

Plant Diseases Classification Using Pre-trained and Transfer Learning Models: A Study on Rice Leaves

Marjan Mavaddat^{ID}, Marjan Naderan*^{ID}, Seyed Enayatallah Alavi^{ID}

Department of Computer Engineering, Faculty of Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran

ABSTRACT: Ocular detection of pests by phytosanitary specialists, as a very imperative and challenging task, appears to be time-consuming, costly, and associated with human error in today's farming processes. In modern agriculture, diagnostic softwares by artificial intelligence are advised to be used by farmers themselves with little time and cost and with more accuracy. In this paper, two different datasets of rice leaf disease have been used with two transfer learning methods for diagnosing rice leaf disease. The first method uses a CNN-based output of a pre-trained model with an appropriate classifier. In the second method, freezing bottom layers, fine-tuning weights of the last layers of the pre-trained network, and adding an appropriate classifier to the model are proposed. For this purpose, seven CNN models have been designed and evaluated. Next, the weights of the best model among these seven proposed models is used to train a second dataset of rice leaves with more disease classes and fewer images. Using these weights, an accuracy of over 96% is reached which is higher than other comparing methods. Furthermore, Grad-CAM, heat map, and ROC diagram are used to observe the diagnostic areas of the best model.

Review History:

Received: Mar. 08, 2024
Revised: May, 25, 2025
Accepted: May, 25, 2025
Available Online: Aug. 01, 2025

Keywords:

Rice Leaf Disease
Transfer Learning
Classification
K-Fold Cross Validation
Grad-CAM

1- Introduction

Millions of hectares of agricultural land are cultivated annually for rice production, as this crop is a staple for people worldwide. Leaf diseases at various stages of growth can harm the plant, leading to a decrease in both the quantity and quality of the final yield. These diseases may be triggered by a range of climatic conditions, including fungal, bacterial, and viral infections. Some of the most significant diseases include blasts, brown spots, bacterial leaf blight, false smut, and tungro virus. However, the excessive use of pesticides to combat these diseases can result in environmental pollution and pose risks to human health.

Traditional methods of diagnosing rice leaf diseases rely on manual and visual inspection, requiring accuracy and experience from the observer. Even expert agronomists can make errors in identifying the diseases correctly. Moreover, these traditional methods are time-consuming and may not be accessible in remote areas lacking specialized agricultural knowledge. Therefore, there is an increasing need for automated detection systems to enable timely and effective preventive measures [1]-[3].

Various approaches have been proposed in the literature to classify rice leaf diseases, including traditional machine-learning methods that utilize image processing for disease classification. Some of the most important of these methods

are: Random Forest (RF), Decision Tree (DT), Gradient augmentation, and Naïve Bayes (NB) classifiers [4]. In [5] and [6], different features were separated using local binary patterns and directional gradient histograms using segmentation by the Otsu method, and classification was done with the Support Vector Machine (SVM). In [7], the appropriate features required for classification are extracted from diseased plant leaf images and given to Nearest Neighbor (NN) networks or Artificial Neural Networks (ANN) for classification.

In [8] and [9], various classifiers including updateable Naïve Bayes, Bayes Net, Part, J48, Decision stump, Logistic Model Tree (LMT), RF, Jrip, OneR, Filtered Classifier, Multi-class classifier, Instance-Based learning with parameter K (IBK), and Logistic LibSVM from Weka tools are utilized for classification. The RGB features of each class are computed based on the affected leaf area in [8]. Meanwhile, in [9], features are obtained using the Otsu on Vegetation Index (VI) method to extract energy, correlation, homogeneity, contrast, and entropy of the contaminated areas. Subsequently, the tissue value is determined using the Gray-Level Co-occurrence Matrix (GLCM) technique, and the damaged tissue is then fed into the classifier for classification.

In the realm of traditional intelligence methods, outcomes were reliant on features, and image preprocessing played a crucial role. Consequently, researchers transitioned to deep learning networks like Convolutional Neural Networks

*Corresponding author's email: m.naderan@scu.ac.ir

(CNNs) due to their ability to achieve high detection accuracy and perform automatic feature extraction [10]. For instance, in [11], a combination of K-means, SVM, and CNN clustering methods is employed. In [12], SVM classification is utilized with features extracted from the DCNN model and trained with InceptionV3. Subsequently, researchers started leveraging pre-trained networks, particularly pre-trained CNNs such as VGG16, AlexNet, ResNet152V2, InceptionV3, InceptionResNetV2, Xception, MobileNetDenseNet169, and NasNetMobile [1], [2], [3], [10], [13], [14] and [15].

In addition, models such as ResNet50, ResNet50V2, ResNet101V2, VGG19, SqueezeNet, MobileNetv2, and DENS-INCEP methods, which are a combination of DenseNet and Inception are presented in [13], [16], [17]. A proposed CNN and MobileNet model with UNet, as the basic model for image segmentation, is also proposed in [18], [19], and [20].

To study the rice leaf disease problem a number of different datasets have been prepared by the researches with different classes of diseases. The rice_leaf_diseases database in UCI repository with 120 images and three classes of False Smut, Brown Spot, Bacterial Blight and with equal distribution of 40 samples from each class is used in many previous works including [4], [5], [10], [18], [21]-[23]. It should be noted that the dataset under study contains a limited number of images and disease classes. In previous research, the riceleafs database from Kaggle was utilized, which consisted of 3355 images and four classes including Leaf blast, Hispa, Healthy leaves, and Brown Spot, as referenced in [3] and [13]. Another database mentioned in [15] consisted of 1426 images and nine classes, namely Sheath Blight, Brown Spot, Stemborer, Brown Plant Hopper (BPH), Hispa, False Smut, Neck Blast, Bacterial Leaf Blight (BLB), and Others. Additionally, [24] referred to a database available on the Mendeley site, which comprised 5932 images and four classes: Brown Spot, Leaf Blast, Tungro, and Bacterial Blight. In [25], accurate identification of rice leaf diseases with crowded backgrounds was discussed using the residual attention mechanism. A network called RiceDRA-Net was designed using a re-attention mechanism and convolution blocks and dense blocks, and a final accuracy of 99.60% was achieved. In [26], the classification of rice leaf diseases was performed based on the design of a model called ICAI-V4, which consists of Coordinate Attention, Inception-iv, and Reduction-iv modules, and the accuracy reached 95.57%. All datasets used in these studies are openly available.

As seen from previous studies, many works have been proposed for the detection of rice leaf disease which are based on different datasets. Most of the recent studies have utilized transfer learning since the datasets accumulated for this problem do not have many images. Several well-known architectures such as VGG, ResNet, Inception, and MobileNet have been investigated for this purpose. The exact use of which architecture depends on the application (whether a light detection or precise result is needed) and on the type of images in the dataset. In addition, most previous works relied on the accuracy of detection, while for multi-class

classifications the F1-score is more reliable for evaluation. On the other hand, using weights of one network trained on a dataset and transferring them to test another dataset has not been conducted in any of the previous works.

This paper presents several automated methods based on a transfer learning approach for classifying rice leaf diseases using two datasets: Mendeley's Rice Leaf Disease [24] and Rahman et al. [17] datasets. On the first dataset, two techniques are employed in applying transfer learning. The initial approach involves utilizing a pre-trained CNN model, whereas the second approach involves fixing lower layers of the pre-trained model. Subsequently, the weights of the final layers of the pre-trained network are adjusted, and a suitable classifier is incorporated into the model. As a result, a total of seven models are put forward. To achieve this, three well-known pre-trained networks - VGG16, InceptionV3, and Resnet152v2 - are utilized with varying numbers of final layers for fine-tuning in order to produce these seven models. Finally, the weights of the best model among these seven proposed models are used to train a second dataset on rice leaves, which has more disease classes and fewer images. This transition of weights from a trained network to test another dataset does not exist in previous studies, to the best of our knowledge.

The remainder of this article includes the following sections: Section 2 outlines the dataset, transfer learning basics, and the proposed networks. Section 3 presents the simulation results of the proposed models. Lastly, Section 4 covers conclusions and future works.

2- Proposed Networks

2- 1- The dataset

The dataset used in this research includes 5932 images from the Mendeley public. Rice leaf disease images were classified into four classes: bacterial blight, blast, brown spot, and tungro (Fig. 1). Each class consists of 1584, 1440, 1600, and 1308 images, respectively, which all classes have an almost uniform distribution [24]. These images were collected with the background of the field. The second public dataset is from the Resized_Original_Data folder used in [17], which contains 1426 images grouped into nine classes: pseudosmooth, brown plant hopper, bacterial leaf blight, neck blast, stemborer, hyspa, rot or pod burn, brown spot, and the last category is called others, which includes healthy leaves and stems, healthy yellow seeds, and dead leaves and stems (Fig. 2).

These classes have been collected from different stages of the disease. Utilizing images with authentic backgrounds and training the networks with them yields more realistic results compared to images where backgrounds have been eliminated, leaving only the patient leaf image.

