diagnosing the rice false smut disease. Jagan *et al.* [4] developed a two-stage technique for detecting rice plant diseases, in which they detected the disease-affected region using Haar-like characteristics and the AdaBoost classifier in the first stage. After identifying the disease-affected area, they employed scale invariant feature transform to extract features. Finally, support vector machine and k-nearest neighbor classifiers were used for the detecting procedure. An optimal deep CNN architecture has been presented at [5] for diagnosing rice plant diseases. A cloud based infrastructure has been developed where Inception-ResNet V2 model is used for the feature extraction process while weighted extreme learning machine performs the disease classification. The work by Ramesh and Vydeki [6] presents an optimized neural network model to identify 4 types of paddy plant diseases. Initially, the red, green, blue (RGB) images were converted into hue, saturation, value (HSV) images and analyzing the hue and saturation they extracted binary images from the hue and saturation difference. This work integrated Jaya optimization algorithm with the neural network. Jaya algorithm has been used to update the weights of the proposed neural network in identifying rice diseases. The work by Krishnamoorthy *et al.* [7] used InceptionResNetV2 model for detecting multiple paddy leaf diseases. By using transfer learning method and tuning the hyperparameters the proposed model achieved an accuracy of 95.67%. The method provided by Shrivastava *et al.* [8] uses the AlexNet model using transfer learning to detect 3 different rice plant diseases. For an 80-20 train and test data split, the AlexNet model achieved a classification accuracy of 91.37%. A simple CNN architecture for diagnosing various rice plant diseases from Bangladesh has been proposed at [9]. The work focuses on providing a lightweight CNN model by hyper-parameter tuning which can provide admissible accuracy. The statistical analysis shows that using Adam optimizer their proposed model achieved a validation accuracy of 95.4%. Similar CNN model has been proposed for mango [10], wheat [11], banana [12], apple [13], and peach [14] classification.

Recent developments in various computer vision models faster R-CNN, YOLOv3, mask R-CNN, and RetinaNet have significantly improved the accuracy of object detection modules. These models not only detect the disease but also identifies the exact location of the disease occurrence. Sethy *et al.* [15] presented a faster R-CNN object detection method for rice false smut disease. After collecting 50 images of false smut they used image augmentation methods to increase the images in the dataset. After extracting the features using the ResNet-50 model, the feature vectors are fed through the fully connected layer to predict the smuts using bounding boxes. In case of numerous false smut are present, the model occasionally fails to detect all smuts using bounding boxes. Bari *et al.* [16] proposed a faster R-CNN based real-time rice leaf disease diagnosis system. Experimental analysis shows that, the proposed method can localize the disease-affected portions with higher precision. However, the model requires multiple recurrences to extract all objects inside a single image which makes the model slower. Considering these computer vision models, we developed a CNN model based on depthwise separable convolutions to diagnose 12 types of rice plant diseases.

## 2. METHOD

In this section, the details of the dataset collection process and proposed models are discussed in appropriate subheadings. Also, the model hyperparameters are discussed here. Figure 1 illustrates the details of our method.

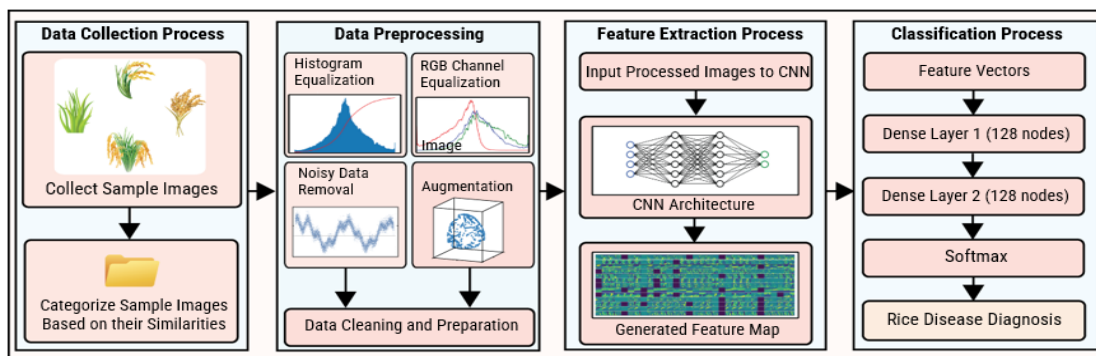


Figure 1. Method of rice plant disease detection process

Our experimental study starts with data acquisition followed by data pre-processing. Then the processed images are fed through the CNN model. The CNN model generates feature maps and eventually the classification is performed by the fully connected dense layers.

### 2.1. Dataset collection

We have taken some extensive measurements to collect 1,677 rice plant raw sample images from different regions of Bangladesh, consisting 12 types of fungal and pest diseases along with healthy plants. The data were collected under different weather conditions over a total 4-month span. In addition to our accumulated sample images, we took some more sample images from the dataset stated by Rahman *et al.* [17]. The images were taken on different heterogeneous backgrounds under different lighting conditions both in sunny and overcast environments. The collected types of diseases we contemplated are bacterial blight, brown plant hopper, brown spot, false smut, hispa, leaf blast, leaf scald, leaf smut, neck blast, sheath blight rot, stackburn and stemborer as shown in Figure 2(a) to (l). Along with these 12 fungal and pest diseases, we have also taken images of healthy leaves and panicles to differentiate them from the disease-affected plants. The dataset can be found here at [18]. Figure 2 shows the collected sample images of our dataset.

The images were captured during different stages of the disease infection considering different symptoms of the diseases; hence we get a fully representative dataset considering all aspects of the disease. There are 10 classes in our dataset that contain multiple symptom variations. Table 1 shows the number of sample images collected for different symptoms of diseases and pests.

Bacterial leaf blight caused by a bacteria called *Xanthomonas oryzae pv. oryzae* which attacks the leaves of the rice plant causing it to dry out [19]. Brown spot is a fungal disease caused by *Cochliobolus miyabeanus*. This disease mostly occurs at the protective sheath covering the leaf of the rice plants, leaves and *spikelets* [20]. False smut is a rice grain disease caused by a plant pathogen named *Ustilaginoidea virens* which transforms the individual rice grains into yellow fruiting bodies [21]. In some cases, the smuts turn black in color. We have considered both types of false smut images in our dataset. Leaf blast, neck blast, leaf smut, leaf scald, sheath blight and stackburn are similar types of disease considering the fact that they mostly occur at the rice leaf. Each of the diseases causes small spots or lesions in the rice plants. In different cases, there can be single or multiple spots with similar diameter in a single leaf. Neck blast also attacks the neck calm of the rice plant making it vulnerable. Leaf blast and neck blast are more deadly than the other diseases and in severe cases, the yield loss can be as high as 100% as indicated in [22].

