An Efficient Convolutional Neural Network Based Deep Learning Framework for Rice Leaf Disease Classification

Jaspreet Singh [1], Er Karandeep Singh [2]
[1] Department of Computer Science & Engineering, Punjabi University, Patiala, Punjab, India
[2] Assistant Professor, Department of Computer Science & Engineering, Punjabi University, Patiala, Punjab, India

* Corresponding Author: **Jaspreet Singh**

Article Info

P-ISSN: 3051-3421
E-ISSN: 3051-343X
Volume: 07
Issue: 02
Received: 21-04-2026
Accepted: 23-05-2026
Published: 25-06-2026
Page No: 01-08

Abstract
Rice is one of the most popular and extensively grown crops. More than fifty percent of the global population consumes it as a staple food. However, a number of diseases affect the quality of crop which also affect the total production of the yield. Identification of such disorders is important for crop management. Traditional disease detection methods are time consuming and susceptible to human error because they rely on personal evaluation by professionals.
In our work, we have built a task specific system. This system uses a convolutional neural network based framework for detection and classification of rice leaf diseases by looking at leaf pictures. Two datasets are used in this work. The rice leaf disease dataset contains 16000 images which are distributed into four classes with 4000 images in each class and additionally 139 images of brown spot class from A dataset of rice leaf disease, to add variation in the dataset, is used. Duplicate images were cleaned and removed. Images were preprocessed. For fair evaluation, the dataset was divided into training set (70%), validation set (15%), and testing set (15%). With 25 epochs, by employing Adam optimizer, model was trained. It is designed to extract visual features using four convolutional layers, five batch normalization layers, four pooling layers and two fully connected layers.
To see how well trained model is performed, it is evaluated against some transfer learning models like MobileNetV2, ResNet50 and Vgg16 on the same dataset. Performance evaluation using accuracy, model size, time per image, and number of parameters showed that the proposed model achieved high accuracy and provided better discrimination between visually similar disease classes. Proposed model achieves 99.33% accuracy, has model size of 1.618 MB, 9.512 ms time per image, 424260 parameters. In contrast MobileNetV2 achieved 97.17% accuracy, has model size of 8.633 MB, 14.141 ms time per image and 2263108 parameters. ResNet50 achieved 44.93% accuracy, has model size of 90.01 MB, 64.661 ms time per image and 23595908 parameters. Vgg16 achieved 79.75% accuracy, has model size of 56.13 MB, 193.830 ms time per image and 14716740 parameters. Proposed model has performed well and doesn't need numerous parameters, this shows that for practical implementation, it can be appropriate.

DOI: https://doi.org/10.54660/JADR.2026.7.2.01-08

Keywords: deep learning, rice leaf disease, convolutional neural network

1. Introduction
Rice represents one of the most important staple crops. It is a critical crop that provides food for billions of people. Moreover, rice cultivation seems important especially in Asian countries, where It plays a key part in food security and agricultural economy. Majority of Indians preferred rice as an staple diet [25]. Rice crop growth depends on numerous components. Plant disease is one of the major important biological components that lowers crop production [15].

Here are some of the diseases of rice leaf:

Bacterial leaf blight: Bacterium Xanthomonas oryzae is responsible for causing this disease**.** It attacks rice plant leaves and spreads rapidly under favorable environment conditions. It is characterized by elongated yellow or pale lesions along leaf edges.

Brown spot: It is caused by Bipolaris oryzae. It looks like as small circular or oval shaped brown lesions on the leaf surface.

Leaf smut: It is a fungal disease. It is caused by Entyloma oryzae. It appears as a narrow black or dark brown streaks on the surface of rice leaves.