2- 2- Basics of transfer learning

Convolutional Neural Networks require large amounts of data and resources during the training process. For instance, the ImageNet dataset consists of 14,197,122 images categorized into various subcategories, which prove to be

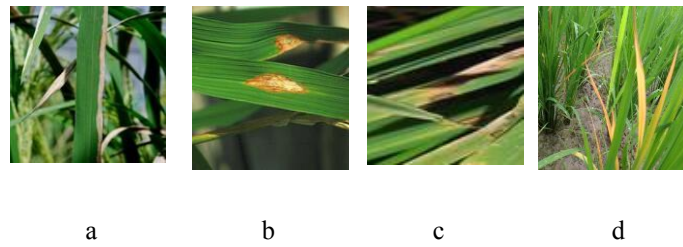


Fig. 1. Four classes of rice leaf diseases in the Mendeley dataset, a) bacterial blight, b) blast, c) brown spot, d) tungro.

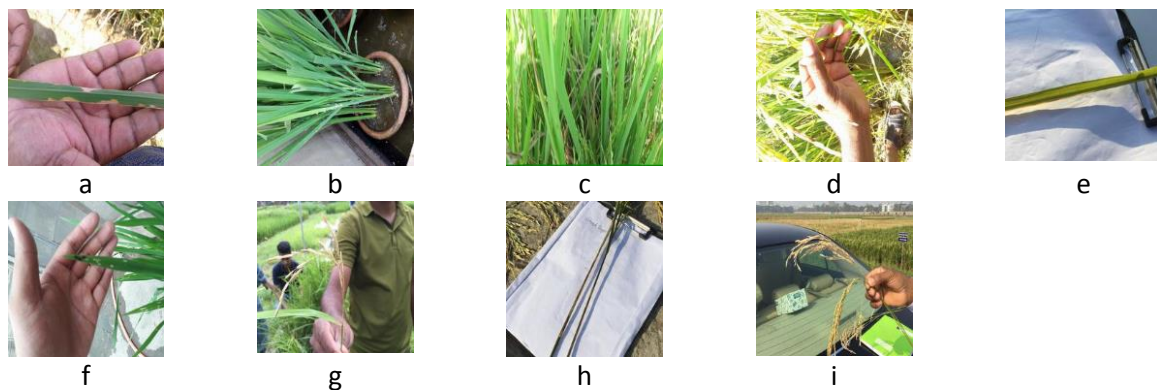


Fig. 2. Nine classes of rice leaf diseases in Rahman et al dataset, a) bacterial leaf blight, b) brown plant hopper, c) brown spot, d) false smut, e) healthy (others), f) hispa, g) neck blast, h) sheath blight rot, i) stemborer.

beneficial for supervised machine learning tasks like object localization, image classification, and object recognition. The Scale-Invariant Feature Transform (SIFT) utilized in computer vision aids in identifying local features within an image. The ImageNet dataset encompasses 1,000 image classes with SIFT features trained over a period of two to three weeks using multiple GPUs. In transfer learning, initial network layers are responsible for edge extraction, while middle layers focus on shape extraction. Only the final layers are retrained to capture clearer elements of the images. Additional layers can be incorporated into the CNN model based on specific objectives. Three strategies exist for employing pre-trained networks in transfer learning:

1. Training the complete model using a dataset.
2. Keeping the majority of the network frozen, only the final layers being retrained.
3. Fixing the base CNN weights and incorporating a new classifier.

The techniques outlined in this paper are established from strategies two and three, as our dataset contains over 1000 images per category and differs from the ImageNet dataset (even though it includes images of plants and trees). Consequently, both strategies two and three have been evaluated. The initial model proposed is built on strategy three and is accompanied by an appropriate classifier for the initial

dataset. The subsequent six models proposed are constructed on strategy two and are paired with suitable classifiers for the initial dataset.

2- 3- VGG16, InceptionV3 and Resnet152v2 pre-trained networks

Three pre-training networks were utilized in this research: VGG16 from Oxford, InceptionV3 from Google, and Resnet152v2 from Microsoft. These models were chosen as they are well-known models and previous studies have also used them as in [1], [2], [10], [16], and [17] and reached promising results. The weights for all three networks were established through training on the ImageNet dataset. An analysis comparing these three networks is presented in the study [27].

2- 4- The proposed models

VGG16 network was initially employed in the proposed approach, employing strategy 3. In this instance, the convolutional base of the network utilized ImageNet weights, while the last fully connected layer was eliminated. Subsequently, we designed our classifier at the network end, following the structure depicted in Figure 3. Additionally, image sizes were modified to 224x224x3 and the following layers were incorporated:

1. Utilizing a dense layer consisting of 2048 neurons,

employing the Relu activation function, and having feature dimensions of $7 \times 7 \times 512$ (equivalent to the final output shape of the base model VGG16 post removal of the fully connected layer). Subsequently, a dropout operation is executed with a regularization value of 0.5 to face the over-fitting problem.

2. Implementing a dense layer comprising 512 neurons, maintaining the same feature dimensions and activation function, along with a dropout operation set the same as the previous step.

3. Incorporating a dense layer with 128 neurons, keeping the feature dimensions and activation function consistent, and applying a dropout operation with the same specified value.

4. Establishing a dense layer with four neurons (as there are four classes for classification) and utilizing the Softmax activation function for the classification process.

This model is specifically tailored for VGG16 architecture and is not compatible with InceptionV3 and Resnet152v2 architectures due to its design being optimized for VGG16's dimensions.

In the second approach, fine-tuning was implemented on VGG16, InceptionV3, and Resnet152v2 networks. Initially, network weights from the ImageNet dataset were utilized for the base CNN of all three networks, excluding the last k layers, and the classifier was eliminated. Image dimensions were adjusted to $224 \times 224 \times 3$. The optimal value of k, representing the number of layers for fine-tuning the model, was determined through experimentation. Subsequently, all layers of networks were frozen, except the last six layers of

Resnet152v2, the last 12 layers of InceptionV3, and the last two layers of the VGG16 network. These specific layers were retrained. A classifier was then constructed using the Global Average Pooling (GAP) technique, resulting in the final model configuration (Fig. 4):

- A dense layer with 512 neurons, a Relu activation function, and dropout with a value of 0.3
- The same three layers are repeated once more
- A last dense layer with four neurons and a Softmax activation function

In the third approach, the selection of the optimal method for the second dataset was based on outcomes obtained from previous strategies. The preferred approach was determined by considering the results achieved on the first dataset of rice. To accomplish this, weights of the proposed model for the second strategy were derived from VGG16 network, excluding the last layer. Subsequently, a dense layer with nine neurons (representing the number of categories in the second dataset) and a Softmax activation function were added after removing the last layer, as illustrated in Figure 5. This model was then trained and evaluated for the second dataset using two specific methods:

- Using 10 fold cross validation
- Without using k fold cross validation

In addition, the weights of VGG16 network were used once again with fine-tuning the last three layers, and a two-dimensional UpSample layer was used before the final classifier in this approach.

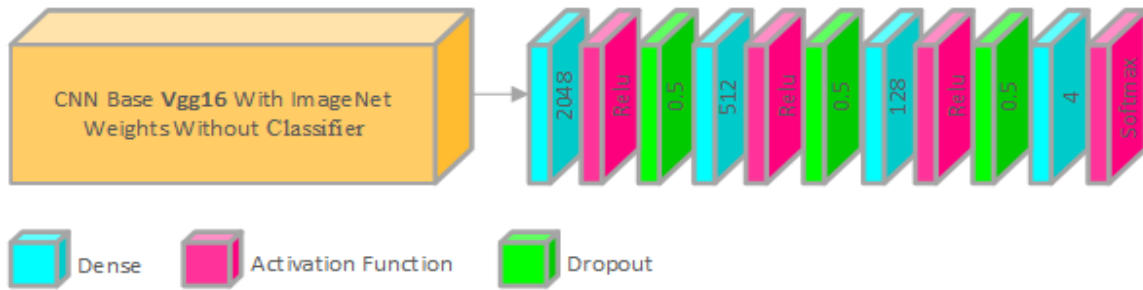


Fig. 3. First proposed strategy with VGG16 and the classifier.

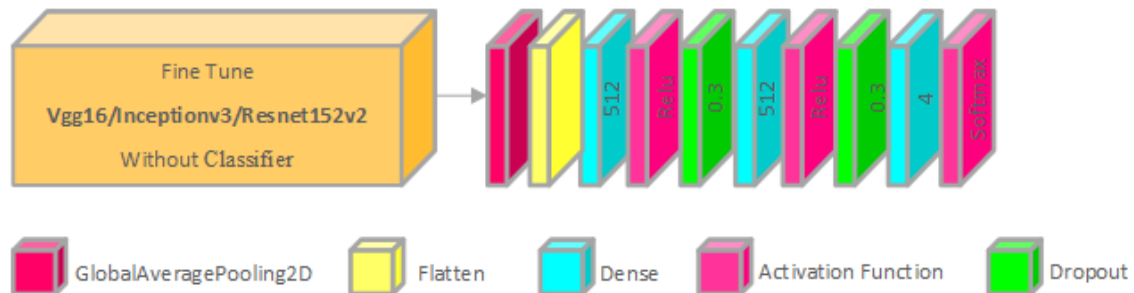


Fig. 4. Second proposed strategy with fine-tuning.