In our dataset, we have also considered certain pests and bugs. The Brown planthopper is a disease caused by a pest named *Nilaparvata lugens*. The pests mostly reside at the root of the plants then spreading through the entire rice plant causing the plant to dry out and turn brown [23]. Hispa is another disease that is caused by a pest named *Dicladispa armigera* where the armigera insect scrapes the upper surface of the leaves leaving only the lower epidermis. Similar to hispa, stemborer is a pest that also attacks the leaves of rice plants. The stem borer initially resides at the base of the rice plants and later the stem borer larva drill through the upper nodes and feeds the tillers which cause the tillers to dry out [24]. Different environmental factors along with excessive use of nitrogen fertilizers are the main reason for these disease occurrences.

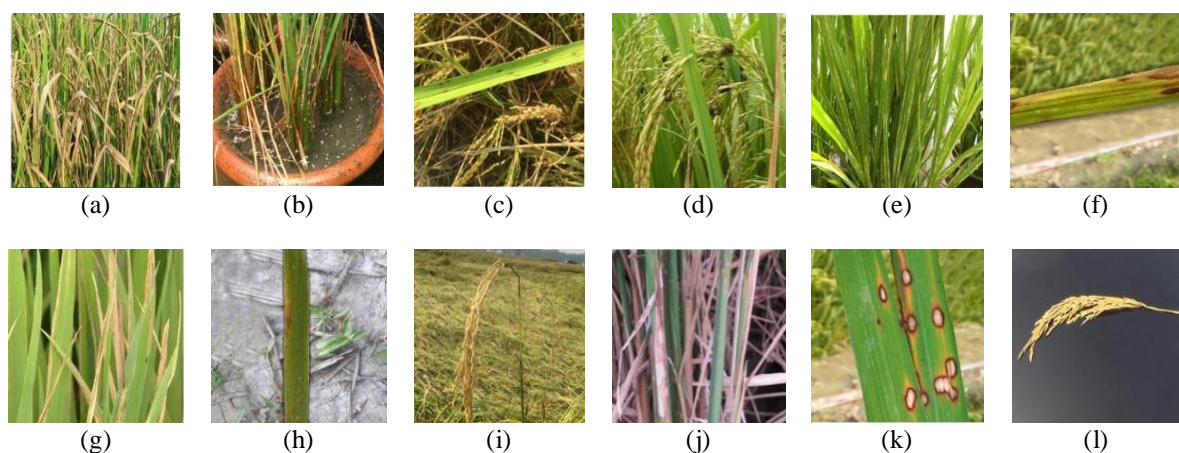


Figure 2. Sample images of our collected rice plant disease dataset (a) bacterial blight, (b) brown planthopper, (c) brown spot, (d) false smut, (e) hispa, (f) leaf blast, (g) leaf scald, (h) leaf smut, (i) neck blast, (j) sheath blight, (k) stackburn, and (l) stemborer

Table 1. Total collected sample images with different symptom variations

Name	Sample image variation	No. of collected images	Total images
Bacterial blight	Early stage yellowish brown	46	126
	Later stage full brown	80	
Brown plant hopper	Early stage of attack	50	75
	Later stage of attack	25	
Brown spot	Small size spots	102	102
False smut	Black smut	78	112
	Brown smut	34	
Hispa	Visible pests on the leaves	56	86
	Visible spots on the leaves without pests	30	
Leaf blast	Deep brown spot	110	136
	Yellowish grey spot	26	
Neck blast	Blast spot on neck	225	225
Leaf scald	Dark brown lesion	23	73
	Greyish white lesion	50	
Leaf smut	Spots on the whole leaf	103	103
Sheath blight	Black stem	112	228
	White stem	116	
Stemborer	Grain symptom	171	187
	Stem symptom	16	
Stackburn	Symptom on whole leaf	49	71
	Symptom on a part	22	
Healthy	Healthy leaf	90	153
	Healthy grain	63	

## 2.2. Image preprocessing

After collecting the rice plant sample images, we processed the sample images before inputting them to the convolutional neural network model. At first, we removed out-of-focus and blurry images from our collected dataset. Once the faulty images were discarded, we move on to the image augmentation process to increase the size of our dataset. Since we have small number of images for each of the class indicated by Table 1, hence for better interaction and considering all possible dependencies at the feature level, we perform image augmentation. The first augmentation method we carried out is the contrast stretching method. Due to the gloomy weather and mobile camera lens, few sample pictures had very low contrast, hence making it difficult to differentiate between the foreground subject and the background. This sort of low contrasting pictures will prompt false positive and false negative rates in the network resulting in low accuracy. Therefore, we utilised the contrast limited adaptive histogram equalization method described by Pizer *et al.* [25] to enhance the low contrast sample images. We performed local histogram enhancement using the contrast limited adaptive histogram equalization (CLAHE) method. CLAHE restricts the amplification level by clipping the histogram at a predefined value before calculating the cumulative distribution function. The clipping value depends upon the normalization of the histogram hence on the size of the local area. The result of the histogram equalization process is illustrated in Figure 3, where Figure 3(a) shows input image with distribution range in Figures 3(b) and (c). Figure 3(d) shows histogram equalized image with distribution range in Figure 3(e) and (f).

It is evident that the intensity values of histogram equalized images are spread through the full dynamic range hence the details are considerably more discernible. After performing the histogram equalization method, we performed various image augmentation methods on our dataset. We performed flipping, skewing, random rotation, random zoom, shear transformation, noise and distortion as distinct image augmentation processes. Using these procedures, 10 augmented images have been created for each example image and thus our dataset increased to a total of 16,770 training images. We used 80-20 data split on our dataset by which we have 13,415 and 3,355 images for training and validating the model respectively.

## 2.3. State-of-the-art CNN architectures

Eight different state-of-the-art CNN architectures have been employed on our rice disease dataset based on different criteria. We used VGG net which has 3×3 filters throughout the CNN architecture and they are very deep CNN architectures with a large number of parameters [26]. The Inception-v3 architecture by Szegedy *et al.* [27] has an inception module. Despite being a very deep architecture, the inception module greatly reduces the amount of the parameters. By the use of asymmetric convolution, factorization is performed which reduces the complexity of this model. ResNet architecture has residual networks with skip connections and overcomes the problem of overfitting in ImageNet dataset [28]. The Xception [29] introduced depthwise separable convolutions along with residual connections. Mobilenet v2 architecture also uses depthwise separable convolutions having inverted shortcut network with linear bottleneck [30]. NasNet Mobile architecture is similar to Mobilenet v2 architecture and it is used primarily for mobile devices. We

intended to see how our dataset performs on these models and later compared the results of these state-of-the-art CNN models with our proposed CNN model.