These diseases can be detected through visible symptoms on plants leaves. If these diseases are not detected, then they may spread and can affect production. Traditionally, specialists used to detect disease by visual inspection. Those techniques were effective but they had some limitations. They required lots of time and effort. In remote or rural regions, it may not be accessible to farmers. As AI and deep learning technologies have emerged, agricultural practitioners can use image-based analysis tools that can be helpful in earlier disease detection. Thus, deep learning methodologies, especially CNNs, showed superb achievement in visual pattern recognition tasks. From raw images, CNNs can identify relevant image features by themselves and eliminate the need for manual feature extraction. This can improve classification accuracy. Several studies used well known deep learning models like CNN, MobileNetV2, ResNet50 VGG16. Although they performed well but they had some limitations like they require millions of parameters which make them computationally expensive and they are difficult to deploy. Therefore, there is need of a lightweight or efficient model which can classify or identify rice leaf disease accurately. Which also maintains low computational complexity.

Motivated by this need, in this work, a custom CNN-based model is proposed that could classify rice leaf diseases using leaf pictures. Commonly used performance metrics are used to measure or analyze the model and also compared it with transfer learning methods. The statistics suggest that the proposed model holds up on accuracy, making it appropriate for practical implementation in the field.

2. Major contributions of this study

Suggested model shows high classification accuracy and performs better than transfer learning models like MobileNetV2, ResNet50 and VGG16 on the given dataset. It has lightweight architecture, which has only about 424K parameters. It reduces memory usage which makes it appropriate for implementing on gadgets that have confined computational resources. The model also provides faster inference time (~10 ms per image). The proposed framework is a workable and scalable solution for actual agricultural applications since it generally accomplished a fair balance among accuracy, efficiency and complexity.

3. Related work

By adopting approaches of machine learning and deep learning, in past few years, several research studies have examined automated diagnosis of plant disease. Along with traditional classifiers, earlier methods used handcrafted features. While these research achieved success, their performance was strongly depended on feature design. Plant disease classification accuracy has been improved by deep learning models like CNNs. On several crop datasets, researchers applied various models like AlexNet, VGGNet, Inception, and ResNet. Transfer learning models have achieved high accuracy, but they have very high computation cost which makes the not well suitable for deployment in resource-constrained environments. This study differs by developing a custom CNN architecture which achieves competitive accuracy and also maintains lower computational complexity.

For plant disease classification. several studies have analyzed the employment of prelearnt deep learning frameworks like AlexNet, VGGNet, GoogLeNet, ResNet, and DenseNet through transfer learning. These models showed excellent performance, but usually they required a large number of parameters and require high computational resources which makes them less fit for utilization on budget friendly or mobile agricultural appliance [7].

Kurmi *et al*. projected a deep CNN based schema which combined region-of-interest localization with CNN-based feature extraction for crop leaf disease detection to address computational efficiency and robustness. It removed redundant background information from leaf images, improved classification accuracy and generalization. The framework was examined on the PlantVillage dataset. It achieved higher performance compared to conventional methods of image processing [7].

Priyadharshini *et al*. achieved 97.89% accuracy with 80:20 train-test split by modifying LeNet architecture for identifying diseases in maze leaf [13]. Jasrotia *et al*. achieved 96.76% accuracy for four maize disease classes by combining CLAHE preprocessing with custom CNN [22].

Upadhyay and Kumar used CNN with Otsu's thresholding and achieved 99.7% accuracy for rice disease classificatio006E [15].

For apple disease classification by developing a custom CNN Vishnoi *et al*. achieved 98% accuracy [9].

Despite significant progress, many models performed well on controlled datasets but shown reduced accuracy on field-acquired images. This shows the need for lightweight yet accurate CNN architectures which can generalize well across diverse crop species and environmental conditions.

For automated crop leaf disease classification, the reviewed literature shows that CNN based models can offer an effective and scalable solution. However, further study is necessary to develop optimized CNN architectures that balance accuracy, computational efficiency, and real-world applicability. This forms the inspiration for the proposed work, which attempts to design a CNN framework for accurate identification of diseases in crop leaf images.

4. Description of The Dataset

In the current work, rice-leaf dataset [26] and A dataset of rice leaf diseases [27] are used. In Rice leaf dataset, it consists of 16000 images sorted into four different categories or classes (each class contains 4000 images): different diseases classifications (bacterial leaf blight, brown spot and leaf smut) and healthy leaves. From a dataset of rice leaf disease only brown spot class’s images (139 images) are used to add variation in the dataset. These were collected from kaggle. Duplicate images are removed and dataset had total 16058 images.