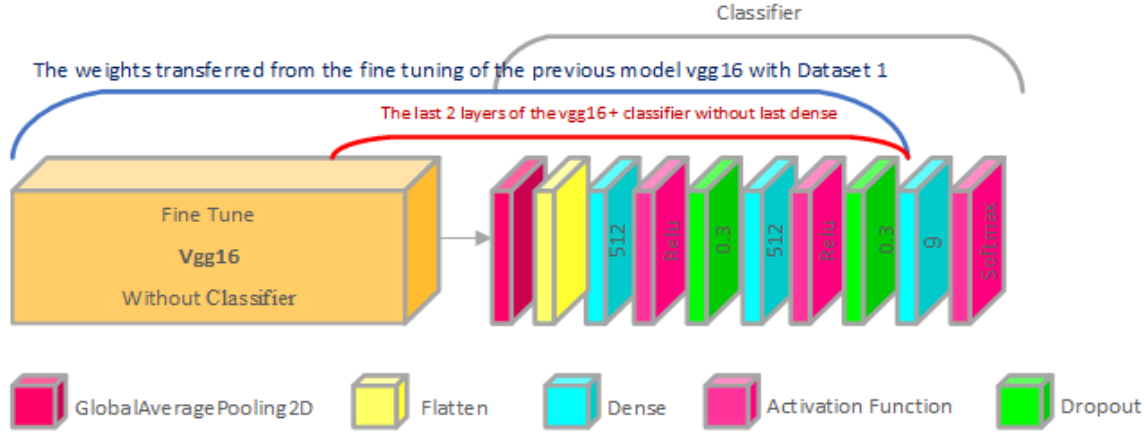


Fig. 5. Third proposed strategy with transfer learning.

3- Simulation Results and Comparison

To implement the proposed models, Python version 3.7.13, Keras version 2.8.0, Google CoLab environment, and its free GPU, Tesla k80, have been used to execute the codes. Both datasets (Mendeley and Rahman et al.) were divided into 80%-20% for training and test/validation respectively, which equals 4743 images for training and 1189 for model validation/testing for the first dataset.

3- 1- Initializations and evaluation metrics

The initial model presented in Figure 3 underwent the normalization of images and was trained using a batch size of 64. Loss values were computed using the cross-entropy function, and the model was optimized using the RMSprop function with a learning rate of 0.0001 over 30 epochs. Moving on to the second model in Figure 4, data normalization was applied, and training was conducted with 32 batches over 30 epochs. Adam optimizer was utilized with an initial learning rate of 0.001, along with a cross-entropy loss function. Feedback mechanisms were employed to adjust the learning rate during training. In cases where accuracy did not improve for three consecutive validation rounds, the learning rate was decreased by a factor of 0.4 to facilitate a slower network learning process.

For the third proposed model in Fig. 5, both methods of the third strategy were used with 10-fold and without using k-fold with the same parameters. The categorical_crossentropy cost function was used together with Adam's optimizer function with an initial learning rate of 0.001. If the accuracy of validation data did not improve for three consecutive rounds, the learning rate would be decreased by 0.4. The training was conducted with 32 batches over 30 epochs for the method without k-fold. For the 10-fold method, the value of random state was five, shuffle=true and it was executed in 100 rounds. The reason for using 100 rounds of execution was to set similar conditions with the base paper for comparison. For VGG16 with fine-tuning the last three

layers, the SGD optimizer function was used with an initial learning rate of 0.01, the image input size was 256x256, and the batch size was set to 16. The model was trained in 100 rounds with early stopping and patience with a value of 20, which means if the validation error is not reduced after 20 consecutive rounds, execution is stopped. Other parameters are the same as before.

In order to assess the effectiveness of the suggested networks, analyses were conducted on various metrics including accuracy, precision, recall, F1-score, and loss function as outlined in (1)-(4) correspondingly.

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

$$Precision = TP / (TP + FP) \quad (2)$$

$$Recall = TP / (TP + FN) \quad (3)$$

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

such that TP, TN, FP, and FN are computed as:

- True Positive (TP): where the sample is positive (a leaf is affected by the desired disease class) and the pattern is accurately identified.
- True Negative (TN): indicates that the sample is negative (a leaf has a disease other than the desired disease class) and the pattern is correctly identified.
- False Positive (FP): where the sample is positive, but its pattern is not diagnosed correctly, resulting in the identification of an incorrect disease class.
- False Negative (FN): such that the sample is negative, but the pattern is not diagnosed correctly, leading to the identification of an incorrect disease class.

Furthermore, the confusion matrix, ROC curve, and AUC diagram are used for evaluation according to the following definitions:

- Confusion matrix: shows the number of true instances of a specific class against the number of predicted class instances, such that values on the main diagonal are true values (classified correctly).
- ROC (Receiver Operating Characteristic) curve: shows the performance of a classification model at all classification thresholds and is created by plotting True Positive Rate (TPR), in terms of False Positive Rate (FPR).
- Area Under ROC Curve (AUC): the area under ROC graph is used as a measure to evaluate the performance of a classifier. The numerical value of AUC is a number between zero and one.

Finally, heat map and Grad-CAM (Gradient-weighted Class Activation Mapping) techniques are used to interpret the outputs or predictions of a proposed model. These techniques actually highlight regions of the input image where the model pays close attention during the classification process. In today's research, interpretation of neural models is one of the most important challenges of diagnosis methods since it gives the experts an assurance that their proposed model is well able to make diagnoses and recognize correct points in the images.

In heat maps, the coldest color (blue) shows parts that have no effect on producing predictions. Parts that have the most influence on producing predictions are colored in yellow or

red. The Grad-CAM technique is itself a CNN and an image is given as input to a CNN network. Extracted feature maps from the last convolution layer are passed through a fully connected layer and the winning neuron of Softmax is determined. More details on this technique are provided in [28].

3- 2- Results of experiments

Results of precision, recall, and F1-score criteria for the seven proposed models are given in Tables 1, 2, 3, 4. The support column represents an overall count of occurrences identified for a specific class type. These tables clearly indicate that the values exceed 98%. Results for the four models, as depicted in Table 4, remain consistent:

- VGG16 with two-layer fine-tuning
- InceptionV3 with 12-layer fine-tuning
- Resnet152v2 with five- and six-layer fine-tuning

Results of accuracy, sensitivity, and F1-score for the second proposed model without K-fold in the third strategy are given in Table 5. As can be seen, the F1-score for this table has an average value of 0.96. As seen from Table 6, for the VGG16 model with the last three reset layers, the F1-score has an average value of about 0.98.

Figures 6, 7, and 8 show accuracy and loss charts for training and validation data of the seven proposed models separately. Figure 6.a illustrates that the VGG16 model with two-layer fine-tuning achieves higher accuracy values in both training and validation phases compared to the same model with one-layer fine-tuning and the model without fine-tuning.

Table 1. Mrasures for VGG16 Without Fine-tune

Classes	precision	recall	f1-score	support
Bacterial blight	0.99	1.00	0.99	314
Blast	1.00	0.99	0.99	293
Brown spot	1.00	1.00	1.00	320
Tungro	1.00	1.00	1.00	262

Table 2. Mrasures for VGG16 With One-layer Fine-tune

Classes	precision	recall	f1-score	support
Bacterial blight	0.99	1.00	1.00	314
Blast	1.00	0.99	0.99	293
Brown spot	1.00	1.00	1.00	320
Tungro	1.00	1.00	1.00	262

Table 3. Mrasures for InceptionV3 With 11-layer Fine-tune.

Classes	precision	recall	f1-score	support
Bacterial blight	0.99	1.00	0.99	313
Blast	0.99	0.98	0.99	294
Brown spot	0.99	0.99	0.99	320
Tungro	1.00	1.00	1.00	262

Table 4. VGG16 With Two-layer, InceptionV3 With 12-Layer, and Resnet152V2 With five and six-laye Fine-tune.