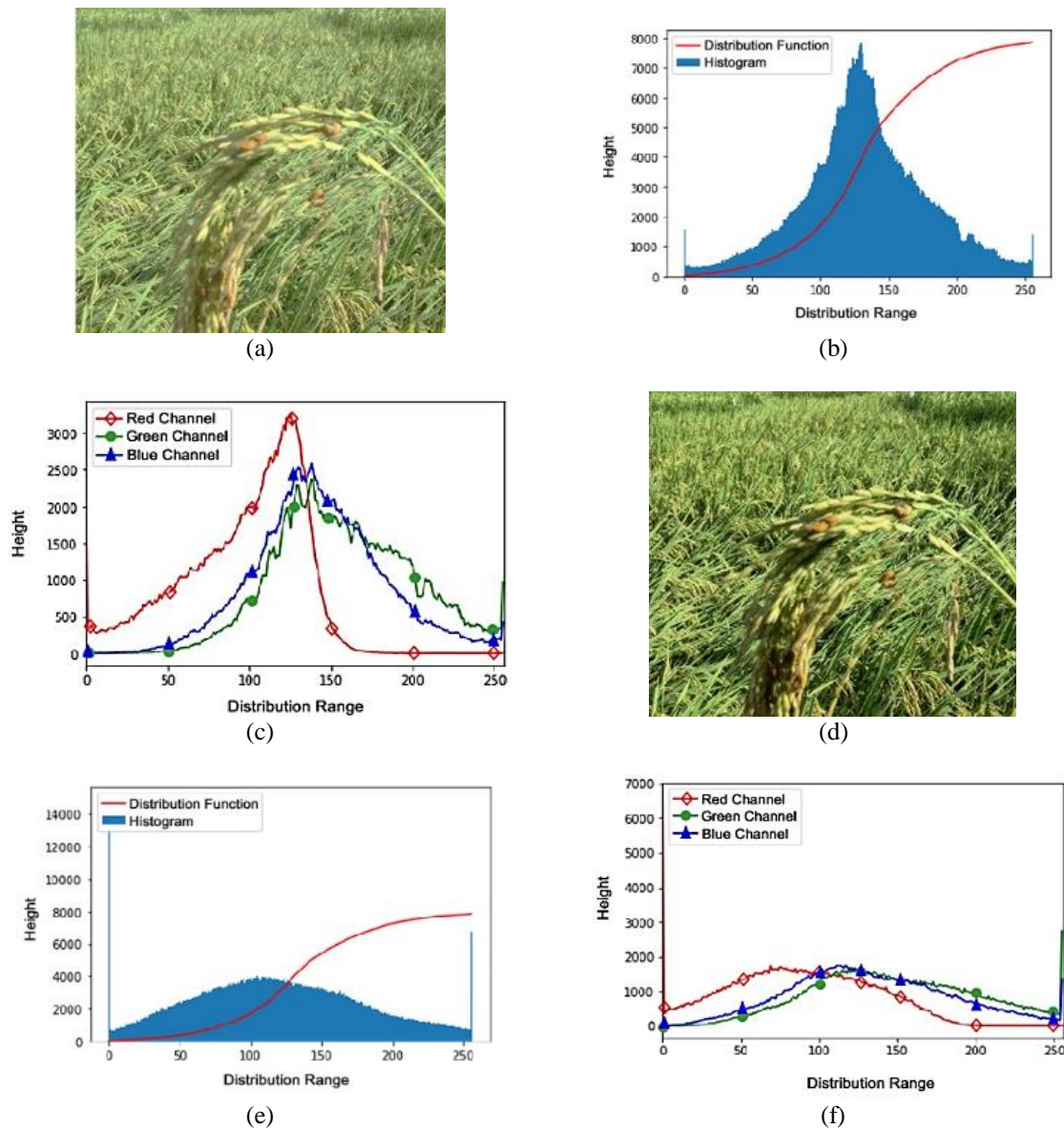


Figure 3. Contrast limited adaptive histogram equalization method (a) low contrast input image, (b) histogram of input image, (c) RGB channel of input image, (d) high contrast output image, (e) histogram of output image, and (f) RGB channel of output image

#### 2.4. Proposed depthwise CNN model

We proposed a deep CNN architecture based on depthwise separable convolution. In the 1st layer of our proposed CNN model, we use a  $3 \times 3$  2D convolution followed by batch normalization and  $2 \times 2$  max-pooling. In the 2nd layer, we used a depthwise separable convolution layer which consists of a  $3 \times 3$  depthwise convolution followed by a  $1 \times 1$  point-wise convolution. Unlike the normal convolution, depthwise separable convolution deals with not only the spatial dimension but also the depth dimension.

For normal convolution considering a kernel  $K$  having spatial height  $K_h$  and spatial width of  $K_w$  and image tensor with input and output channels of  $I_i$  and  $I_o$  respectively. The convolution layer is represented by,  $C \in \mathbb{R}^{I_i \times I_o \times K_h \times K_w}$ . After applying the filter to a 3D tensor  $x$ , with size  $K_h \times K_w \times I_i$ , we obtain a response vector  $y \in \mathbb{R}^{I_o}$ .

$$y = C * x \quad (1)$$

where  $y_o = \sum_{i=1}^I C_{o,i} * x_i$ ,  $o \in I_o$ ,  $i \in I_i$ . Here  $*$  is the convolution operator.  $C_{o,i} = C[o, i, :, :]$  is a tensor patch along the  $i$ -th input channel and  $o$ -th output channel.  $x_i = x[i, :, :]$  is a tensor patch along the  $i$ -th input channel of 3D tensor  $x$ .

For depthwise separable convolution, the convolutions are performed at each depth by which the output is split for each dimension. Then pointwise convolution is performed for projecting the outputs into a single channel space. Pointwise convolution is done by performing  $1 \times 1$  cross-channel convolutions. Since the depthwise convolution is performed at each depth at a time hence the kernel tensor is  $C_d \in \mathbb{R}^{I_i \times 1 \times K_h \times K_w}$  and for pointwise convolution the kernel tensor is  $C_p \in \mathbb{R}^{I_i \times I_o \times 1 \times 1}$ . After applying this to a 3D tensor  $x$ , we obtain a response vector  $y'$  as,

$$y' = (C_p \circ C_d) * x \quad (2)$$

where  $y_o' = \sum_{i=1}^I C_{p_{o,i}}(C_{d_{o,i}} * x_i)$ ,  $\circ$  is a element wise operation.  $C_{p_{(o,i)}} = C_p[o, i, :, :]$  is a tensor patch along the  $i$ -th input channel and  $o$ -th output channel.  $C_{d_{(i)}} = C_p[i, :, :, :]$  is a tensor patch along the  $i$ -th input channel of 3D tensor  $x$ .