Table 1: Number of images per class

Class name	Number of images
Healthy	4000
Bacterial leaf blight	4084
Brown spot	4009
Leaf smut	3965

Every image is sorted to a respective class and the dataset was divided into training set, test set andvalidation set (70%: 15%: 15% respectively).

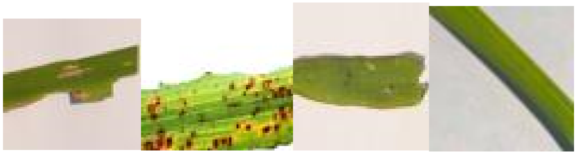

Fig 1: Sample rice leaf images from different disease categories used in this study [26, 27]: (a) Bacterial leaf blight (b) Brown spot (c) Leaf smut (d) healthy

4.1. Dataset statistics

Dataset contains four classes. For every class, dataset in independently divided as shown in table II.

Table 2: Class distribution

Dataset split	Total sample images	Objective
Training set	11238	Model learning
Validation set	2409	Hyper parameter tuning
Test set	2411	Final evaluation

Preprocessing leads training, images are rescaled to 224×224 pixel dimensions and pixel value is normalized to ensure compatibility with deep learning techniques. Thus, the training set experience data augmentation techniques like rotation, zooming and allows model to learn meaningful feature representations from images. During training, to optimize model hyper parameters and detect overfitting validation set is employed. Test set is utilized to evaluate the model. It contains images that are not seen by the model before.

5. Methodology

5.1. Framework Used

The proposed model is implemented using Jupyter Notebook, open-source web-based execution platform. Tensorflow and Keras are utilized to construct and train the model. Code execution, documentation, and visualization are all made easier using Jupyter.

5.2. Image Preprocessing and Data Augmentation

For improving model performance, image preprocessing plays an important role. Images are rescaled to 224×224 pixel dimensions and value of pixels is normalized to range 0-1 through dividing by 255 to ensure compatibility with deep learning techniques.

$$x_{norm} = \frac{x}{255}$$

In the dataset to include variability, data augmentation is used which avoids overfitting. The augmentation techniques used in this study include rotation, flipping, zooming and contrast adjustment. So that it learns better.

6. Proposed Model Architecture

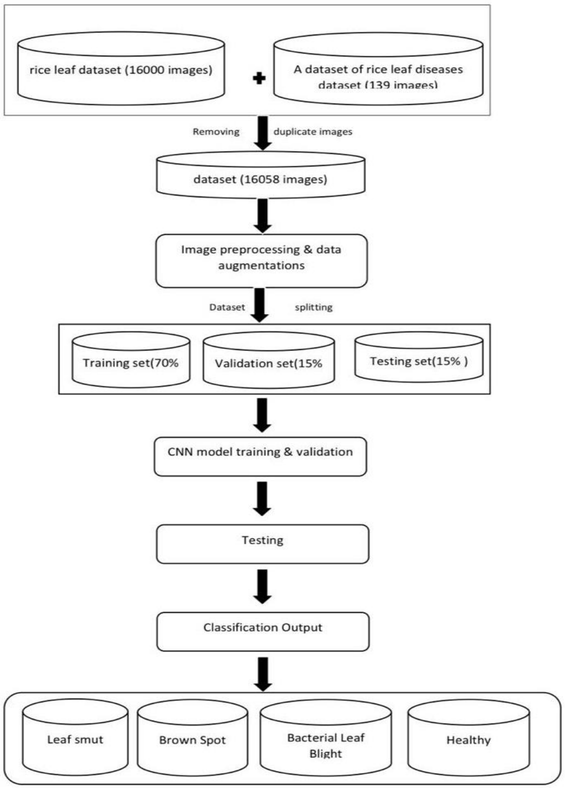

Fig 2: Proposed methodology

In this study, for automatic rice leaf disease categorization, a custom convolutional neural network (CNN) is developed. From leaf images the model extracts hierarchical features while maintaining relatively low computational complexity. The input images which it processes are of size 224x224x3. It includes four convolutional layers, five batch normalization layers, four pooling layers and two fully connected layers.

