Deep Learning Based Method to Predict Plant Diseases: A Case Study with Rice Plant Disease Prediction

*

Mohammad Asifur Rahim[1] , Rumana Akter[2], Ashif Reza[3], Tauhidur Rahman[4], Prof.Dr.Mohammad Shafiul Alam[5]
Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
mohammadasifurrahim@gmail.com [1],rumananipa143@gmail.com[2]
ashifcse247@gmail.com[3], rahmantauhidur33@gmail.com [4], shuvo23@gmail.com[5]

***Abstract*—Bangladesh, a nation that largely depends on rice for both its economy and dietary needs, is the target of the project, which intends to establish an easy and effective approach for recognizing rice plant diseases there. The research made use of an original dataset consisting of 5932 pictures of four various forms of rice plant illnesses. In order to extract features, We leveraged Convolutional Neural Networks (CNN) with Transfer Learning techniques and pre-trained models like VGG16, Inception-V3, VGG19, and ResNet-50. To increase the training and test accuracy, augmentation methods including flipping, zooming, and resizing the dataset were also performed. The VGG-19 model exceeded all other models with a test accuracy of 98%, while the customized CNN model obtained an accuracy of 97%. In addition, traditional machine learning models such as KNN, SVM, AdaBoost, Decision Tree, and Random Forest are used, with the Random Forest classifier achieving the highest accuracy of 96.6%. The suggested model performed better than the majority of earlier studies that have been done on rice plant diseases in Bangladesh but used smaller datasets and showed lower performance levels.**

***Index Terms*—Rice Plant Diseases, CNN, Transfer Learning, Gabor, Sobel**

I. Introduction

Bangladesh is an agricultural country. Rice is a major food for the people of our country. Rice cultivation provides about 48%[1] of rural development. Around 75% of the overall harvested area and above 80% of the overall irrigated zone are cultivated through rice plants[2]. Consequently, Rice plays an important role for the people of Bangladesh. But the quality and quantity of rice production may be decreased because of Rice plant Diseases. Having the disease in plants is quite natural. If proper step is not taken in this regard then it causes serious effects on Rice plants. Therefore, it is the major task to identify Rice plant Disease in the early stages. Early detection of Rice plant diseases is very crucial for the crop protection system of our country.

At present, Most of the farmers of our country use their own experience to detect diseases, farmers use pesticides in excessive quantities which can not help in the prevention of disease but can have a malignant effect on plants. Thus their misclassification creates a bad impact on rice cultivation. So, they need advice from rice disease specialists. In remote or rural areas, Rice disease specialists are not able to give quick remedies or advice to the farmers at the right time and they also require expensive equipment and a large amount of time for manually identifying and classifying rice diseases. Moreover, traditional visual observation methods are mostly inaccurate. Besides that laboratory testing requires time and can be expensive. The research that has been done on Bangladeshi Rice plant disease is not sufficient. Most of the research is done over a small number of datasets. Nowadays smartphones are available. If the farmers can somehow afford a smartphone it will be easier for them to use image-processing methods to detect Rice plant diseases. Therefore, an automated identification and classification system can help farmers to accurately detect plant diseases and take necessary measures beforehand.

Correspondingly, We tried to classify some severe Rice plant diseases leveraging machine learning and deep learning techniques. The use of advanced DL models to address the complexity and effectiveness of precision agriculture research in recent years [2] provides evidence that this is indeed the case. The following set of contributions to the research are made in light of our investigation.

- We focused on creating a customized CNN architecture that can get high accuracy in classifying four major Rice plant diseases from the leaves. Moreover, to overcome the

[1]http://www.knowledgebank-brri.org/riceinban.php
[2]http://www.knowledgebank-brri.org/riceinban.php

small dataset issue augmentation techniques have been used to increase the size of the dataset which eventually resulted in getting higher accuracies in classifying diseases.
- We used unique feature extraction features in our images to get higher accuracies using machine learning models.
- Made a comparison analysis between our results and with other previous research as well where we got higher accuracies than most of the previous research.

II. Related Works

Md Jahid Hasan et al. [3] proposed a paper for rice disease identification and classification by integrating a support vector machine with a Deep Convolutional Neural Network (D-CNN). Transfer Learning techniques were used to improve the proposed model. The paper uses a complex AI system that integrates SVM and D-CNN for identifying rice plant diseases. The proposed model achieves an accuracy of 97.5%. However, the dataset used for training is a small collection of 1080 images and for testing, it was 270 images. The proposed methodology is best suited for a smaller number of images.

Tejas Tawde et al. [4] proposed a paper for rice plant disease detection and classification using Convolutional Neural Network (CNN) to distinguish different methods based on the classifier used. A hardware prototype and model using CNN are proposed.