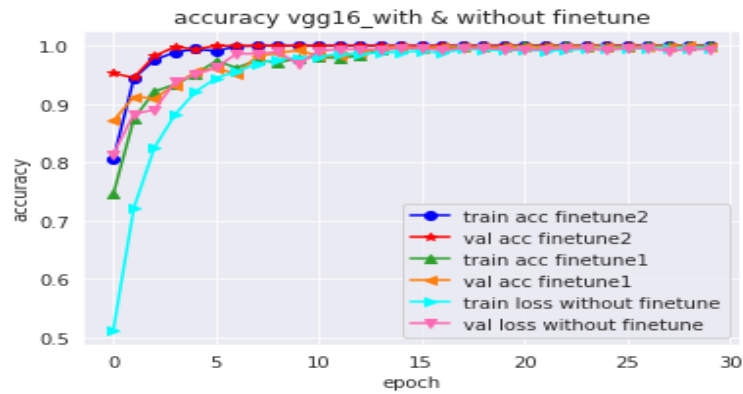
Classes	precision	recall	f1-score	support
Bacterial blight	1.00	1.00	1.00	317
Blast	1.00	1.00	1.00	290
Brown spot	1.00	1.00	1.00	320
Tungro	1.00	1.00	1.00	262

Table 5. Criteria results for the third proposed model without K-fold cross-validation

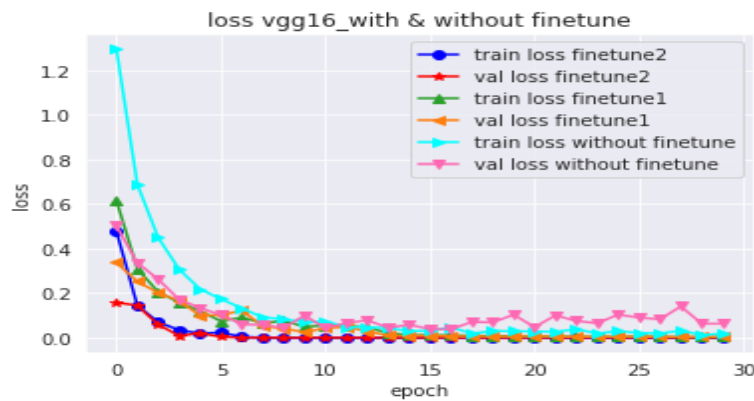
Classes	precision	recall	f1-score	support
Bacterial Leaf Blight (BLB)	0.96	0.96	0.96	28
Brown Plant Hopper (BPH)	0.93	0.82	0.87	17
Brown Spot	1.00	1.00	1.00	23
False Smut	1.00	1.00	1.00	19
Others	0.94	0.94	0.94	47
Hispa	0.87	0.93	0.90	14
Neck Blast	0.98	.100	0.99	57
Sheath Blight and/or Sheath Rot	0.93	0.89	0.91	46
Stemborer	0.95	1.00	0.97	39

Table 6. Criteria results for the third proposed model by rearranging last three layers with VGG16

Classes	precision	recall	f1-score	support
Bacterial Leaf Blight (BLB)	1.00	0.90	0.95	31
Brown Plant Hopper (BPH)	0.93	0.88	0.90	16
Brown Spot	1.00	1.00	1000	23
False Smut	95.0	1.00	0.97	18
Others	1.00	1.00	1.00	47
Hispa	0.87	0.93	0.90	14
Neck Blast	1.00	1.00	1.00	58
Sheath Blight and/or Sheath Rot	95	1.00	0.98	42
Stemborer	1.00	1.00	1.00	41

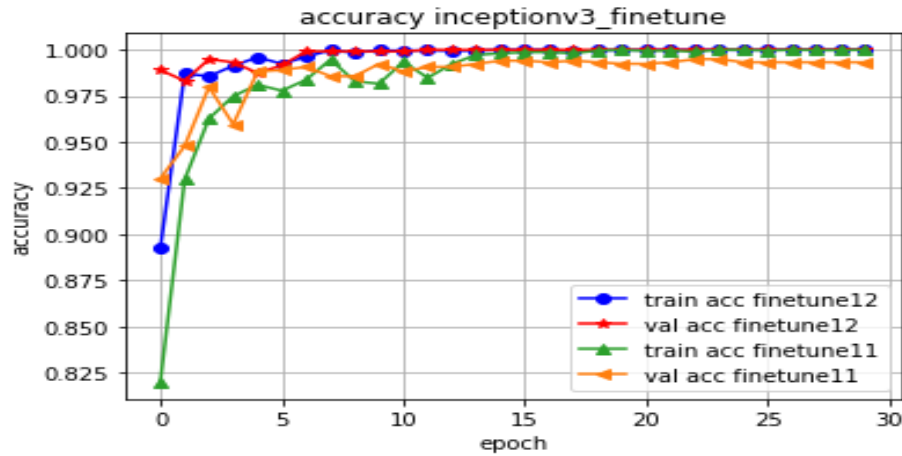


a

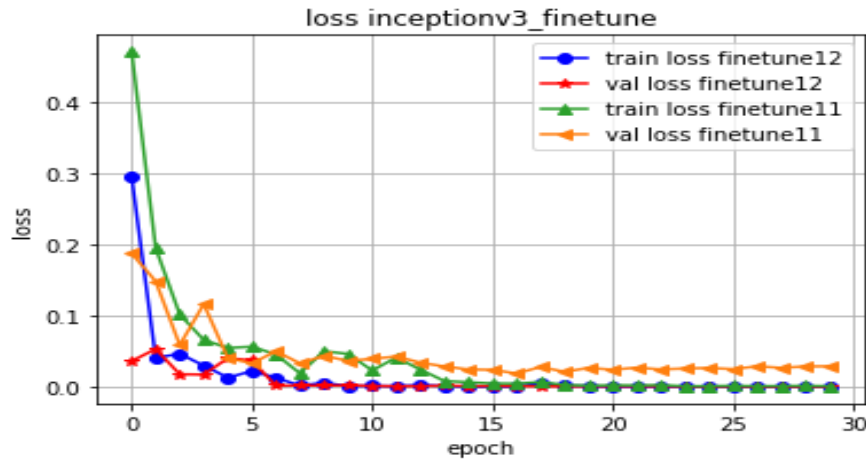


b

Fig. 6. Diagrams of a) accuracy, and b) loss for VGG16 without fine-tune, and one-layer and two-layer fine-tuning.



a



b

Fig. 7. Diagrams of a) accuracy, b) loss for InceptionV3 with 11- and 12-layer fine-tuning.

Additionally, the error chart in Figure 6.b demonstrates that the VGG16 model with two-layer fine-tuning has the lowest error rate for both training and validation phases. Moving on to Figure 7. a, it can be observed that the InceptionV3 model with 12-layer fine-tuning exhibits higher accuracy values for both training and validation phases compared to the results of the same model with 11-layer fine-tuning. Furthermore, the error rate of the InceptionV3 model with 12-layer fine-tuning is lower (Fig. 7.b).

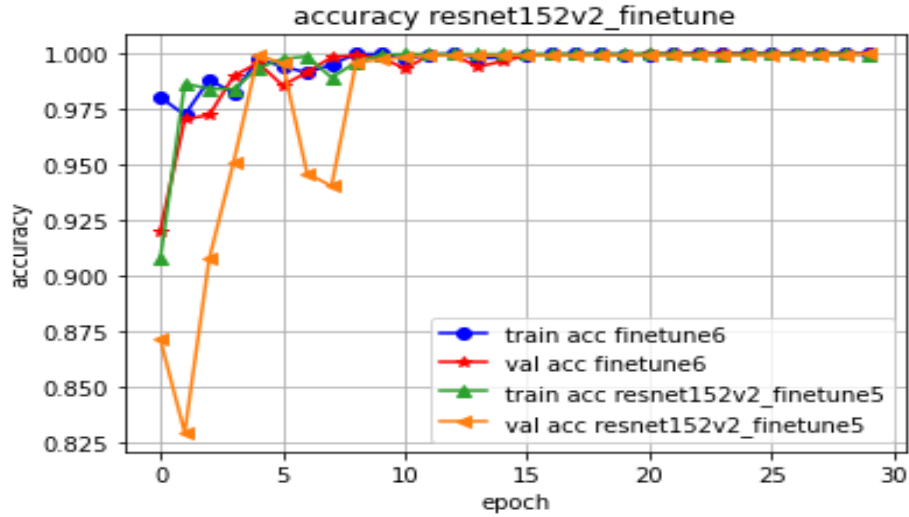
In Fig. 8.a, accuracy values for the Resnet152v2 model in training and validation phases with six-layer fine-tuning fluctuate less than that of the same model with five-layer fine-tuning. In Fig. 9 accuracy and error diagrams for the third proposed model are presented and on the second dataset. As can be seen, the accuracy of this graph for validation data is above 96% and its error is less than 0.2.

Fig. 10 also shows the accuracy and loss of the third

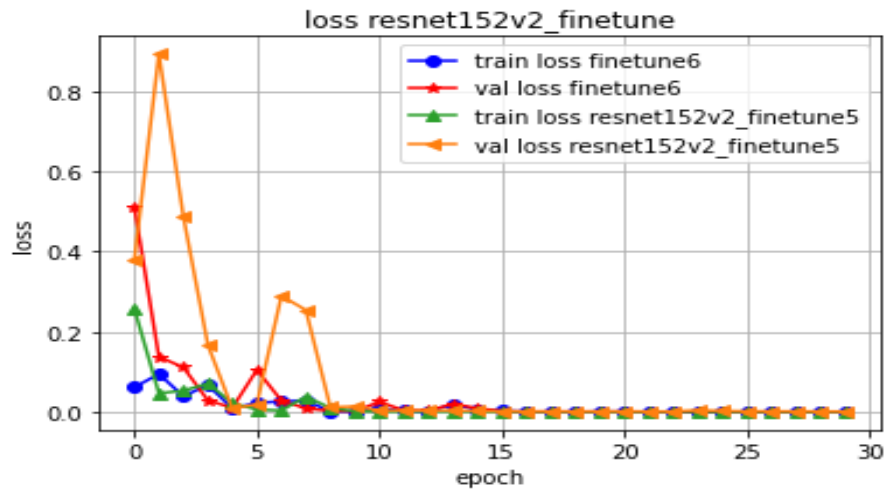
model on the second dataset with three-layer of fine tuning. As shown in this figure, the accuracy of this model is 97.93% and the error is 0.1259 for the second dataset, which is 1.38% different from the accuracy of our previous method without K-fold. This is due to the second dataset which has fewer images compared to ImageNet. In addition, smaller batch sizes and the use of a different SGD optimizer are other reasons for the low accuracy of this model.