As we can see, rather than performing the convolution in spatial dimension in RGB images, depthwise separable convolutions are performed for each color channel separately. Due to single-channel calculation, the overall complexity of the depthwise separable convolution is less than the normal convolution on spatial dimension which accelerates the overall classification process. Because of the small parameter size and faster execution, we specifically used depthwise separable convolution in our proposed CNN model.

Our proposed CNN network starts with a normal  $3 \times 3$  convolution followed by 2 depthwise separable convolutions and then again a normal  $3 \times 3$  convolution. Finally, we used 2 fully connected layers having 128 neurons each and a 0.3 dropout layer in between the dense layer to reduce overfitting.  $2 \times 2$  max pooling and batch normalization were performed after every convolution. The rectified linear unit (ReLU) activation function has been used for every layer except for the final dense layer which uses the softmax activation function for the classification task. For updating the weights, several loss functions such as Adam, RmsProp, Stochastic gradient descent, Hingeloss were utilized, and their performance was evaluated. Our proposed depthwise CNN architecture is demonstrated in Figure 4.

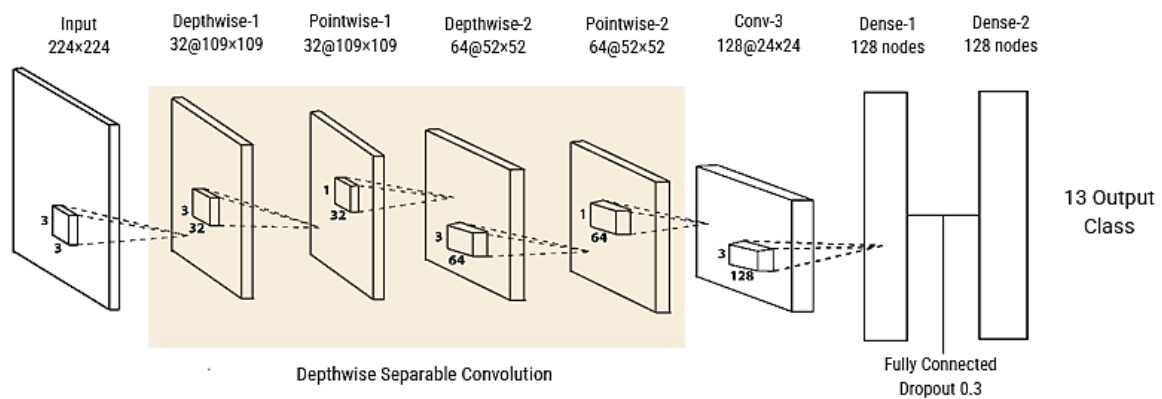


Figure 4. Proposed depthwise separable CNN model

Our proposed CNN model's hyperparameters are listed below.

- Fold: 10-fold cross validation
- MiniBatch: 64
- Epochs: 100
- Activation Function: ReLU
- Momentum: 0.9
- Dropout: 0.3
- Total Data: 16770
- Data Split: 80-20
- Training Data: 13415
- Validation Data: 3355
- Testing Data: 1600
- Image Size: 224x224

All the images were resized to  $224 \times 224$  pixels before inputting them in the CNN model. The first layer of our CNN model takes an input  $(224, 224, 3)$  and produces an output tensor of  $(222, 222, 32)$ . After every convolution, batch normalization and  $2 \times 2$  max-pooling are performed. Table 2 shows the kernel dimension of each layer along with the generated feature maps of our proposed CNN model.

The output tensor  $(12, 12, 28)$  is fed through the flatten layer which maps the input matrix into a 1-D vector of 18432 values. This vector is passed through two fully connected layers each of them consisting of 128 neurons and finally, we get 13 probabilistic values and thus the classification is performed. Figure 5 demonstrates the input images along with their feature maps generated by the last convolution layer.

Table 2. Proposed model generated feature vector output of each layer

Layer name	Kernel dimension	Stride	Input feature vector	Output feature vector
Conv_2d	$3 \times 3$	-	$224 \times 224 \times 3$	$224 \times 224 \times 32$
Maxpooling1_2d	$2 \times 2$	2	$224 \times 224 \times 32$	$112 \times 112 \times 32$
SeparableConv1_2d	$3 \times 3$	-	$112 \times 112 \times 32$	$109 \times 109 \times 32$
Maxpooling2_2d	$2 \times 2$	2	$109 \times 109 \times 32$	$54 \times 54 \times 32$
SeparableConv2_2d	$3 \times 3$	-	$54 \times 54 \times 32$	$52 \times 52 \times 64$
Maxpooling3_2d	$2 \times 2$	2	$52 \times 52 \times 64$	$26 \times 26 \times 64$
Conv3_2d	$3 \times 3$	-	$26 \times 26 \times 64$	$24 \times 24 \times 128$
Maxpooling4_2d	$2 \times 2$	2	$24 \times 24 \times 128$	$12 \times 12 \times 128$
Flatten	-	-	$12 \times 12 \times 128$	18432

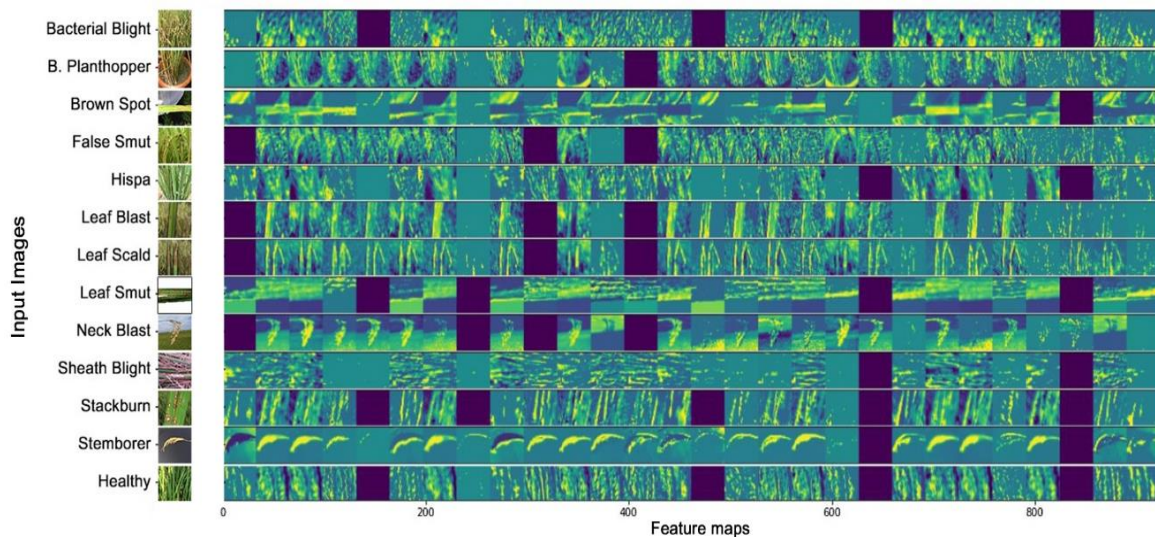


Figure 5. Final convolution layer feature map output for each class

### 3. RESULTS AND DISCUSSION

In this section, the detailed analysis of various state-of-the-art CNN models performance along with our proposed model are discussed. Also, the findings are reported extensively. The complete result analysis has been discussed in appropriate subheadings.