Kawcher Ahmed et al. [5] proposed a model for rice leaf disease detection using machine learning techniques and CNN. They achieved 96.77% accuracy when trained using KNN, Logistic Regression, j48, and Naive Bayes, but only 400+ images were used for the training process. Three plant diseases were detected and no deep-learning models were used. Image classification gives better accuracy with faster performance with deep learning models, but machine learning models used in this proposed model generally give a slower performance.

S Ramesh et al. [6] proposed a model for Rice Blast Disease detection and classification using a machine learning algorithm. Images were classified using ANN and K-Means Clustering. 99% accuracy was achieved for blast-infected images and 86% for healthy images. However, the model showed over-fitting tendencies due to only 300 leaf samples used as the dataset.

Anam Islam et al. [7] proposed a model for rice leaf disease recognition using Local threshold-based segmentation and Deep CNN. Three CNN architecture models VGG16, ResNet50, and Dense-Net121 are used for classification. A dataset of 786 images was used for training, with a testing accuracy of 78.84%.

S Ramesh et al. [8] proposed a model for the application of machine learning in the detection of blast disease in South Indian rice crops. K-means Clustering is used for image segmentation and an ANN classifier is used to determine whether an image is an image of an infected crop or not. The ANN classifier gave 99 accuracies for normal images and 100% for blast-infected images, but it only detects a single type of disease (Rice Blast). Other methods give a more accurate result.

Md. Sazzadul Islam et al. [9] proposed a model for the Identification of Various Rice Plant Diseases Using Optimized Convolutional Neural Networks (CNN). 1677 images were used to increase the size of the dataset, and the model uses a depth-wise convolution to reduce computational cost and parameter size. However, a large number of classes is causing bias and misclassifications among some of the diseases with similar characteristics.

Vimal K. Shrivastava et al. [10] proposed a model for Rice Plant Disease Classification using a Deep Convolutional Neural Network (CNN) as a feature extractor and a Support Vector Machine (SVM) as a classifier. The accuracy achieved was 91.37%À large ImageNet dataset of 1.2 Million Images 1000 classes was used, but a relatively small dataset of 619 images was used for identifying 3 types of rice diseases. The paper also mentioned the unavailability of labeled rice disease images.

III. Methodology

Traditional machine learning models and deep learning models as classifiers for comparison have both been used to detect plant diseases. The proposed model has been tested using five machine learning algorithms, including KNN, Adaboost, Decision Tree, Random Forest, etc in the classification step of classical machine learning. For classifiers using deep learning, for greater model accuracy, a customized CNN model and transfer learning approaches have been applied. Augmentation techniques have been applied to train the model with deep learning models. In the case of machine learning, a Gabor Filter [17] followed by a Sobel Filter [18] have been used for feature extraction purposes. Our work can be divided into two approaches. The two approaches are: • Rice plant Disease Classification using a machine-learning approach • Rice plant Disease Classification using a Deep Learning Approach

For both classification cases: some steps are followed: 1. Dataset Acquisition 2. Dataset pre-processing 3. Feature Extraction 4. Classification

The proposed method that we applied in our work is shown in Fig 1.

A. Dataset Acquisition

At first, we collected a suitable dataset [1]. The dataset contains 5932 images of 4 major Rice plant diseases in Bangladesh as well as other countries. Sample images of the datasets are shown in Fig 2.

B. Dataset pre-processing

The datasets are split into two sets which are training and validation/testing sets having a ratio of 80:20 respectively. The augmentation techniques are applied to the training dataset only. Here is the total data of 5932 images where 4745 images are used for training and 1187 images for training. In training distribution, 1267 images for Bacterial Blight, 1152 for Blast, 1280 for Brown Spot, and 1046 for Tungro are used. In test

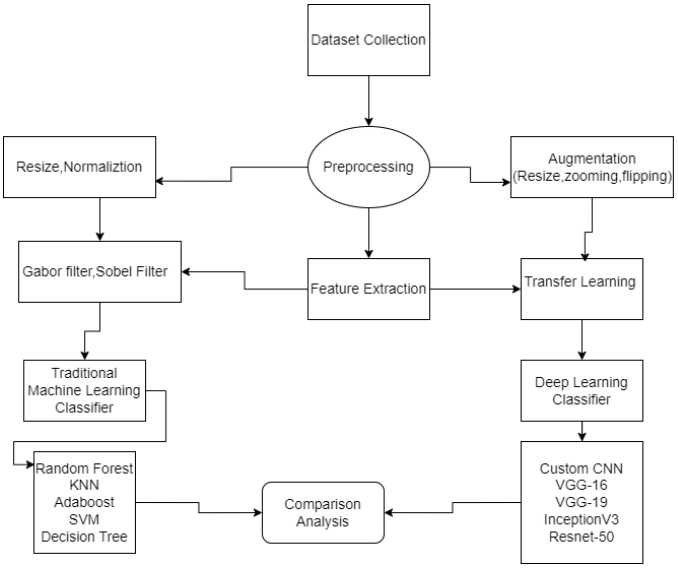

Fig. 1. Proposed model for Rice plant disease Classification

Fig. 2. Sample images

distribution, 317 images for Bacterial Blight, 288 for Blast, 320 for Brown Spot, and 262 for Tungro are used.