Before performing final categorization, the network progressively learns increasingly abstract representation of the input image. It contains approximately 424,262 parameters which make it lighter than deeper architectures such as VGG16 and ResNet50.

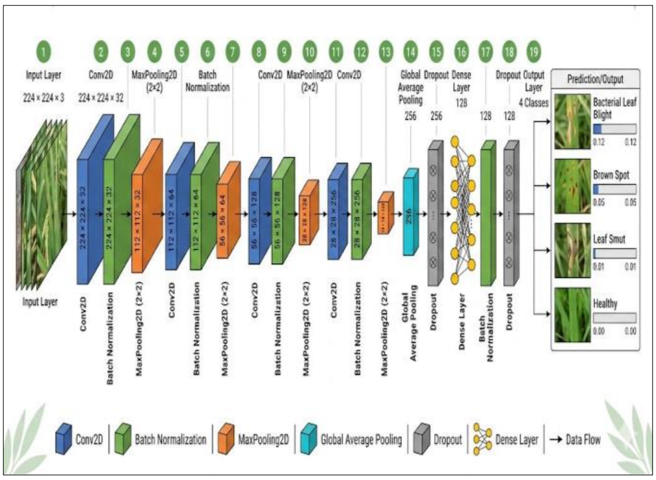

Fig 3: Proposed model's fundamental architecture

Table 3: Proposed framework's Layer wise architecture

Sr. No.	*Layer*	*Output Size*
1.	Input layer	224×224×3
2.	Conv2D	224×224×32
3.	Batch normalization	224×224×32
4.	Maxpooling2D	112×112×32
5.	Conv2D	112×112×64
6.	Batch normalization	112×112×64
7.	Maxpooling2D	56×56×64
8.	Conv2D	56×56×128
9.	Batch normalization	56×56×128
10.	Maxpooling2D	28×28×128
11.	Conv2D	28×28×256
12.	Batch normalization	28×28×256
13.	Maxpooling2D	14×14×256
14.	Global average pooling	256
15.	Dropout	256
16.	Dense	128
17.	Batch normalization	128
18.	Dropout	128
19.	Output layer	4

6.1. Input layer

Here RGB images are accepted. Images are scaled to 224 × 224 pixels. Each image is represented as a 3-D tensor (224×224×3). The image height and width is 224. Here 3 represents RGB color channels.
Prior to feature extraction, pixel values are normalized using a rescaling layer. The input values i.e. [0,255] are rescaled to [0,1].

6.2. Convolutional layers

In convolutional neural networks, convolutional layer is an essential part. Spatial feature from input image are extracted by these layers. They do this by using trainable filters. These filters scan across the input image's spatial dimensions (height and width). An element wise multiplication and summation is performed by each filter and creates a feature map. Patterns like edges, textures and disease lesions are recognized by it.
The convolution operation:

$$y(i,j) = \sum_m \sum_n x(i+m, j+n).(m,n) + b$$

here:
x = input feature map, w = convolution kernel, b = bias term, y(i,j) = resulting feature map
There are four convolution blocks are in this architecture with progressively more filters as in table IV.

Table 4: Architecture of convolution blocks

Layer	Filters	Output size
Conv layer 1	32	224 × 224 × 32
Conv layer 2	64	112 × 112 × 64
Conv layer 3	128	56 × 56 × 128
Conv layer 4	256	28 × 28 × 256

The output size of each convolutional layer is represented in the form of height × width × number of feature maps. The spatial dimensions (height and width) decrease progressively due to pooling operations, while the number of filters increases to capture more complex and abstract features at deeper layers of the network. As the depth grows, more complex visual features can be learnt by the network by adding additional filters as the depth increases.