#### 3.1. State-of-the-art CNN models performance

For experimental work, Keras framework with TensorFlow back-end consisting Tesla P-100 GPU has been used to train the models. For the training process, we used two different methodology transfer learning and fine-tuning. In the transfer learning method, every model had pre-trained ImageNet weights associated with them. Randomly initialized weights are only used to train the final dense layer. The fine-tuning technique however unfreezes the final dense layer of the CNN and freezes all other layer weights. Then the dense layers are trained using our rice plant dataset and different weights are initialized to the nodes. The networks later adjusted the weights using optimizers. 10-fold cross-validation accuracy has been considered as the model performance metric. We used different train, validation and testing set for multiple instances and recorded the mean results. Table 3 shows the mean training, validation and testing result after 100 iterations of the rice plant disease dataset on different state-of-the-art CNN architectures.

The result reported at Table 3 indicates that, fine-tuning produced a better result than the transfer learning method for all of the CNN models. The performance curve of 8 state of the art CNN architectures are shown in Figure 6 with transfer learning validation accuracy in Figure 6(a) and fine tuning validation accuracy in Figure 6(b). Among the state-of-the-art CNN architectures, MobileNet v2 architecture performed best in terms of accuracy. VGG-16 architecture provided high mean validation and testing accuracy. The other deep CNN architectures (e.g. Inception, ResNet-50) with big parameters size mostly failed to identify the test and validation rice plant images correctly. In the transfer learning method, we noticed that most of the CNN architectures showed overfitting problem thus resulting in low testing accuracy. Even during the fine-tuning method, Inception v3 and ResNet-50 model showed overfitting characteristics. The architectures performed better in training data but failed in validation and test data.

Table 3. Statistical analysis of different CNN architectures on rice plant disease dataset. Here bold numbers indicate best result

Architecture	Training method	Mean train accuracy	Mean validation accuracy	Mean test accuracy	Total parameters
VGG 16	Transfer learning	94.2±1.8%	80.3±1.5%	78.2±1.0%	134 million
	Fine tuning	98.6±0.9%	97.1±0.4%	95.7±0.5%	
VGG 19	Transfer learning	89.5±1.5%	81.3±3.5%	79.1±2.0%	143 million
	Fine tuning	97.3±0.7%	93.6±0.3%	86.2±1.5%	
Inception v3	Transfer learning	84.8±3.0%	65.2±2.5%	61.7±3.0%	24 million
	Fine tuning	98.6±0.4%	84.1±1.5%	83.2±2.0%	
Xception	Transfer learning	90.2±2.5%	84.2±1.5%	79.4±2.0%	20 million
	Fine tuning	99.3±0.6%	95.1±0.8%	93.1±0.6%	
MobileNet v2	Transfer learning	88.5±3.0%	83.2±2.0%	78.6±1.4%	3.5 million
	Fine tuning	98.9±0.5%	97.3±0.4%	95.7±0.3%	
ResNet-50	Transfer learning	39.2±6.0%	24.1±4.5%	20.2±5.0%	25 million
	Fine tuning	67.7±1.5%	58.5±3.0%	52.1±2.5%	
DenseNet-121	Transfer learning	94.2±2.8%	84.1±0.9%	78.0±1.5%	8 million
	Fine tuning	98.1±0.9%	95.7±0.5%	92.3±1.0%	
NasNet Mobile	Transfer learning	94.1±1.0%	85.0±1.0%	78.4±1.8%	4.3 million
	Fine tuning	97.6±0.5%	93.2±1.0%	91.0±1.2%	
Proposed CNN	Train from scratch	98.7±0.3%	96.5±0.3%	95.3±0.5%	2.4 million

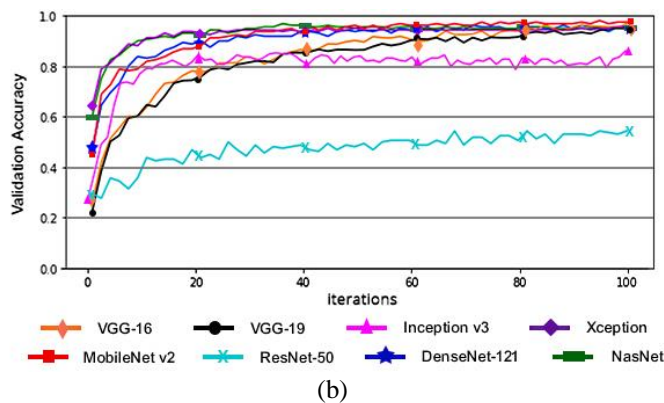
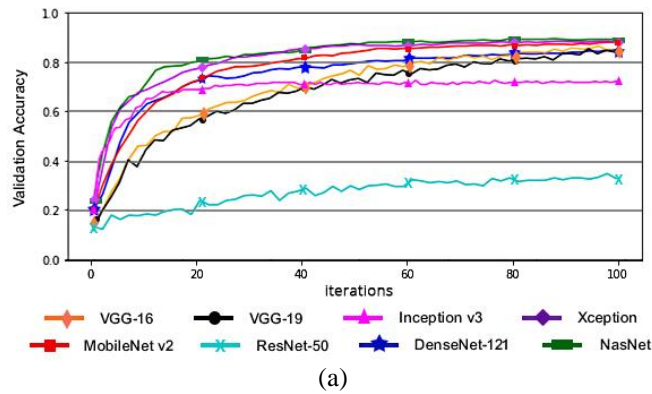


Figure 6. Performance curve of different CNN models on transfer learning and fine tuning method, (a) transfer learning validation accuracy and (b) fine tuning validation accuracy



### 3.2. Proposed CNN model performance

Our proposed model was trained from scratch and the weights were updated accordingly. We used different hyperparameter combination and evaluated the performance thus selected the most suitable parameters for our proposed model. By hyperparameter tuning, our model achieved a mean validation and mean testing accuracy of 96.5% and 95.3% respectively. Considering the small parameter size, our model performed significantly well in diagnosing rice plant diseases. We used different optimizers in our proposed model and achieved the best accuracy using the Adam optimizer. Figure 7 demonstrates the best loss and accuracy curve achieved with our proposed model using Adam optimizer with a 0.001 learning rate.