1) Normalization: In image processing normalization, is a process that changes the range of pixel intensity values. The possible values for each pixel are 0 to 256. The computation of high numeric values may become more difficult when using the image as it is and running through a Deep neural network. So, the pixel values are normalized by dividing the values by 255 and converting them to a range from 0 to 1.

2) Image Augmentation: For deep networks to perform well, a lot of training data is required. Image Augmentation is typically needed to improve deep network performance in order to create a powerful image classifier with little training data. As deep neural networks need lots of training data to perform well, we increased the training samples by applying several processing techniques like rotation, shifts, flips, etc. We applied zoom-range=0.50 which means zoom-in and zoom-out by 50%. Moreover, the rotation range we set to 45, which means we randomly rotated each image between 0 and 45 degrees. We set the wide-shift range and height-shift range to 0.2. It means shifting the images along the X-axis and Y-axis by 20%. And finally, for mirror reflection, we set the horizontal flip=true. Some examples of augmented images are shown in Fig. 3.

C. Feature Extraction

The technique of turning raw data into numerical features that can be handled while keeping the information in the original data set is known as feature extraction. The interesting

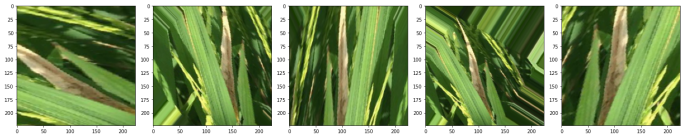

Fig. 3. Augmentad Images

portions of an image are represented as a compact feature vector using feature extraction for image data.

1) Feature Extraction for Machine Learning: For feature extraction, the Gabor filter followed by a Sobel filter of size 3X3 is used. Here 256 filters are generated and convoluted over the training and testing images. For Gabor kernels, the kernel size is set to 5X5.

Gabor Features	Kernel Size	Theta	Sigma	Lambda	Gamma
256	5×5	0 - .78	1 & 3	0 - .78	.05 & .5

TABLE I
PARAMETERS OF GABOR KERNELS

Table I The Gabor features have been used over the training images for a total of 256 times for the Machine Learning model. With each iteration, the images undergo a feature extraction process based on the Gabor filter's different parameters. The filter uses parameters that control orientation(Theta), Phase Offset(Lambda), Gaussian Factor (Sigma), and Ellipticity(Gamma). All of these properties dictate the way how the information is extracted from the training images. This allows for the training to have images with highly crucial features required to detect the particular type of disease for a specific label. Figure 4 shows different stages of feature extraction with each iteration.

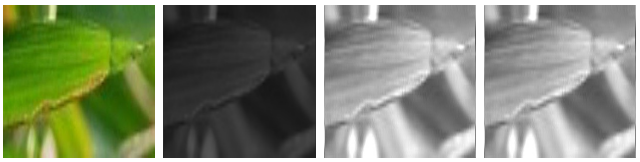

Fig. 4. (From left to right)- Normal image, Seventh Stage of, 63th stage and 256th stage of Gabor Fea-tures

2) Feature Extraction for Deep Learning: For Deep learning models, the transfer learning technique was applied for feature extraction purposes. This is advantageous because many of the low-level traits that have been discovered from a large amount of immediately accessible data may be applied to tasks with less readily available data. Transfer learning is a common choice when there is a small amount of data on which to base the model. In Transfer learning, also known as ImageNet pre-training, a pre-trained network is improved upon using a new dataset or task. The practice of feature reuse, whereby features obtained from ImageNet are useful for other datasets or tasks, implicitly justifies the practice. We chose Vgg-16, Vgg-19, and Inception-V3 as our pre-trained model for feature extraction. We followed the strategy by keeping all

the top layer's weights untrainable and feeding them to the classifier. We also built a custom CNN model for comparison purposes.

D. Model Architecture Of Proposed CNN Model

A custom CNN model has been proposed by our own to train the preprocessed dataset. The model consists of 11 layers of which 10 are hidden layers and 1 input and 1 out layer. The layers include a convolutional layer, pooling layer, flattened layer, and fully-connected or dense layer. Relu and Softmax activation functions are used.