Program execution time in this method is 377.1392 seconds, equivalent to approximately 6.2857 minutes. The size of this model is 58.2 MB and the accuracy convergence score in this model is 0.97.

The confusion matrices for all seven models are displayed in Figure 11. It is evident from this figure that VGG16, without fine-tuning, misclassified four bacterial blight samples as blast samples (Figure 11.a). Additionally, it achieved an accuracy of 99.66% and an error rate of 0.0764% as stated



a



b

Fig. 8. a) Accuracy, b) loss for Resnet152v2 with five- and six-layer fine-tuning.

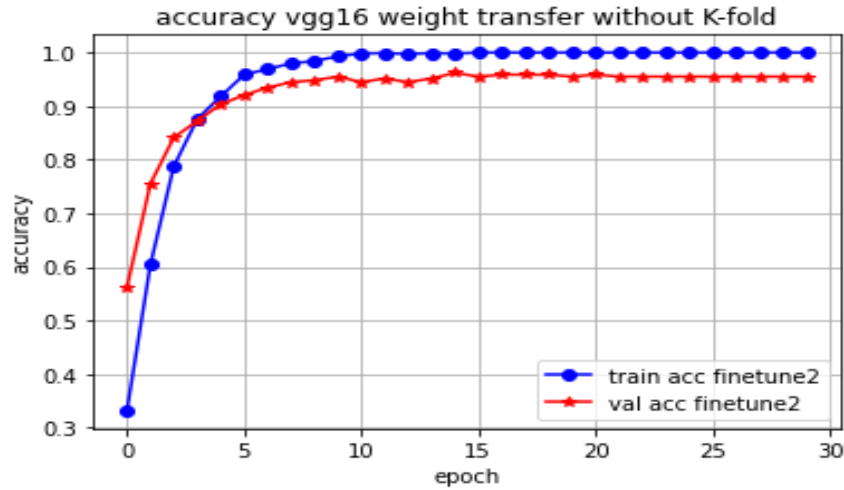
in Table 9.

Figure 12 shows confusion matrices for the third proposed model. It can be seen from this figure that without k-fold for the second dataset, 13 cases are misdiagnosed. In Fig. 12.b, the model has been evaluated for the second dataset and, as in its base paper, using the last three-layer fine-tuning method with VGG16, all data have been used for testing. It can be seen that this model misdiagnosed six cases. The model presented in Fig. 12.c with 10-fold diagnosed correctly all cases.

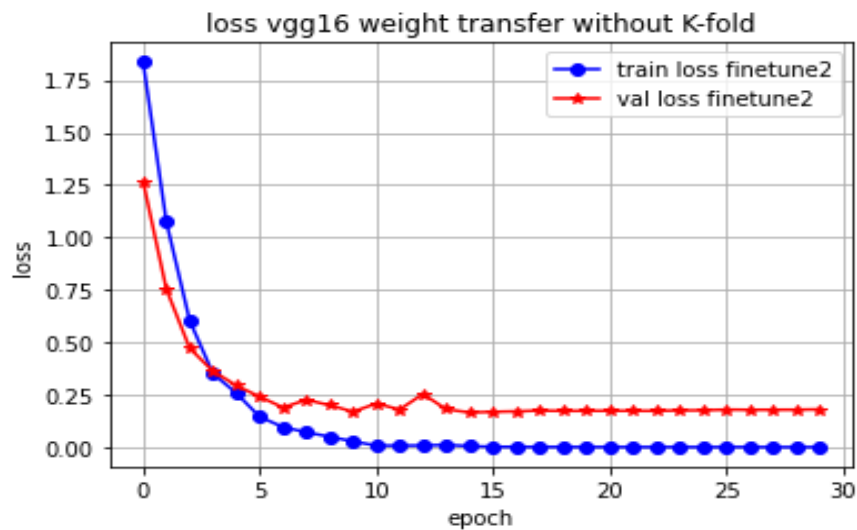
In Fig. 13 and Fig. 14 ROC curves are displayed for the first and second datasets, respectively. As seen, these

diagrams for all categories are close to one and AUC value for all of them has been calculated, which is equal to the area under the ROC curve.

Finally, heat map and Grad-CAM results are presented in this section, to represent the last displayable layer of the proposed models. As seen from Table 7, for the second dataset, in the first column on the right, the main image and in the next column the heat map corresponding to that image are displayed. Yellow or light areas are the diagnostic area of that layer, and in the next column, using Grad CAM, those areas tending to red in the original image are shown. In the



a



b

Fig. 9. Diagrams of a) accuracy, and b) loss for VGG16 without k-fold on Rahman et al dataset.

next column, the name of the category to which this leaf really belongs, and in the last column name of the category recognized by our model is written.

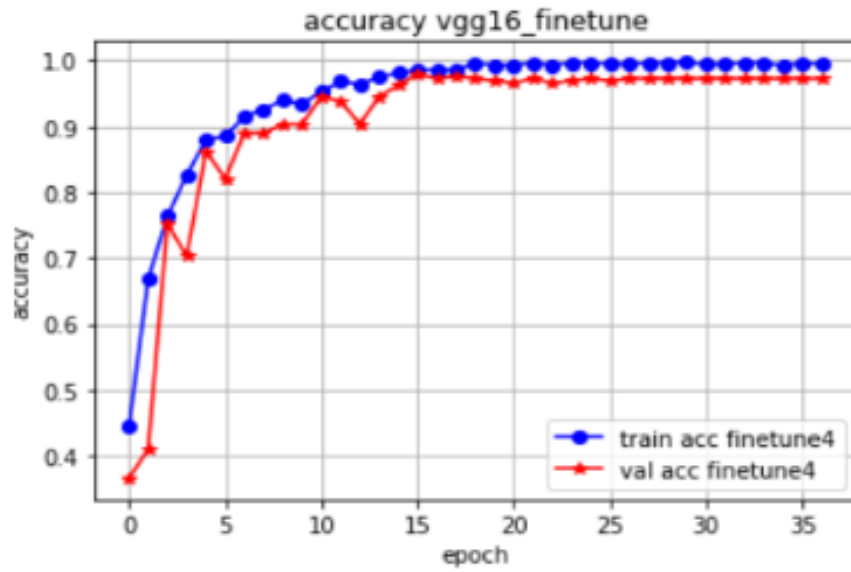
As seen from Table 7, the fifth row is an example of model misdiagnosis. The same information of heat map and Grad-CAM methods is shown in Table 8 for the first dataset.

Table 9 presents a comparison of our proposed models in this paper with other studies. According to this table, VGG-16 without fine-tuning and with one-layer fine-tuning has incorrectly classified four and three samples, respectively, of bacterial blight (Fig. 11.a and Fig.11.b) and reached an

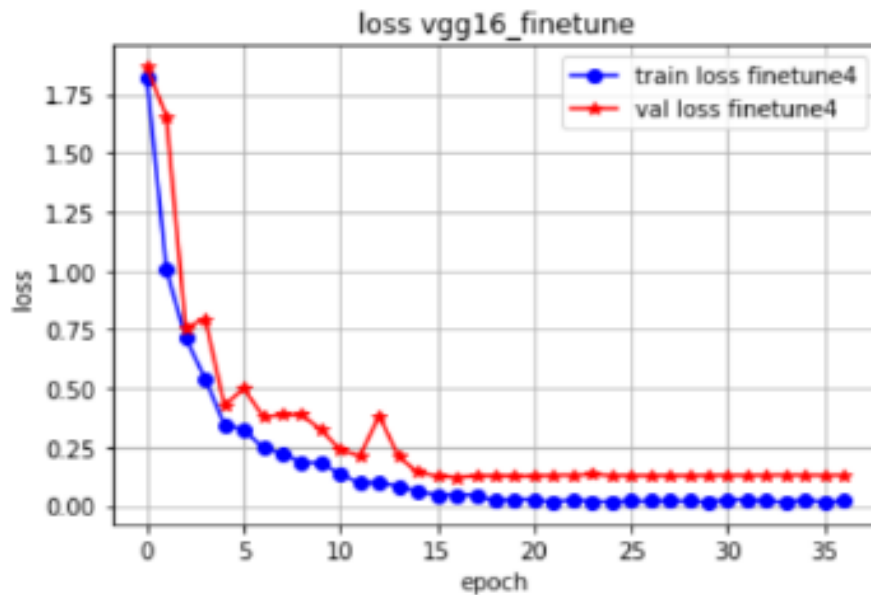
accuracy of 99.66% and 99.71%, respectively. Parameters in Table 9 were acquired through the summary function in the Tensorflow library. Accuracy, loss, parameters, and simulation times for [1], [2], [16], and [24] are also displayed in Table 9 for comparison purposes.

In [24], the same dataset was utilized as in this study, resulting in higher accuracy. Moreover, the shortest training time among the suggested models belongs to VGG16 without fine-tuning, which lasts around one minute. This is approximately six seconds less than that documented in [24].