From the loss and accuracy curve, it is evident that our proposed model slowly converged to the without showing any overfitting characteristics. We employed a variety of optimizers, including Adam, stochastic gradient descent (SGD), Rmsprop, Hingeloss along with variable learning rates on our model. Among the optimizers, adaptive moment estimation (Adam) [31] outperformed all the other optimizers in terms of accuracy. Rmsprop optimizer performed similarly to Adam optimizer. Table 4 shows the performance of using different optimizers with variable learning rates on our rice plant disease dataset.

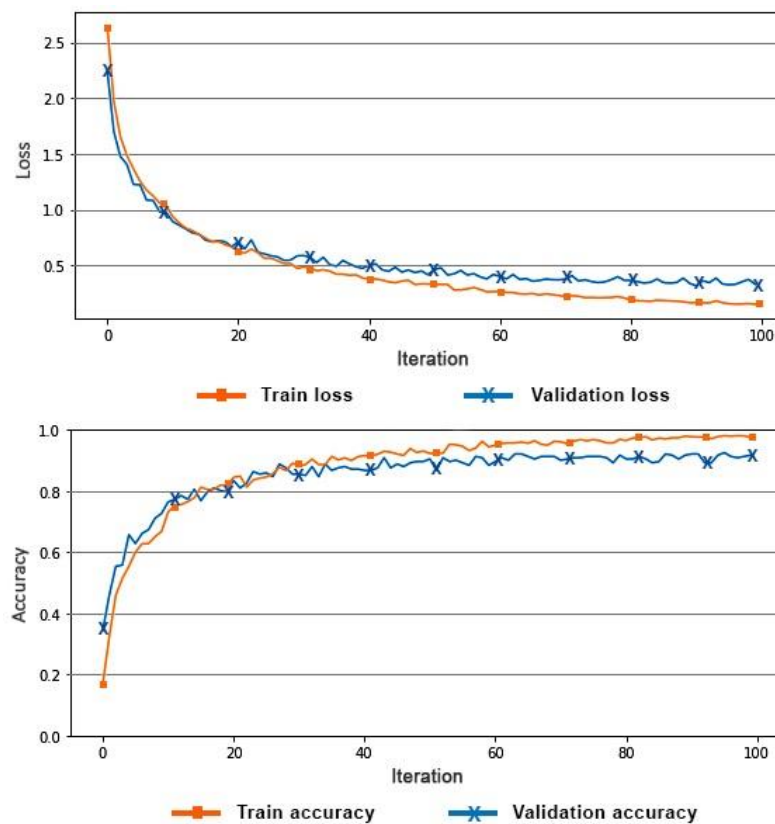


Figure 7. Performance curve of the proposed CNN model using Adam optimizer with 0.001 learning rate

Table 4. Different optimizers performance with variable learning rate on rice plant disease dataset

Optimizer	Learning rate	Mean validation accuracy
Adam	0.01	95.4±0.4%
	0.001	<b>96.5±0.3%</b>
	0.0001	95.1±0.4%
RmsProp	0.01	94.1±0.3%
	0.001	95.01±0.3%
	0.0001	94.4±0.4%
HingeLoss	0.01	78.2±1.8%
	0.001	82.7±1.2%
	0.0001	73.5±1.5%
SGD	0.01	89.2±1.0%
	0.001	88.5±1.5%
	0.0001	82.1±1.5%

From Table 4 it is evident that, stochastic gradient descent (SGD) and the support vector machine SVM optimizer Hingeloss didn't perform as well as Adam or Rmsprop. Adam and Rmsprop performed quite similarly. The accuracy curve of the Adam optimizer with variable learning rates on our dataset is illustrated in Figure 8.

From Figure 8 we can observe that the learning rate of 0.01 converged very quickly to our rice disease dataset. On the other hand, the learning rate of 0.001 converged slowly to our dataset but achieved overall maximum accuracy in the validation dataset. The learning rate of 0.0001 also performed well and provided good accuracy. Hence, the optimal learning rate for our proposed CNN model is 0.001. However, the difference is subtle and learning rate of 0.01 and 0.0001 can be used on our proposed model. The normalized confusion matrix for the testing data of our proposed model is shown in Figure 9.