- Layer 1: At first a convolutional layer has been used with an input of 224x224x3.Here 224x224 is the input size of the image and 1 is the number of channels or depth. The number of filters used here is 16 and there is no padding in the layer. stride, S=1, and filter size, F=3. So from the input, we get the following output image size,
  - output depth, D2=number of filters=64
  - output width, W2 = (224-3+2x0)/1+1=222 and
  - output height, H2 = (224-3+2x0)/1+1=222
  - output volume is 222x222x16
  - number of parameters = 3x3x3x16+16=448
- Layer 2: Here, a pooling layer has been used to re-size the input and extract the important pixel values. The Max pooling technique is applied here. The input volume is 222×222×16. For pooling, we need the value for spatial extend, filter, and stride, S. We have used F = 2 and S = 2 which is a very commonly used size. So from the input volume, we get the following output image size,
  - output depth, D2=number of filters=16
  - output width, W2 = $\lfloor 222/2 \rfloor$=111 and
  - output height, H2 = $\lfloor 222/2 \rfloor$=111 and
  - output volume is 111x111x16
  - number of parameters=0
- Layer 3: And Then a convolutional layer is used with the output from the previous layer which is the input of this layer, 111x111x16.Here 111x111 is the input size of the image and 16 is the number of channels or depth. The number of filters used here is 32 and there is no padding in the layer. Stride, S=1, and filter size, F=3. So from the input, we get the following information
  - output depth, D2=number of filters=32
  - output width, W2 = (111-3+2x0)/1+1=109 and
  - output height, H2 = (111-3+2x0)/1+1=109 and
  - output volume is 109x109x32
  - number of parameters=3x3x16x32+32=4640
- Layer 4: Then, another pooling layer was used to resize the input and extract the important pixel values. The Max pooling technique is applied here. The input volume is 109×109×32. For filter or pooling, we need the value for spatial to extend, filter, and stride, S. We have used F = 2 and S = 2 which is a very commonly used size. So from the input volume, we get the following output image size
  - output depth, D2=number of filters=16
  - output width, W2 = $\lfloor 109/2 \rfloor$=54 and
  - output height, H2 = $\lfloor 109/2 \rfloor$=54 and
  - output volume is 54x54x32
  - number of parameters = 0
- Layer 5: And then a convolutional layer has been used with the output from the previous layer which is the input of this layer, 54x54x32. Here 54x54 is the input size of the image and 32 is the number of channels or depth. The number of filters used here is 64 and there is no padding in the layer. Stride, S=1, and filter size, F=3. So from the input, we get the following information
  - output depth, D2=number of filters=64
  - output width, W2 = (54-3+2x0)/1+1=52 and
  - output height, H2 = (54-3+2x0)/1+1=52 and
  - output volume is 52x52x64
  - number of parameters=3x3x32x64+64=18496
- Layer 6: Then, another pooling layer has been used to resize the input and extract the important pixel values. The Max pooling technique is applied here. The input volume is 52×52×64. For filter or pooling, we need the value for spatial to extend, filter, and stride, S. We have used F = 2 and S = 2 which is a very commonly used size. So from the input volume, we get the following output image size
  - output depth, D2=Number of filters=16
  - output width, W2 = $\lfloor 52/2 \rfloor$=26 and
  - output Height, H2 = $\lfloor 52/2 \rfloor$=26 and
  - output volume is 26x26x64
  - number of parameters=0
- Layer 7: Then, another flattened layer has been used to the input size of 26x26x64 from the input volume we get the following output image size
  - output image size D2=26x26x64=43264
  - number of parameters=0
- Layer 8: Then, a dropout layer has been added to the input size of 43264 from the input volume we get the following output image size
  - output image size D2=26x26x64=43264
  - number of parameters=0
- Layer 9: Then, a dense layer has been added to the input size of 43264 from the input volume we get the following output image size
  - output image size D2=128
  - number of parameters=43264x128+128=5537920
- Layer 10: Then, another dropout layer has been added to the input size of 5537920 from the input volume, and using dropout of (.2) ratio we get the following output image size
  - output image size D2=128
  - number of parameters=0
- Layer 11: Then, another dense layer has been added to the input size of 128 from the input volume and we get the following output information
  - number of classes D2=4
  - number of parameters=128X4+128=516

Here softmax activation has been used for classification purposes.

IV. Result and Discussion

A. Performance Analysis for Machine Learning Models

1) Random Forest: The model used here for the performance analysis using machine learning is Random Forest.

Class Name	Precision	Recall	F1-score
Bacterial blight	0.95	0.99	0.97
Blast	0.99	0.90	0.94
Brown spot	0.96	0.99	0.98
Tungro	1.00	0.93	0.96
weigted avg	0.97	0.97	0.97

TABLE II
Result of Evaluation Metrics for Random Forest

Table II shows the precision, recall, and overall F1-score achieved for every individual disease type. The weighted average precision, recall and F1-score score this model showed

2) KNN (K-Nearest Neighbour): KNN is one of the more simpler machine learning algorithms, which is classified by determining the neighbors around a certain label or class and the maximum euclidean distance is considered for the label.