6.3. Batch normalization

After each convolution layer, for balancing process of learning, batch normalization layer is deployed. During training, it lowers the internal covariate shift or variation in the distribution of layer inputs. Model can learn more effectively with the help of batch normalization as it preserves a uniform distribution.
Each layer's output is normalized by applying the following equation:

$$\hat{x} = \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}$$

here:
x = input to the layer, σ2 = batch's variance, μ = batch's mean, ϵ = a small constant for numerical stability
It improves training speed and lowers the risk of disappearing or exploding gradients.

6.4. Max pooling layer

After convolutional blocks max pooling layers come. It reduces spatial resolution as well as preserves essential features. it reduces the number of parameters and network activities which helps in reducing computational complexity and managing overfitting.
From a small region of the feature map the maximum value is selected by the max pooling operation:

$$y = \max(x_{1,}\, x_{2,\ldots.}\, x_n)$$

here x = input features map.
Using 2 × 2 pooling windows the feature maps are down sampled in this architecture. It retains the most prominent features while discarding less significant information.
The feature map dimensions are reduced as in table V.

Table 5: Feature map dimension's architecture

Stage	*Output size*
After pooling 1	112 × 112 × 32
After pooling 2	56 × 56 × 64
After pooling 3	28 × 28 × 128
After pooling 4	14 × 14 × 256

The output size of each pooling layer is represented in the form of height × width × number of feature maps.

6.5. Global pooling average

Global average pooling is used instead of using traditional flattening layers in this architecture. This layer reduces spatial dimensions to single vector by calculating average value of every feature map. 14 × 14 × 256 feature map's dimensions are converted to 256 by global average pooling. It decreases the parameter count and helps avoid overfitting.

6.6. Fully connected layer

For performing classification, through fully connected layers the extracted feature vector is passed. First dense layer contains 128 neurons which learn high-level feature interactions. By randomly disabling neurons during training dropout regularization is applied to reduce overfitting. Prior to classification, for feature learning, 128 neurons are utilized in fully connected dense layer; but the number of features

produced after global average pooling is 256.
The dropout operation can be represented as:

$$y = r.x$$

where r is a binary mask that randomly drops neurons during training.

6.7. Output layer

There are 4 neurons in the final output layer which corresponding to the four classes: healthy, leaf smut, bacterial leaf blight and brown spot.
For converting the output values into class probabilities a Softmax activation function is applied:

$$\mathrm{P(y_i)} = \frac{\mathrm{e}^{\mathrm{z_i}}}{\sum_{\mathrm{j=1}}^{\mathrm{k}} \mathrm{e}^{\mathrm{z_j}}}$$

here: z_i= output score for class i, k = number of classes.
Class having highest probability is chosen as predicted class.

6.8. Model complexity

Proposed model's complexity is analyzed with regards to its parameters counts. Proposed CNN architecture contains 424,262 total parameters. Out of total parameters 423,044 are trainable parameters which are actively updated during backpropagation. These trainable parameters include convolutional kernel weights and biases in fully connected layers, which are optimized using the Adam optimizer. Model is of size approximately 1.62 MB. The proposed model is significantly lighter while still achieving competitive classification performance as Compared to deeper architectures such as VGG16, MobileNetV2 and ResNet50. It lowers complexity which makes the model appropriate for resource-constrained environments and real time agricultural implementations.

6.9. Evaluation metrics

Standard classification metrics such as accuracy, precision recall and F1 score are applied to analyze model's performance. They also help to check classification quality. For class-wise predictions analysis and to recognize patterns of misclassification, confusion matrix is used.
overall classification accuracy appears defined as:
Accuracy shows proportions of predictions model got correct.

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

Precision could examine positive predictions. Furthermore, precision checks how many were actually right.

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

Recall may represent model's capability to correctly detect all relevant samples:

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

The significant F1-score could provide important measure between precision and recall:

$$F1 = 2 * Precision * \frac{Recall}{Precision+Recall}$$

Here, TP, TN, FP and FN represents true positives, true negatives, false positives and false negatives respectively.

7. Experimental Findings and Discussions

7.1. Qualitative Performance Analysis

The proposed model is trained for 25 epochs. The model is trained using 70-15-15 dataset split (11238 training, 2409 validation and 2411 test images). Using accuracy and loss curves, the learning behavior of the model is analyzed.