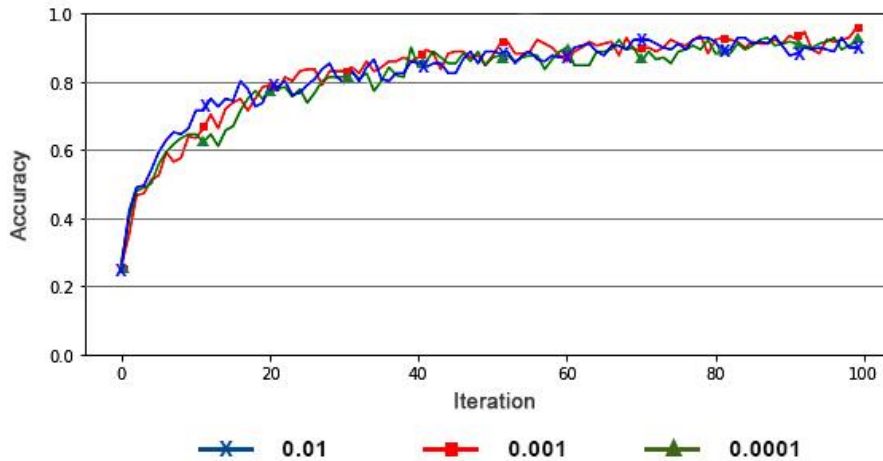


Figure 8. Adam optimizer with variable learning rate

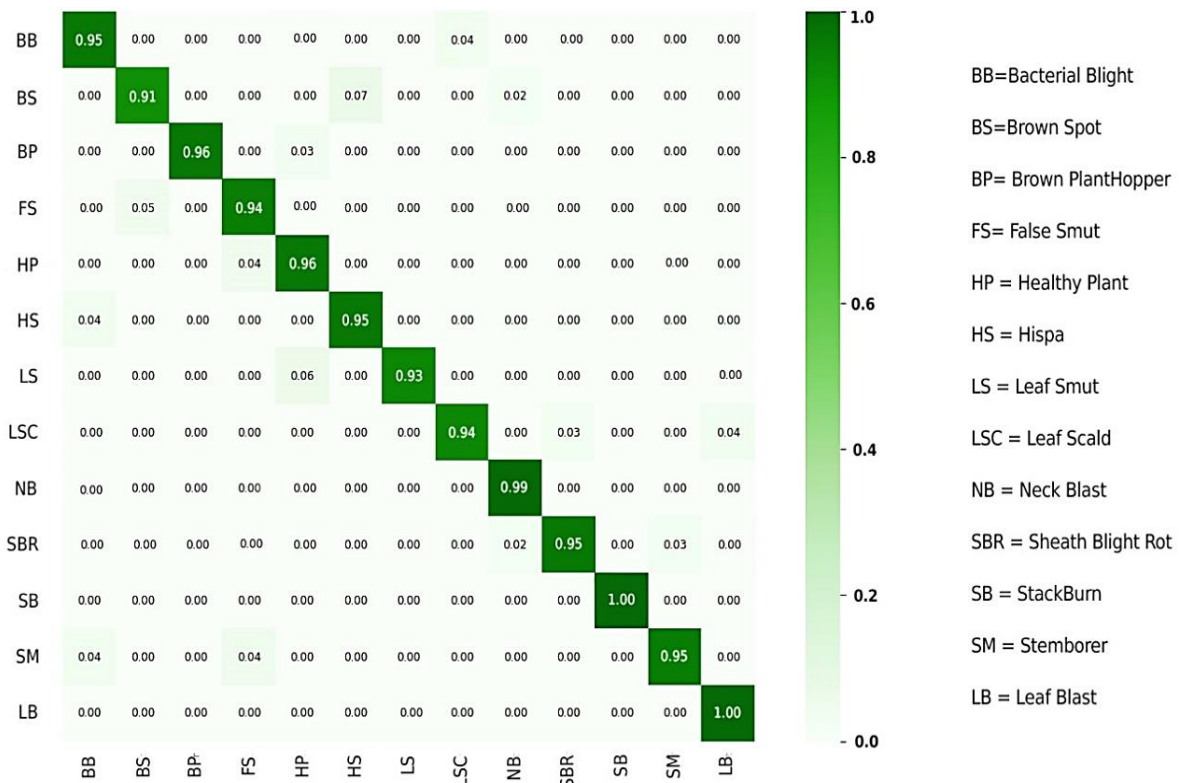


Figure 9. Normalized confusion matrix

Our proposed model achieved a mean testing accuracy of 95.3%. By analyzing the normalized confusion matrix, we observe that, apart from the brown spot disease all of the other disease classes have high detection accuracy which clearly indicates that our proposed model can recognize the disease classes correctly. The comparison of our proposed model with existing related works is represented in Table 5. We observe from Table 5 that the majority of the research has focused on a small number of disease types. And their proposed method comprising lower accuracy than us. Taking a modest parameter size into account, it is evident that our proposed model performed significantly well in detecting rice plant diseases.

Table 5. Comparison of our proposed model with existing works

Authors	Proposed Method	Dataset	Iteration	Performance
Lu <i>et al.</i> [2]	A simple 5-layer CNN architecture	500 sample images of 10 classes collected from fields	10	Simple CNN: 95.38%
Liu <i>et al.</i> [3]	A custom 6-layer CNN architecture	693 sample images of 2 classes collected from fields	10,000	Custom CNN: 99.5% AlexNet-8: 98% VGGNet-11: 96.7%
Jagan <i>et al.</i> [4]	SIFT combined with different classifiers	120 sample images of 3 classes	-	SVM classifier: 91.1% KNN classifier: 93.33%
Sowmyalakshmi <i>et al.</i> [5]	Inception ResNet v2 integrated with OWELM	115 sample images of 3 classes	-	Inception ResNet v2: 94.2%
Krishnamoorthy <i>et al.</i> [7]	Inception ResNet v2 with transfer learning method	3 types of disease classes along with healthy plants	-	Inception ResNet v2: 95.67% Simple CNN: 84.75%
Sethy <i>et al.</i> [15]	Faster R-CNN to localize disease spots	50 sample images of false smut	5	In most cases the model can identify the false smuts
Rahman <i>et al.</i> [17]	A simple CNN architecture	1,426 sample images of 9 classes collected from fields	100	Simple CNN: 94.33% VGG-16: 97.12%
Proposed CNN	A CNN model based on Depthwise separable convolutions	1,677 sample images of 13 classes collected from fields	100	Proposed CNN: 96.5%

### 3.3. Discussion

From our experimental analysis we have obtained some interesting findings regarding our rice disease dataset. We identified some interesting facts about how different model parameter metrics improve the performance of classification model. The findings are reported as follows: i) using transfer learning method with pre-trained ImageNet weights on state-of-the-art CNN architectures cannot provide higher classification accuracy. In all eight state-of-the-art CNN architectures, the fine-tuning method outperforms the transfer learning method in terms of accuracy; ii) the lightweight CNN architectures (MobileNet v2, NasNet Mobile, DenseNet) converge quickly to our dataset and provides a greater degree of precision; iii) the deep CNN architectures with pre-trained ImageNet weights cannot converge quickly to our rice plant disease dataset and mostly shows overfitting characteristics; iv) categorical cross-entropy with Adam optimizer performs better in updating the weights for the dense layer on proposed model. However, different learning rate provides almost similar performance which indicates that learning rate has very limited effect on updating the weights, and v) on our proposed model, a learning rate of 0.001 and momentum of 0.9 performs well and produces better accuracy in classifying rice plant diseases.

## 4. CONCLUSION

In this paper, a depthwise separable convolution based neural network model has been proposed that can effectively identify 12 distinct rice plant diseases along with healthy rice plants. Initially, the raw sample images were collected from different regions of Bangladesh and during the pre-processing method contrast limited adaptive histogram equalization algorithm (CLAHE) has been implemented to enhance the contrast of the images and to reduce the noise. Along with histogram equalization different image augmentation method has been performed by which the sample images of the dataset are increased to 16770 images. Besides our proposed model, 8 different state-of-the-art CNN architectures have been used on the rice disease dataset. Among all CNN architectures in terms of accuracy, MobileNet v2 was found to have the best validation and testing accuracy of 97.3% and 95.7% respectively. The proposed CNN model has been evaluated with different loss functions along with variable learning rates. Adam optimizer with a learning rate of 0.001 provided the best mean validation and mean test accuracy of 96.5% and 95.3% respectively for our proposed CNN model. Based on our model's performance, it is evident that the proposed model was able to properly diagnose a variety of rice plant diseases. Considering a small parameter size of 2.4 million we conclude that, the proposed model is a substantial improvement over the traditional convolutional neural network architectures in rice plant disease detection.




## ACKNOWLEDGEMENTS

We gratefully acknowledge the ICT division of the Government of the People's Republic of Bangladesh for funding this research work.