Class Name	Precision	Recall	F1-score
Bacterial blight	0.98	0.68	0.80
Blast	0.68	0.69	0.69
Brown spot	0.67	0.91	0.77
Tungro	0.95	0.50	0.65
weigted avg	0.80	0.75	0.75

TABLE III
Result of Evaluation Metrics for KNN

Table III shows the precision, recall, and F1-Score for all the classes.

3) Adaboost: Adaboost being an Adaptive boosting algorithm is used as an Ensemble method.

Class Name	Precision	Recall	F1-score
Bacterial blight	0.86	0.94	0.90
Blast	0.91	0.80	0.85
Brown spot	0.92	0.94	0.93
Tungro	0.95	0.86	0.91
weigted avg	0.90	0.90	0.90

TABLE IV
Result of Evaluation Metrics for Adaboost

The table IV shows that the precision achieved for rice Tungro is the highest whereas the Brown spot gives the best recall score. Brown spot is observed to have the overall best score out of all other diseases.

4) SVM (Support Vector Machine): The SVM algorithm is effective in cases where the number of dimensions is greater than the number of samples. However this machine learning algorithm works by putting data points, above and below the classifying hyperplane, so this means there is no probabilistic explanation for the classification.

Class Name	Precision	Recall	F1-score
Bacterial blight	0.81	0.94	0.87
Blast	0.90	0.47	0.62
Brown spot	0.78	0.93	0.85
Tungro	0.89	0.58	0.71
weigted avg	0.82	0.81	0.79

TABLE V
Result of Evaluation Metrics for SVM

The table V shows that Rice Blast has the most precise classification since there are a lesser number of false positives. However, in terms of recall, it shows an unsatisfactory value.

5) Decision Tree: Compared to other algorithms decision trees require less effort for data preparation during preprocessing.

ClassName	Precision	Recall	F1-score
Bacterial blight	0.97	0.89	0.93
Blast	0.91	0.80	0.85
Brown spot	0.94	0.93	0.94
Tungro	0.53	0.89	0.66
weigted avg	0.91	0.89	0.89

TABLE VI
Result of Evaluation Metrics for Decision Tree

Here it is observed that the values are relatively less consistent compared to the deep learning models as shown in table XIV.

Classifiers	Accuracy	Precision	Recall	F1-score
Random Forest	0.966	0.97	0.97	0.97
KNN	0.754	0.80	0.75	0.75
AdaBoost	0.902	0.90	0.90	0.90
SVM	0.81	0.82	0.81	0.79
Decision Tree	0.89	0.91	0.89	0.89

TABLE VII
Result of Evaluation Metrics for Machine Learning Models

It is observed that Random Forest, Adaboost and Decision Tree perform the best out of all the machine learning models. Their precision and recall scores are also consistent showing relatively low numbers of false detections. SVM is relatively sound and its metrics show similar accuracy. However, KNN seems to have disparity in it's precision and recall scores. This means KNN yields a high number of false negatives compared to false positives. This means the algorithm falsely detects healthy leaves which leads to problematic results. So, in terms of machine learning models, Random Forest shows impressive accuracy of **96.6%** and also shows the best metrics.

B. Performance Analysis for Deep Learning Models

The table displays the different combinations for batch sizes we have considered for better hyperparameter tuning. It shows different accuracies for different batch sizes. Batches of 32, 16, and 8 have been considered. An epoch of 40 has been considered based on optimal conditions and our limited resources. Adam has been Used as Optimizer, The Learning rate was set to .0001 and

Class Name	Batch Size	Epoch	Accuracy (%)
Custom CNN	32	40	85
	16		88
	8		97
InceptionV3	32	40	89
	16		90
	8		98
VGG-16	32	40	91
	16		94
	8		97
VGG-19	32	40	89
	16		95
	8		98
ResNet-50	32	40	69
	16		65
	8		71

TABLE VIII
HYPERPARAMETER TUNING

It can be observed in VIII that the accuracy for all the deep learning models, shows better results for a batch size of 8.

1) Custom Model: The custom model proposed for the deep learning approach outperforms the pre-trained CNN model resnet=50 and gives equivalent results for all the other CNN models used for the deep learning approach. But the parameter size for our custom model is around 5 million. This parameter size is much smaller compared to the pre-trained models used in this study. This makes the proposed model much more lightweight and easy to process compared to the other heavy CNN models.