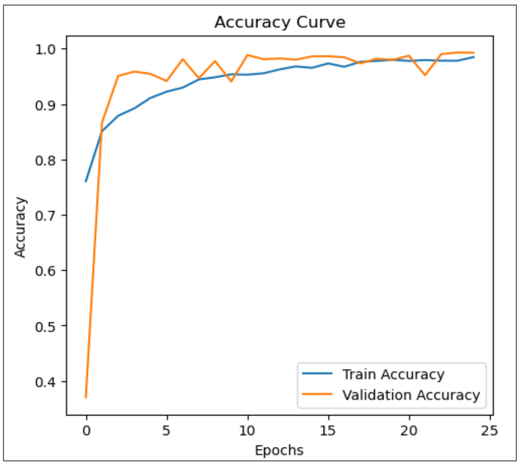

Fig 4: Graph of model accuracy

Fig. 4 shows that on the validation and test sets, around 98% - 99% training accuracy is attained by the trained CNN model and it hits around 99% validation accuracy. It shows that proposed model effectively identified the essential features and there is absence of significant overfitting.

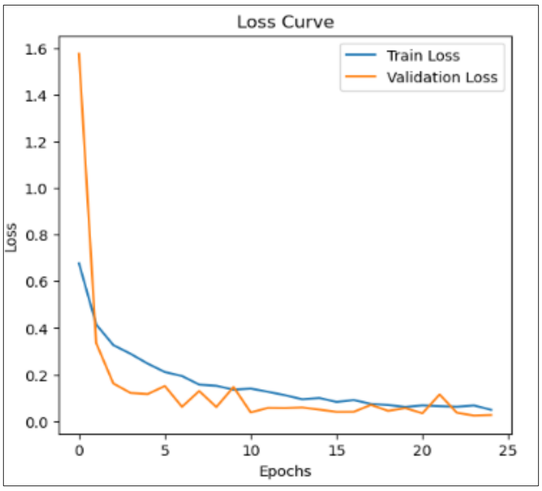

Fig 5: Model loss graph

Fig. 5 reveals that training loss decreased and reached nearly 0.03-0.05. Validation loss is also decreased from 2 to 0.05. This figure confirms there is stable convergence, high confidence predictions and effective feature learning.

Table 6: Proposed model's class-wise performance results

Name of the class	Precision	Recall	F1 score	Support
Bacterial leaf blight	1.00	0.98	0.99	613
Brown spot	1.00	0.99	0.99	602
Healthy	0.98	1.00	0.99	601
Leaf smut	0.99	1.00	1.00	595

Across all disease classes, strong performance is shown by the model and attained nearly perfect precision, recall and F1 score. Model has attained perfect or near perfect precision and recall for all classes. Under the given dataset conditions, high reliable and robust classification capability is shown by the model.

7.2. Test set evaluation

To analyze model's performance on new data, it was tested on novel test set which was excluded from validation and training. On the test data, the model sustained good performance and attained accuracy around 98.75%. No evidence of train-test data leakage was observed. Thus, unseen rice leaf images can be handled efficiently by the proposed CNN model.

7.3. Confusion matrix analysis

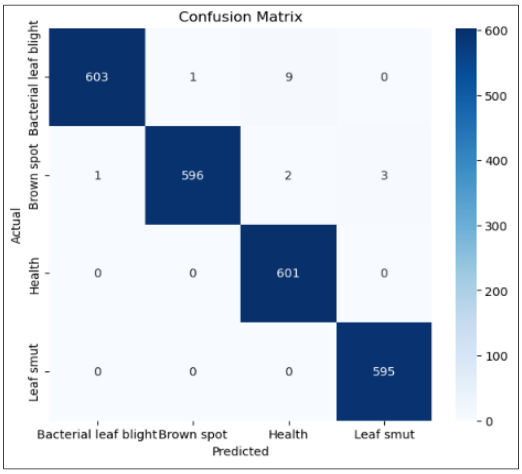

Fig 6: Proposed model's confusion matrix

Proposed model's confusion matrix of indicates that across all classes, majority of the samples are classified accurately with minor classification failure that occurs between bacterial leaf blight and brown spot and between healthy and brown spot. This may be due to same visual patterns such as distribution of colors and lesion texture. This is common observation among prior research. This error states that in future enhancements could focus on fine-grained feature learning.