The Inceptionv3 model with 11-layer fine-tuning has



a



b

Fig. 10. a) Diagrams of a) accuracy, and b) loss for VGG16 with three-layer fine tuning on Rahman et al dataset.

incorrectly classified eight samples (Fig. 11.c) and reached an accuracy of 99.22% and an error of 0.0333%. The rest of the models, including VGG16 with two-layer fine-tuning, InceptionV3 with 12-layer fine-tuning, and Resnet152v2 with five and six-layer fine-tuning have all reached 100% accuracy and have identified all samples correctly.

In Table 10, results obtained from the three proposed models for the second dataset are given along with results of

[17] that are available for the second dataset. As can be seen, among the three proposed methods, VGG16 using 10_Fold from the first to second dataset has higher average accuracy than other proposed methods (according to Fig. 9 and Fig. 10). Alternatively, the number of misdiagnosed cases of the VGG16 method with fine-tuning last three layers is less (according to Fig. 12). In general, we have reached a higher accuracy than [17] and have had more cases of correct diagnosis.

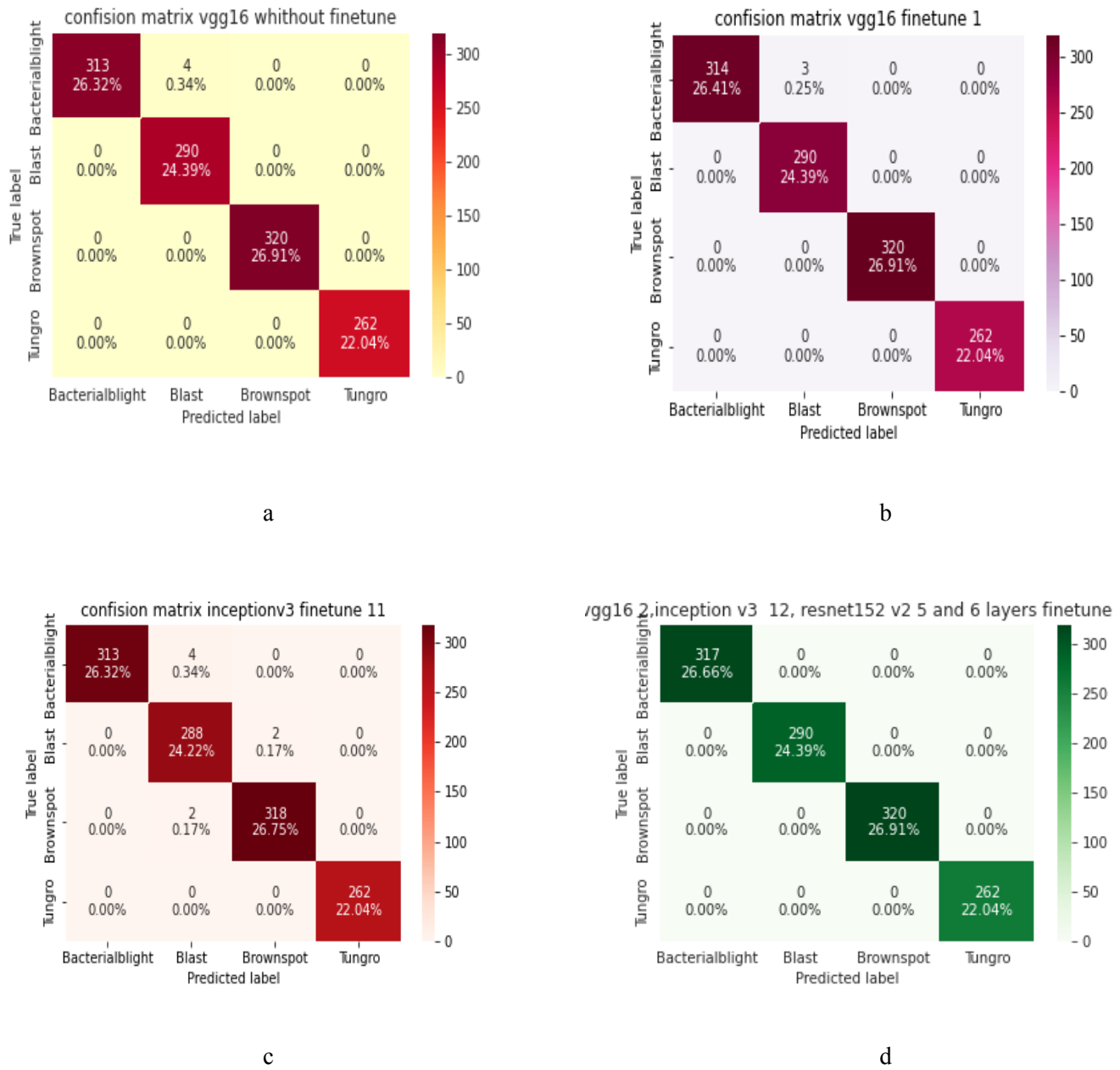


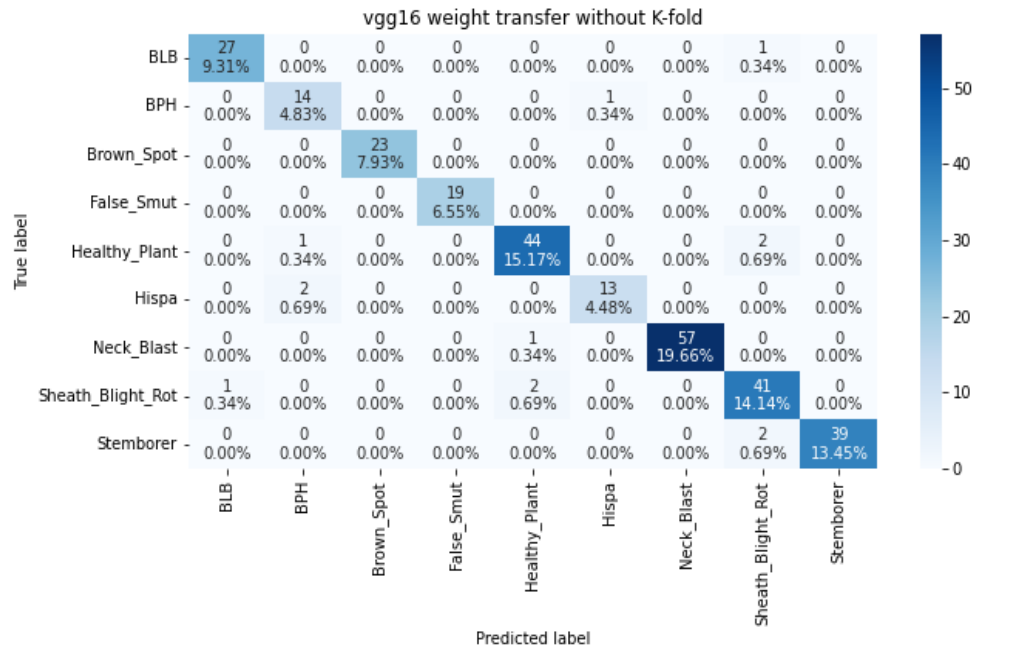
Fig. 11. Confusion matrices for a) VGG16 without fine-tuning, b) VGG16 with one-layer fine-tuning, c) InceptionV3 with 11-layer fine-tuning, d) VGG16 with two-layer, InceptionV3 with 12-layers and Resnet152v2 with five- and six-layers fine-tuning.

4- Conclusion and Directions for Further Study

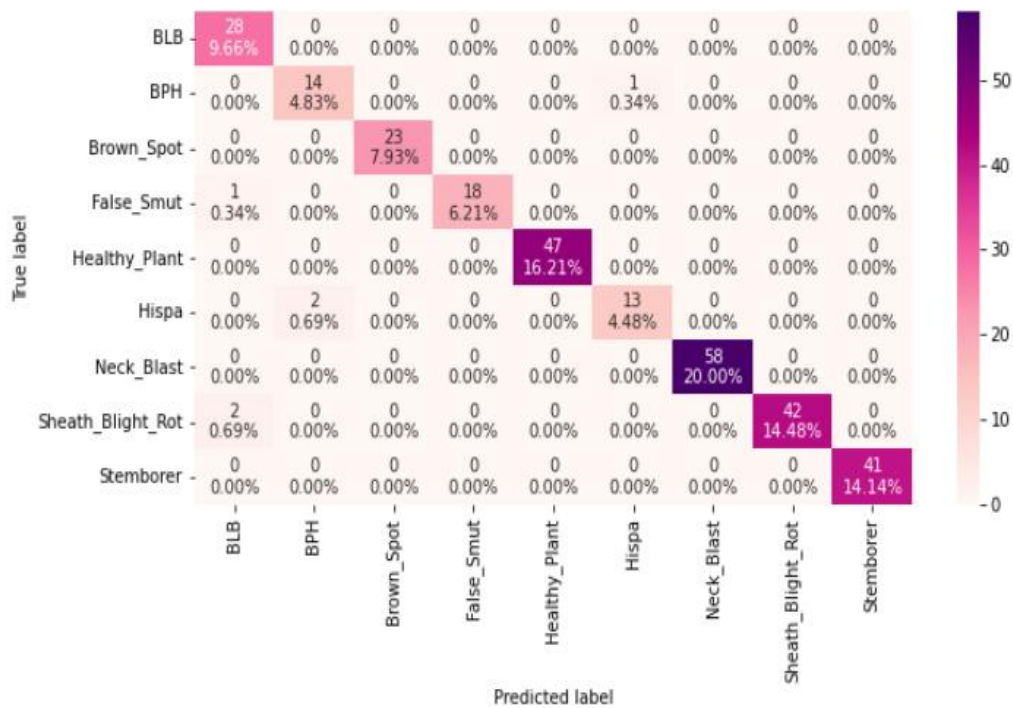
In this study, we investigated the effectiveness of seven CNN models based on pre-trained architectures for classifying rice leaf diseases. From the suggested models, four of them:

- VGG16 with two-layer fine-tuning
 - InceptionV3 with 12-layer fine-tuning
 - Resnet152v2 with five and six-layer fine-tuning
- from the second strategy reached 100% accuracy and F1-

score. In contrast, VGG16 achieved an accuracy of 99.66% and an F1-score of 99.5% without fine-tuning. By adding a single layer of fine-tuning, accuracy is improved to 99.71% and F1-score increased to 99.75%. InceptionV3 achieved an accuracy of 99.22% and an F1-score of 99.25% with 11 layers of fine-tuning. Consequently, out of all the aforementioned models, VGG16 with two layers of fine-tuning demonstrated the best performance and for fine-tuning, boasting a perfect



a



b

Fig. 12. Confusion matrices for a) VGG16 without K_Fold, b) three-layer fine-tuning, and c) with 10_Fold on Rahman et al dataset.(Continued)

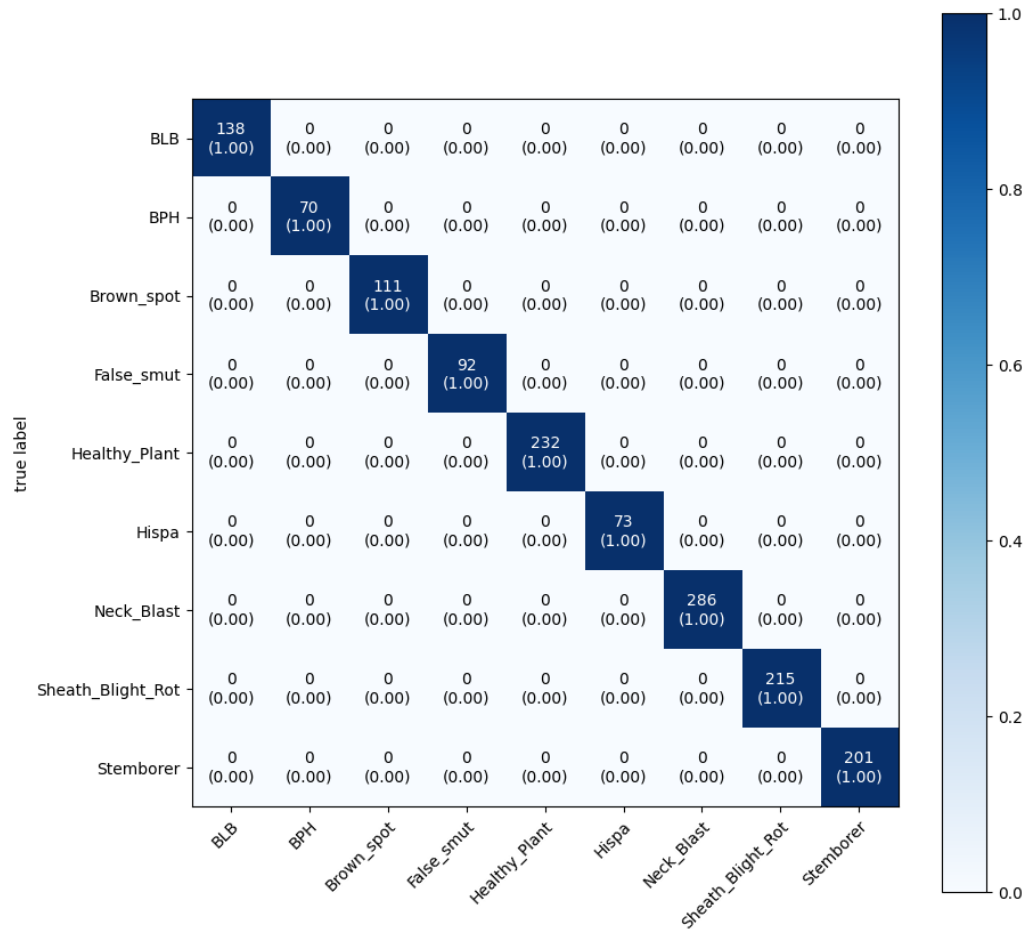


Fig. 12. Confusion matrices for a) VGG16 without K_Fold, b) three-layer fine-tuning, and c) with 10_Fold on Rahman et al dataset.

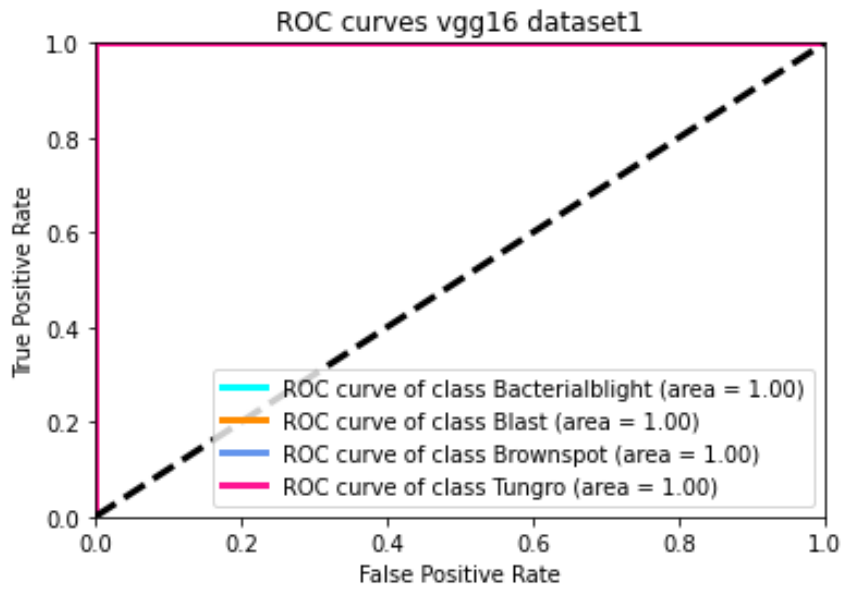


Fig. 13. ROC curve for the second approach. VGG16 model with last two layers of fine tuning.

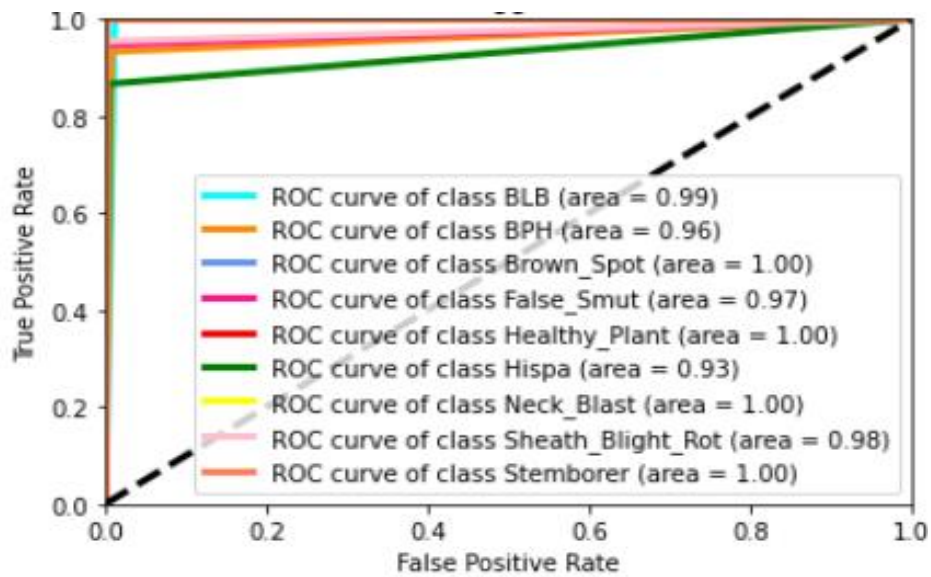


Fig. 14. ROC plot and AUC value

Table 7. Thermal map and Grad-CAM results for images from different categories of dataset Rahman et al.

Class detected	Real Class	Grad Cam	Heat map	Original image
Bacterial leaf blight	Bacterial leaf blight			
Brown Plant Hopper	Brown Plant Hopper			
brown spot	brown spot			
False Smut	False Smut			
Sheath Blight and/or Sheath Rot	Others			
Hispa	Hispa			
Neck Blast	Neck Blast			
Sheath Blight and/or Sheath Rot	Sheath Blight and/or Sheath Rot			
Stemborer	Stemborer			

Table 8. Thermal map and Grad-CAM results for images from different categories of the first dataset.