Class Name	Precision	Recall	F1-score
Bacterial blight	0.91	1.00	0.95
Blast	1.00	0.88	0.94
Brown spot	1.00	1.00	1.00
Tungro	1.00	1.00	1.00
weigted avg	0.97	0.97	0.97

TABLE IX
RESULT OF EVALUATION METRICS FOR CUSTOM MODEL

The table shows the superiority of the deep learning models compared to every machine learning model used in this study. The proposed custom model outperforms the random forest model and gives an accuracy of **97%**. Since this is a deep learning model it also performs much faster compared to the machine learning model.

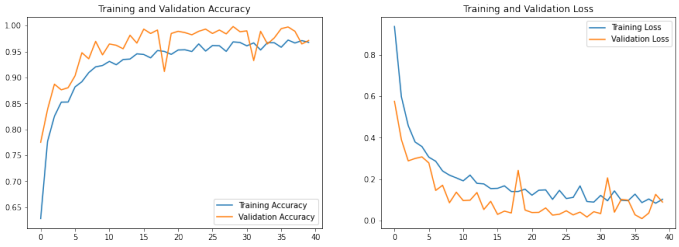

Fig. 5. Training vs Validation (Accuracy & Loss) for **Custom Model**

Both the Accuracy and Loss graphs show that the training and validation move uniformly. The accuracy steadily increases and loss steadily declines. This is the desired behavior for a deeo learning model. Besides some very minor fluctuations, the model performs very well throughout the training and validation process.

2) VGG-19: The pre-trained CNN model VGG-19 was chosen for the deep learning models in this study. Since this is a CNN model 19 layers deep, it has a large number of parameters (around 138 Million). Since every layer isn't important, they can be dropped. It is still a large number of parameters even after adding dropout. Since this is a model trained by millions of images, it performs very well as a CNN model.

Class Name	Precision	Recall	F1-score
Bacterial blight	0.99	0.98	0.99
Blast	0.99	0.94	0.97
Brown spot	0.95	1.00	0.98
Tungro	1.00	1.00	1.00
weigted avg	0.98	0.98	0.98

TABLE X
RESULT OF EVALUATION METRICS FOR VGG-19

The evaluation metrics show that the precision and recall achieved for almost every disease are near perfect for this model. It is observed that Rice Tungro gives a perfect score which means every image used in the validation set has been successfully classified for that disease. Thus giving an accuracy of **98%** which is the highest accuracy achieved for a deep learning model.

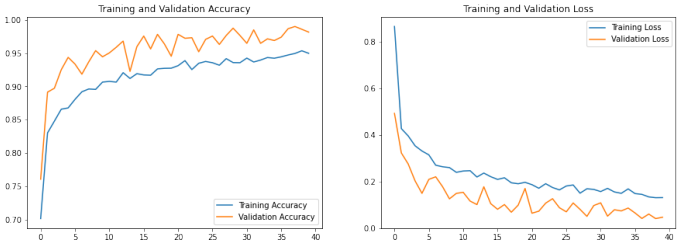

Fig. 6. Training vs Validation (Accuracy & Loss) for **VGG-19**

The training vs validation graphs shows that the process does not yield any overfitting results. The validation accuracy and loss are uniformly increasing and decreasing

respectively. This shows the result is near perfect with little to no false positives or negatives.

3) InceptionV3: InceptionV3 is one of the more preferred CNN models that are pre-trained. It has a deeper network compared to the Inception V1 and V2 models, but its speed isn't compromised. The efficiency of this model is really impressive. It is also computationally less expensive thus giving it a better edge against other CNN models.

Class Name	Precision	Recall	F1-score
Bacterial blight	0.99	0.99	0.99
Blast	0.92	1.00	0.96
Brown spot	1.00	0.93	0.96
Tungro	1.00	1.00	1.00
weigted avg	0.98	0.98	0.98

TABLE XI
RESULT OF EVALUATION METRICS FOR INCEPTIONV3

The table shows a clear idea that there is almost no false positives in terms of all the diseases. Rice Tungro seems to give perfect results for both precision and recall. So the accuracy for the InceptionV3 model is **98%** which is also the highest accuracy achieved out of all models.

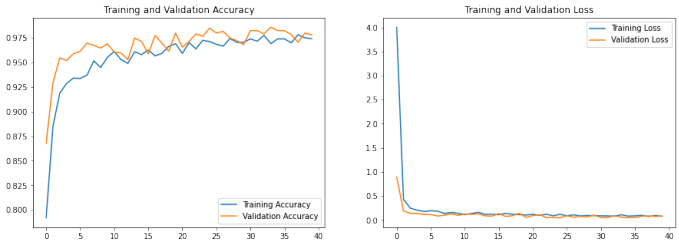

Fig. 7. Training vs Validation (Accuracy & Loss) for **InceptionV3**

The training vs validation graphs shows that the process does not yield any overfitting results. The validation accuracy and loss are uniformly increasing and decreasing respectively. This shows the result is near perfect with little to no false positives or negatives.