7.4. Comparison with transfer learning models

To see how well proposed model performed, it is compared with a transfer learning based frameworks like MobileNetV2, ResNet50 and Vgg16. The transfer learning models, which are already trained on ImageNet, are examined using the same dataset and experimental settings.

Among them comparison is done based on accuracy, model size, time per image and number of parameters. Table VII shows the comparison among them.

Table 7: Comparison among proposed model, MobileNetV2, ResNet50 and VGG16

Model	Accuracy (%)	Model size (MB)	Time per image (ms)	Number of parameters	Overall suitability
Proposed model	99.33	1.618	9.512	424260	Best
MobileNetV2	97.17	8.633	14.141	2263108	Good alternative
ResNet50	46.93	90.01	64.661	23595908	Not suitable
Vgg16	79.75	56.13	193.830	14716740	Heavy

From table VII it is clear that proposed model have better accuracy, time per image despite of having lower parameters and model size than others. So overall, the proposed model is achieved best tradeoff between accuracy, speed and model size which makes it more suitable for real life disease detection.

7.5. Discussion

Experimental results showed that superior classification performance is achieved by the proposed framework by achieving 99.33% accuracy on the test set. It shows models ability to learn visual patterns from images of rice leaves. Its hierarchical feature extraction capability is one of the main reason for its performance. In the early convolutional layers, low-level features present on rice leaf surfaces are learnt by the network. These characteristics can be helpful in identifying differences in background patterns and leaf structure. As the input moves through deeper convolutional layers more complex and disease-specific patterns, like lesion shapes, discoloration regions, spot distributions and texture anomalies that define various rice illnesses, are captured by the model. The network's deeper layers are able to identify disease patterns by merging several low-level features into higher-level representations. The training process is stabilized by batch normalization which also enhances convergence. By reducing spatial dimensionality max-pooling layers permit the model to aim on the most prominent features. By requiring the network to summarize feature maps rather than memorize spatial patterns, number of parameters are reduced and overfitting is minimized by global pooling average. To enhance generalization during training, dropout regularization helps in preventing the model from relying too heavily on specific neurons. As a result, while sustaining a fairly compact model size and computational complexity the proposed CNN is able to effectively learn discriminative features from rice leaf images. In opposite, VGG16 achieved 79.75% accuracy, MobileNetV2 achieved 97.17% accuracy while ResNet50 achieved nearly 46% under same conditions. Despite of having lower parameters than other models, proposed model performed well. This shows that when dataset is domain specific, well-designed task -specific architecture can perform well then deeper pre-trained models. Overall the robustness, effectiveness and efficiency of the presented approach are validated by results for identification or categorization of rice leaf disease.

7.6. Conclusion

In this study, for automated disease categorization of rice leaf using image data, a CNN based framework is presented. While maintaining lower computational requirements, the proposed model hits 99.33% prediction accuracy along with strong recall, precision and F1 score which make it appropriate for real world agricultural implementation. Proposed model has several advantages like it is lightweight architecture, has low computational cost, fast inference time and it is appropriate for mobile deployment. A comparison analysis between proposed model and transfer learning models like MobileNetV2, VGG16 and ResNet50 shows that although already trained models perform well, but better disease classification and greater efficiency can be provided by the proposed CNN model which is task-specific. The results show that for faster and better decisions in real- world, the suggested CNN model can be helpful to detect rice leaf diseases accurately. In future, dataset can be expanded with additional disease classes. In real-time mobile or web-based systems, the model can be deployed.