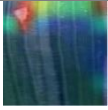





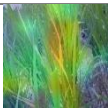
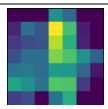

Class detected	Real Class	Grad CAM	Heat map	Original image
Bacterial leaf blight	Bacterial leaf blight			
blast	blast			
brown spot	brown spot			
Tungro	Tungro			

Table 9. Comparison of accuracy, loss, number of trainable and non-trainable parameters, and total number of parameters for all seven proposed models with other methods.

model	Accuracy (%)	loss	Trainable parameters	Non-trainable parameters	Total parameters	Training Time (second)
VGG16 without fine-tune	99.66	0.0318	37,782,852	14,714,688	52,497,540	63.3226
VGG16 fine tune(1)	99.71	0.0053	527,364	14,714,688	15,242,052	893.2215
VGG16 fine tune(2)	100.00	2.0229 e-05	2,887,172	12,354,880	15,242,052	952.0729
InceptionV3 fine tune(12)	100.00	0.0003	1,707,524	21,409,056	23,116,580	821.3197
InceptionV3 fine tune(11)	99.22	0.0333	1,314,308	21,802,272	23,116,580	802.7838
Resnet152v2 fine tune(6)	100.00	0.0005	2,369,540	57,275,904	59,645,444	2873.9854
Resnet152v2 fine tune(5)	100.00	0.0009	2,368,516	57,276,928	59,645,444	2746.6022
ResNet50 (SVM using deep features) [24]	98.38	-	-	-	-	69.04307
Reattention mechanism and convolution blocks and Dense blocks [25]	99.60	-	-	-	-	-
Coordinate Attention, Inception-iv and Reduction-iv modules [26]	95.57	-	-	-	-	-

Table 10. Comparison of accuracy, loss.

Model	Mean Accuracy	Standard Deviation
VGG16 with three-layer fine-tuning, without k-fold	97.93% [Accuracy]	0.1259 [loss]
Without k-fold	96.55% [Accuracy]	0.1353 [loss]
Accuracy with 10-fold	98.57%	(4.29%~+)
Accuracy of [17] with VGG16	97.12%	(2.23%~+)

accuracy and F1-score, as well as a smaller number of layers.

After mentioning the best model, another dataset was chosen with more classes (nine classes) to test that model. This model based on VGG16 using 10-fold from the first to second dataset has the highest average accuracy.

For future studies, we plan to evaluate the enhanced model presented in this paper using a broader dataset encompassing various types of leaves. This model can further be applied to larger datasets containing a wider range of classes to improve disease detection capabilities.

5- Acknowledgment

This work is supported by the Shahid Chamran University of Ahvaz under grant number SCU.EC.1401.205.

References

- [1] V. K. Shrivastava, M. K. Pradhan and M. P. Thakur, «Application of Pre-Trained Deep Convolutional Neural Networks for Rice Plant Disease Classification,” 2021 Int. Conf. on Artificial Intelligence and Smart Systems (ICAIS), pp. 1023-1030.
- [2] S. Ghosal and K. Sarkar, “Rice Leaf Diseases Classification Using CNN With Transfer Learning,” 2020 IEEE Calcutta Conference (CALCON), pp. 230-236.
- [3] H. Andrianto, Suhardi, A. Faizal and F. Armandika, “Smartphone Application for Deep Learning-Based Rice Plant Disease Detection,” 2020 Int. Conf. on Information Technology Systems and Innovation (ICITSI), pp. 387-392.
- [4] P. Mekha and N. Teeyasuksaet, “Image Classification of Rice Leaf Diseases Using Random Forest Algorithm,” 2021 Joint Int. Conf. on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, pp. 165-169.
- [5] M. E. Pothan and M. L. Pai, “Detection of Rice Leaf Diseases Using Image Processing,” 2020 Fourth Int. Conf. on Computing Methodologies and Communication (ICCMC), pp. 424-430.
- [6] T. Kodama and Y. Hata, “Development of Classification System of Rice Disease Using Artificial Intelligence,” 2018 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC), pp. 3699-3702.
- [7] S. Ramesh and D. Vydeki, “Application of machine learning in detection of blast disease in South Indian rice crops,” J. Phytol., vol. 11, pp. 31–37, 2019.
- [8] S. Phadikar and J. Goswami, “Vegetation indices based segmentation for automatic classification of brown spot and blast diseases of rice,” 2016 3rd Int. Conf. on Recent Advances in Information Technology (RAIT), 2016, pp. 284-289.
- [9] T. Islam, M. Sah, S. Baral and R. Roy Choudhury, “A Faster Technique on Rice Disease Detection using Image Processing of Affected Area in Agro-Field,” 2018 Second Int. Conf. on Inventive Communication and Computational Technologies (ICICCT), pp. 62-66.
- [10] M. Hasan Matin, A. Khatun, Md. G. Moazzam and M. Shorif Uddin, “An Efficient Disease Detection Technique of Rice Leaf Using AlexNet,” 2020 8th J. of Computer and Communications, pp. 49-57.
- [11] G. Verma, C. Taluja and A. K. Saxena, “Vision Based Detection and Classification of Disease on Rice Crops Using Convolutional Neural Network,” 2019 Int. Conf. on Cutting-edge Technologies in Engineering (ICon-CuTE), pp. 1-4.
- [12] J. Hasan, S. Mahub, S. Alom and A. Nasim, “Rice Disease Identification and Classification by Integrating Support Vector Machine With Deep Convolutional Neural Network,” 2019 1st Int. Conf. on Advances in Science, Engineering and Robotics Technology (ICASERT), pp. 1-6.
- [13] S. A. Burhan, S. Minhas, A. Tariq and M. Nabeel Hassan, “Comparative Study Of Deep Learning Algorithms For Disease And Pest Detection In Rice Crops,” 2020 12th Int. Conf. on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1-5.
- [14] R. Sharma, S. Das, M. K. Gourisaria, S. S. Rautaray, and M. Pandey, “A Model for Prediction of Paddy

- Crop Disease Using CNN,” in Progress in Computing, Analytics and Networking, 2020, pp. 533–543.
- [15] R. J. Bharathi, “Paddy Plant Disease Identification and Classification of Image Using AlexNet Model,” Int. J. Analytical and Experimental Modal Analysis, vol. XII, no. III, pp. 1094–1098, 2020.
- [16] J. Chen, D. Zhang, Y. A. Nanekaran, D. Li, “Detection of rice plant diseases based on deep transfer learning,” J. of the Science of Food and Agriculture, vol. 100, no. 7, pp. 3246– 3256, 2020.
- [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, 2020.
- [18] R. Wadhawan, M. Garg and A. K. Sahani, “Rice Plant Leaf Disease Detection and Severity Estimation,” 2020 IEEE 15th Int. Conf. on Industrial and Information Systems (ICIIS), 2020, pp. 455-459.
- [19] R. Sharma, V. Kukreja and V. Kadyan, “Hispa Rice Disease Classification using Convolutional Neural Network,” 2021 3rd Int. Conf. on Signal Processing and Communication (ICPSC), 2021, pp. 377-381.
- [20] T. M. S. Sazzad, A. Anwar, M. Hasan and M. I. Hossain, “An Image Processing Framework To Identify Rice Blast,” 2020 Int. Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1-5.
- [21] S. S. Chawathe, “Rice Disease Detection by Image Analysis,” 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0524-0530.
- [22] B. S. Ghyar and G. K. Birajdar, “Computer vision based approach to detect rice leaf diseases using texture and color descriptors,” 2017 Int. Conf. on Inventive Computing and Informatics (ICICI), pp. 1074-1078.
- [23] T. Shrivastava, M.S. Pillai, B. Baranidharan, “Rice Disease Classification using Deep Convolutional Neural Network”, February 2020, Int. J. of Innovative Technology and Exploring Engineering (IJITEE), 9, 2278-3075.
- [24] P. Sethy, N. Barpanda, A. Rath and S. Behera, “Deep feature based rice leaf disease identification using support vector machine,” 2020, Computers and Electronics in Agriculture, 175, 105527.
- [25] J. Peng, Y. Wang, P. Jiang , R. Zhang and H. Chen, “c Backgrounds Using a Res-Attention Mechanism,” Applied Sciences, vol. 19, 2023.
- [26] N. Zeng , G. Gong, G. Zhou and a. C. Hu, “An Accurate Classification of Rice Diseases Based on ICAI-V4,” Plants, vol. 28, 2023.
- [27] <https://keras.io/api/applications/>
- [28] R. R., Selvaraju, et al. “Grad-cam: Visual explanations from deep networks via gradient-based localization,” Proceedings of the IEEE international conference on computer vision, 2017.

HOW TO CITE THIS ARTICLE

M. Mavaddat, M. Naderan, S. E. Alavi. *Plant Diseases Classification Using Pre-trained and Transfer Learning Models: A Study on Rice Leaves*. AUT J. Elec. Eng., 57(2) (Special Issue) (2025) 263-282.

DOI: [10.22060/ej.2024.23015.5580](https://doi.org/10.22060/ej.2024.23015.5580)