4) Resnet-50: This model has a very deep neural network of 50 layers as the name suggests. Its parameter size however is not as deep as the VGG network models. The resnet-50 model is a complex network due to its depth. The main disadvantage of ResNets is that for a deeper network, the detection of errors becomes difficult.

Class Name	Precision	Recall	F1-score
Bacterial blight	0.70	0.75	0.73
Blast	0.73	0.35	0.47
Brown spot	0.76	0.80	0.78
Tungro	0.67	0.96	0.79
Weighted avg	0.72	0.71	0.69

TABLE XII
RESULT OF EVALUATION METRICS FOR RESNET-50

When deeper networks are able to start converging, a degradation problem is exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly.

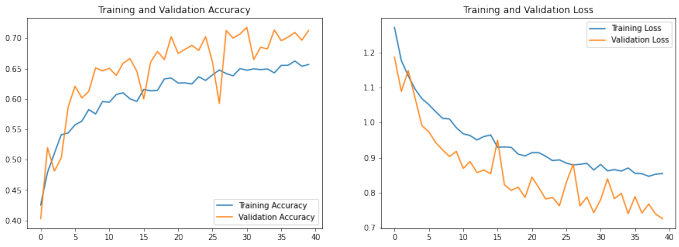

Fig. 8. Training vs Validation (Accuracy & Loss) for **Resnet-50**

The training and validation curves for both accuracy loss seem to be showing an increasing and decreasing trend. However, this seems to have lots of Fluctuations throughout the process. This means lots of false positives and false negatives have been classified here. This is why resnet-50 gives the least satisfactory result of **71%**

5) VGG-16: The other VGG network model used in our work is the VGG-16. So, in this case, the deep learning model, the VGG-16 performs similarly to the previously used VGG-19 model.

	Precision	Recall	F1-score
Bacterial blight	1.00	0.93	0.96
Blast	1.00	0.94	0.97
Brown spot	0.89	1.00	0.94
Tungro	1.00	1.00	1.00
weigted avg	0.97	0.97	0.97

TABLE XIII
RESULT OF EVALUATION METRICS FOR VGG-16

However, The total number of parameters in this model is over 138M, and the size of the model is over 500MB. Table XIII shows that the precision, recall and F1-score achieved using VGG-16 are around **97%**.

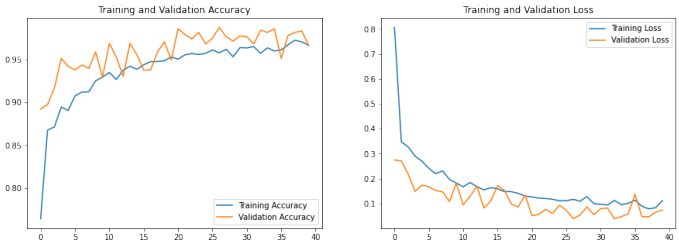

Fig. 9. Training vs Validation (Accuracy & Loss) for **VGG-16**

The training vs. validation graphs show that the process does not yield any overfitting results. The validation accuracy and loss are uniformly increasing and decreasing respectively up to 40 epochs.

C. Comparison Between Deep Learning Models

Table XIV However, in this case, almost every model shows that the precision and recall values are similar.

Classifiers	Accuracy	Precision	Recall	F1-score
Custom CNN	0.97	0.97	0.97	0.97
VGG-16	0.97	0.97	0.97	0.97
VGG-19	0.98	0.98	0.98	0.98
Inception-V3	0.98	0.98	0.98	0.98
ResNet-50	0.71	0.72	0.71	0.69

TABLE XIV
RESULT OF EVALUATION METRICS FOR DEEP LEARNING MODELS

Besides ResNet-50, all the other models yield results **97%** and above. Resnet-50 produces the lowest accuracy with **.71%**

V. CONCLUSION AND FUTURE WORK

Performance analysis of automated Rice plant disease classification is performed in our work. Moreover, five traditional classifiers are applied to classify Rice plant diseases in our images. Then a custom CNN model is designed on our own and applied to include deep learning in our work. Transfer Learning Techniques are applied in our work to get higher accuracy. Here, a dataset consisting of 5932 images was used of four different Rice plant diseases. They have split into 80 and 20 ratios. Various experiments are applied over the datasets by tuning the parameters and hyperparameters of the model. And among them, the best ones are represented. Our custom CNN model produced an accuracy of 97 percent. There are more opportunities for improvement or research on our work in the future.