References

1. Pai P, Amutha S, Basthikodi M, Shafeeq BMA, Chaitra KM. A twin CNN based framework for optimized rice leaf disease classification with feature fusion. J Big Data. 2025.
2. Hang J, Zhang D, Chen P, Zhang J, Wang B. Classification of plant leaf diseases based on improved convolutional neural network. 2019:1-14.
3. Open T, Journal A. Classification of various plant leaf disease using pretrained convolutional neural network on ImageNet. 2024:1-15.
4. Sutaji D, Rosyid H. Convolutional neural network (CNN) models for crop diseases classification. 2022;4(2).
5. Jadhav SB, Udupi VR, Patil SB. Convolutional neural networks for leaf image-based plant disease classification. 2019;8(4):328-341.
6. Tugrul B. Convolutional neural networks in detection of plant leaf diseases: a review. 2022.
7. Kurmi Y, Saxena P, Singh B, Suchi K, Chaurasia V, Goel A. Deep CNN model for crops' diseases detection using leaf images. Multidimens Syst Signal Process. 2022;33(3):981-1000.
8. Walle D, Tsehay G, Muluneh W. Deep convolutional neural network model for classifying common bean leaf diseases. Discov Artif Intell. 2024.
9. Vishnoi VK, Kumar K. Detection of apple plant diseases using leaf images through convolutional neural network. IEEE Access. 2023;11:6594-6609.
10. Development of concise convolutional neural network for tomato plant disease classification based on leaf images.
11. Narayanan S. Enhanced plant disease classification with attention-based convolutional neural network using squeeze and excitation mechanism. 2025.
12. Jadhav SB. Identification of plant diseases using convolutional neural networks. Int J Inf Technol. 2021;13(6):2461-2470.
13. Priyadharshini RA. Maize leaf disease classification using deep convolutional neural networks. Neural Comput Appl. 2019;31(12):8887-8895.
14. Vrochidou E. Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern. IEEE Access. 2023;11:62307-62317.
15. Upadhyay SK, Kumar A. A novel approach for rice plant diseases classification with deep convolutional neural network. Int J Inf Technol. 2022;14(1):185-199.
16. Hassan SKM, Maji AK. Plant disease identification using a novel convolutional neural network. IEEE Access. 2022;10:5390-5401.
17. Daniel T, Marsyita S, Hanafi B. Plant leaf disease detection and classification using convolution neural networks model: a review. 2025.
18. Ghosh H, Rahat IS, Shaik K, Khasim S, Yesubabu M. Potato leaf disease recognition and prediction using convolutional neural networks. 2023;10(6):1-9.
19. Sun X, Li G, Qu P, Xie X, Pan X, Zhang W. Research on plant disease identification based on CNN. Cogn Robot. 2022;2:155-163.
20. Lu J, Tan L. Review on convolutional neural network (CNN) applied to plant leaf disease classification. 2021:1-18.
21. Ayyappan AB, Kumar TGM. Rice plant disease detection using convolutional neural networks. Discov Artif Intell. 2025.
22. Jasrotia S, Yadav J, Rajpal N, Arora M, Chaudhary J. Convolutional neural network based maize plant disease identification. Procedia Comput Sci. 2023;218:1712-1721.

23. Premananda S, Pritamdas K, Jilenkumari K. Custom convolutional neural network for detection and classification of rice plant diseases. Procedia Comput Sci. 2026;218:2026-2040.
24. Wallelign S. Soybean plant disease identification using convolutional neural network. 2017:146-151.
25. India Brand Equity Foundation. Agriculture sector in India. Available from: http://www.ibef.org/industry/agriculture-india.aspx
26. Bahri B. Rice leaf disease dataset. Kaggle. Available from: https://www.kaggle.com/datasets/bahribahri/riceleaf?select=rice-leaf-desease
27. Rahman E. A dataset of rice leaf diseases. Kaggle. Available from: https://www.kaggle.com/datasets/eshatrahman/a-dataset-of-rice-leaf-diseases

How to Cite This Article
Singh J, Singh K. An efficient convolutional neural network based deep learning framework for rice leaf disease classification. J Agric Digit Res. 2026;7(2):1-8. doi:10.54660/JADR.2026.7.2.01-08