## REFERENCES

- [1] S. H. Ou, *Rice diseases*, 2nd ed. Commonwealth Mycological Institute, 1985.
- [2] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, Dec. 2017, doi: 10.1016/j.neucom.2017.06.023.
- [3] C. Liu, C. Xu, S. Liu, D. Xu, and X. Yu, "Study on identification of rice false smut based on CNN in natural environment," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Oct. 2017, pp. 1–5, doi: 10.1109/CISP-BMEI.2017.8302016.
- [4] K. Jagan, M. Balasubramanian, and S. Palanivel, "Detection and recognition of diseases from paddy plant leaf images," *International Journal of Computer Applications*, vol. 144, no. 12, pp. 34–41, Jun. 2016, doi: 10.5120/ijca2016910505.
- [5] R. Sowmyalakshmi *et al.*, "An optimal classification model for rice plant disease detection," *Computers, Materials and Continua*, vol. 68, no. 2, pp. 1751–1767, 2021, doi: 10.32604/cmc.2021.016825.
- [6] S. Ramesh and D. Vydeki, "and classification of paddy leaf diseases using optimized deep neural network with jaya algorithm," *Information Processing in Agriculture*, vol. 7, no. 2, pp. 249–260, Jun. 2020, doi: 10.1016/j.inpa.2019.09.002.
- [7] N. Krishnamoorthy, L. V N. Prasad, C. S. P. Kumar, B. Subedi, H. B. Abraha, and S. V E, "Rice leaf diseases prediction using deep neural networks with transfer learning," *Environmental Research*, vol. 198, Jul. 2021, doi: 10.1016/j.envres.2021.111275.
- [8] V. K. Shrivastava, M. K. Pradhan, S. Minz, and M. P. Thakur, "Rice plant disease classification using transfer learning of deep convolutional neural network," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 631–635, Jul. 2019, doi: 10.5194/isprs-archives-XLII-3-W6-631-2019.
- [9] M. S. I. Prottasha, Z. Tasnim, S. M. S. Reza, and D. A. Hossain, "A lightweight CNN architecture to identify various rice plant diseases in Bangladesh," in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, Feb. 2021, pp. 316–320, doi: 10.1109/ICICT4SD50815.2021.9396927.
- [10] A. S. A. Mettleq, I. M. Dheir, A. A. Elsharif, and S. S. Abu-Naser, "Mango classification using deep learning," *International Journal of Academic Engineering Research (IJAER)*, vol. 3, no. 12, pp. 22–29, 2019.
- [11] M. M. Hasan, J. P. Chopin, H. Laga, and S. J. Miklavcic, "Detection and analysis of wheat spikes using convolutional neural networks," *Plant Methods*, vol. 14, no. 1, Dec. 2018, doi: 10.1186/s13007-018-0366-8.
- [12] N. Saranya, K. Srinivasan, and S. K. P. Kumar, "Banana ripeness stage identification: a deep learning approach," *Journal of Ambient Intelligence and Humanized Computing*, May 2021, doi: 10.1007/s12652-021-03267-w.
- [13] L. Huang and D. He, "Ripe fuji apple detection model analysis in natural tree canopy," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 10, no. 7, pp. 1771–1778, Nov. 2012, doi: 10.11591/telkomnika.v10i7.1574.
- [14] S. Yadav, N. Sengar, A. Singh, A. Singh, and M. K. Dutta, "Identification of disease using deep learning and evaluation of bacteriosis in peach leaf," *Ecological Informatics*, vol. 61, Mar. 2021, doi: 10.1016/j.ecoinf.2021.101247.
- [15] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Rice false smut detection based on faster R-CNN," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1590–1595, Sep. 2020, doi: 10.11591/ijeecs.v19.i3.pp1590-1595.
- [16] B. S. Bari *et al.*, "A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework," *PeerJ Computer Science*, vol. 7, Apr. 2021, doi: 10.7717/peerj-cs.432.
- [17] C. R. Rahman *et al.*, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosystems Engineering*, vol. 194, pp. 112–120, Jun. 2020, doi: 10.1016/j.biosystemseng.2020.03.020.
- [18] M. S. I. Prottasha, "Rice plant disease dataset," *archive.org*. [Online]. Available: <http://archive.org/details/ricediseasedataset>
- [19] Y. Tagami, "Epidemiology of the bacterial leaf blight of rice," *Japanese Journal of Phytopathology*, vol. 31, no. 1, pp. 145–151, 1965, doi: 10.3186/jjphytopath.31.Special1\_145.
- [20] S. Sunder, R. Singh, and R. Agarwal, "Brown spot of rice: an overview," *Indian Phytopathology*, vol. 67, no. 3, pp. 201–215, 2014.
- [21] H. Ikegami, "Studies on the false smut of rice," *Japanese Journal of Phytopathology*, vol. 27, no. 1, pp. 16–23, 1962, doi: 10.3186/jjphytopath.27.16.
- [22] E. C. Roumen and W. S. de Boef, "Latent period to leaf blast in rice and its importance as a component of partial resistance," *Euphytica*, vol. 69, no. 3, pp. 185–190, Jan. 1993, doi: 10.1007/BF00022364.
- [23] J. Xue *et al.*, "Transcriptome analysis of the brown planthopper *nilaparvata lugens*," *PLoS ONE*, vol. 5, no. 12, Dec. 2010, doi: 10.1371/journal.pone.0014233.
- [24] M. D. Pathak and Z. R. Khan, *Insect pests of rice*. Los Banos: International Rice Research Institute, 1994.
- [25] S. M. Pizer *et al.*, "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, Sep. 1987, doi: 10.1016/S0734-189X(87)80186-X.
- [26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2014.
- [27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [29] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.
- [30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [31] J. L. Ba and D. P. Kingma, "Adam: a method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015, pp. 1–15.

**BIOGRAPHIES OF AUTHORS**

**Md. Sazzadul Islam Prottasha**    obtained his bachelor's and master's degree in Information and Communication Engineering from Bangladesh University of Professionals (BUP), Dhaka, Bangladesh. Currently he is working in multiple projects based on image processing and computer vision. His research interests include machine learning, computer vision, image processing, big data analytics, artificial intelligence, and IoT. He can be contacted at email: 19541026@bup.edu.bd.



**Sayed Mohsin Salim Reza**    is an Assistant Professor at the Department of Information and Communication Technology, Bangladesh University of Professionals (BUP), Dhaka, Bangladesh. He is a member of a variety of professional organizations. He is an effective planner and operator, possessing expertise in overseeing academic and administrative operations including organizational standards, and computer science training for faculty. He has published over 50 research articles in ISI/Scopus-indexed journals and several renowned international conferences. He can be contacted at email: salim.reza@bup.edu.bd.