- First of all, dataset size can be increased. The bigger the number of images the better the model is trained
- More types of Rice plant diseases can be added to classify a variety of Rice plant diseases
- the system can be transformed into a web application or mobile Application. Using the mobile application farmers can automatically detect Rice plant diseases

REFERENCES

[1] Sethy, Prabira Kumar. "Rice Leaf Disease Image Samples." Mendeley Data, V1, doi: 10.17632/fwcj7stb8r.1, 2020.

[2] Spyridon Mourtzinis, Paul D. Esker , James E. Specht3 Shawn P. Conley."Advancing agricultural research using machine learning algorithms" *Scientific Reports*, 2023.

[3] Kingma, Diederik P., and Jimmy Ba. "Adam: A Method for Stochastic Optimization." *arXiv preprint arXiv:1412.6980*, 2017.

[4] Hasan, Md Jahid, Mahbub, Shamim Alom, and Md. Shahin Nasim. "Rice Disease Identification and Classification by Integrating Support Vector Machine With Deep Convolutional Neural Network." In *2019 International Conference on Advanced Systems in Electrical Engineering and Computer Science (ICASECS)*, pp. 1-6, doi: 10.1109/ICASERT.2019.8934568, 2019.

[5] Tawde, Tejas, Kunal Deshmukh, Lobhas Verekar, Ajay Reddy, Shailendra Aswale, and Pratiksha Shetgaonkar. "Rice Plant Disease Detection and Classification Techniques: A Survey." *International Journal of Engineering Research Technology (IJERT)*, Volume 10, Issue 07, 2021.

[6] Ahmed, K., Shahidi, T. R., Alam, S. M. Irfanul, and Momen, S. "Rice Leaf Disease Detection Using Machine Learning Techniques." In *International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1-5, doi: 10.1109/STI47673.2019.9068096, 2019.

[7] Ramesh, S. "Rice Blast Disease Detection and Classification Using Machine Learning Algorithm." In *2018 International Conference on Modern Electrical and Energy Systems (ICMEES)*, pp. 255-259, doi: 10.1109/ICMETE.2018.00063, 2018.

[8] Islam, Anam, Redoun Haque, S. M. Rafizul Islam, S.M. Khan, and Mohammad. "Rice Leaf Disease Recognition using Local Threshold Based Segmentation and Deep CNN." *International Journal of Intelligent Systems and Applications*, 13, pp. 35-45, 2021.

[9] Ramesh, S., and Vydek, D. "Application of machine learning in the detection of blast disease in South Indian rice crops."*International Journal of Advanced Research in Computer and Communication Engineering*, 6, issue. 4, 2017.

[10] Prottasha, Md. Sazzadul Islam, A. B. M. Kabir Hossain, Md. Zihadur Rahman, S M Salim Reza, and Dilshad Ara Hossain. "Identification of Various Rice Plant Diseases Using Optimized Convolutional Neural Network." 2021.

[11] Shrivastava, V. K., Pradhan, M. K., Minz, S., and Thakur, M. P. "Rice Plant Disease Classification Using Transfer Learning of Deep Convolutional Neural Network." 2019.

[12] International Rice Research Institute (IRRI) Rice Knowledge Bank. https://www.irri.org/rice-knowledge-bank

[13] Miah, S.A., and Shahjahan, A.K.M. "A Survey of rice diseases in Bangladesh." *International Journal of Pest Management*, 2008.

[14] Sourabh Kumar Kashyap1 , Jitendra Kumar Mishra2 "A Review Paper on Gabor Filter Algorithm for The application of Texture Segmentation." , 2008.

[15] International Rice Research Institute (IRRI) Rice Doctor - Bacterial Blight. http://www.knowledgebank.irri.org/decision-tools/rice-doctor/rice-doctor-factsheets/item/bacterial-blight

[16] International Rice Research Institute (IRRI) Rice Doctor - Blast (Leaf Collar). http://www.knowledgebank.irri.org/training/fact-sheets/pestmanagement/diseases/item/blast-leaf-collar

[17] International Rice Research Institute (IRRI) Rice Knowledge Bank - Pest Management Fact Sheets. http://www.knowledgebank.irri.org/training/fact-sheets/pest-management

[18] International Rice Research Institute (IRRI) Rice Knowledge Bank - Tungro. http://www.knowledgebank.irri.org/training/fact-sheets/pestmanagement/diseases/item/tungro

[19] Kashyap, Sourabh Kumar, Jitendra Kumar Mishra. "A Review Paper on Gabor Filter Algorithm for The application of Texture Segmentation." 2017.

[20] Jana, Susovan, Ranjan Parekh, and Bijay Sarkar. "A semi-supervised approach for automatic detection and segmentation of optic disc from retinal fundus image." 2021.

[21] Andrew P.Bradley,"The use of the area under the ROC curve in the evaluation of machine learning algorithms",1997. https://www.sciencedirect.com/science/article/abs/pii/S0031320396001422 2021.