IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 3, June 2025, pp. 2547~2558
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i3.pp2547-2558

A novel light-weight convolutional neural network for rice leaf disease classification

Parthasarathi Jayaraman[1,2], Muthulakshmi Palaniyandi[1]
[1]Department of Computer Science, Faculty of Science and Humanities, SRM Institute of Science and Technology, Kattankulathur, India
[2]Department of Computer Science and Applications, Faculty of Science and Humanities, SRM Institute of Science and Technology, Vadapalani, India

Article Info

Article history:

Received Mar 27, 2024
Revised Feb 7, 2025
Accepted Mar 15, 2025

Keywords:

Adam optimizer
Convolutional neural network
Deep learning
Rice leaf disease
Transfer learning

ABSTRACT

Rice is one of the primary sources of staple meals. It may turn out to be a disaster as the production of agricultural products is declined due to diseases and therefore it is required to straighten up the situation by taking precautionary measures. Generally, deep learning (DL) architectures are employed for the identification of plant leaf diseases and it is observed that there is a trade-off between the accuracy and parameters. This study introduces a light-weight architecture called rice leaf disease classification convolutional neural network (RLDC-CNN). The objective of the proposed architecture is to improve the accuracy and reduce the loss by using a combination of convolutional layers, maxpooling layers, and fully connected layers. These layers use activation function for non-linearity, dropout for regularization and implements hyperparameter tuning with various optimizers that include Adam, RMSprop, stochastic gradient descent (SGD) and adaptive gradient (AdaGrad). Experiments are conducted on the dataset of 7,096 images with batch size of 32 under various learning rates. The behavior is analyzed by comparing the existing models and the count of parameters (in millions) equipped by RLDC-CNN, DenseNet121, VGG-16, and ResNet50 is 0.65, 8.49, 15.44, 26.49 with accuracy of 99.15%, 98.94%, 97.82%, 96.48% respectively.

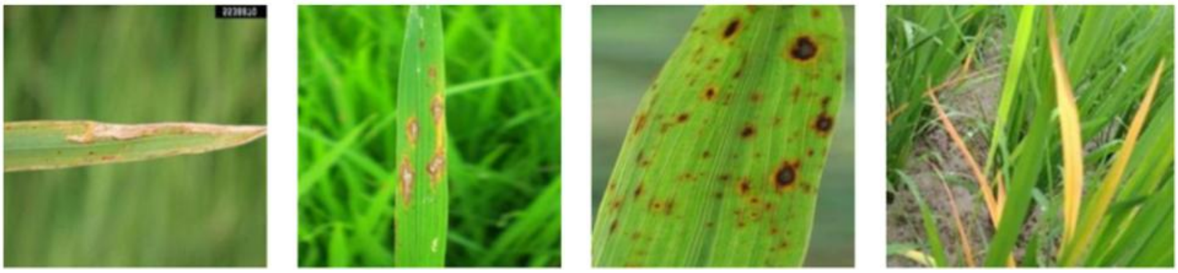
Corresponding Author:

Muthulakshmi Palaniyandi
Department of Computer Science, Faculty of Science and Humanities
SRM Institute of Science and Technology
Kattankulathur, Chennai, Tamilnadu, 603203, India
Email: muthulap@srmist.edu.in

1. INTRODUCTION

In the modern sphere of agriculture, the primary focus has shifted towards the critical issues of ensuring food security and optimizing the crop yields. Rice crops, a dietary staple for a substantial portion of the global population, are constantly under the looming threat of various diseases that are potential to impair both the quantity and quality of yields. The symptoms and indications induced by the pathogens are utilized for the purpose of identifying and categorizing the diseases that affect the rice crops.

Generally, these pathogens are fungi and viruses, that induce the illnesses such as bacterial blight, blast, brown spot, and tungro. Mostly, these diseases impact the foliage of the rice plant. The images representing the pathogens are shown in Figure 1. Figure 1(a) illustrates bacterial blight, a prevalent disease in rice. The disease manifests as chlorosis of foliage or desiccation of seeds. Figure 1(b) illustrates the occurrence of a blast, which is a result of fungal infection. The entire leaf exhibits a charred look, resulting in a decline in grain quality. The fungal spot, also known as brown spot, is depicted in Figure 1(c). It initially appears as a brown dot and then takes on a cylindrical or circular shape. In severe situations, it can affect up to 50% of the yield. Figure 1(d) depicts tungro, a disease caused by the presence of two distinct viruses. Tungro illness manifests as stunted growth and decreased grain production.



(a) (b) (c) (d)

Figure 1. Infected rice leaves (a) bacterial blight, (b) blast, (c) brown spot, and (d) tungro

The manual identification of these diseases poses significant challenges and requires a substantial amount of time. The importance of swiftly and accurately identifying these diseases cannot be overstated as it is vital for the timely deployment of effective strategies to safeguard agricultural productivity and economic stability [1]. Currently, several artificial intelligence (AI) approaches are used to detect and classify crop disease including k-nearest neighbors (kNN), decision tree (DT), support vector machine (SVM) [2], logistic regression, and convolutional neural networks (CNN) [3], [4].

Among several AI approaches, CNN is the most known and widely applied method for image analysis in agricultural research [5], [6]. CNNs are a type of deep, feed-forward artificial neural networks (ANN) that has gained significant prominence in the field of computer vision (CV) and its applications [7], [8]. It is found that these technologies can quickly learn complex problems because of the ability to share weights and known for encouraging scalability and massive parallelization [9], [10]. The main advantage of the CNN model is its ordered structure and huge learning capacity that which may solve complex challenges with flexibility and adaptability [11], [12]. Though the models of transfer learning (TL) have proved their learning abilities on larger datasets, it is found from those models that are proposed in recent past are capable of learning from smaller datasets [13], [14]. In contrast, training requirements for CNN could be more practical when considering scalability issues that focus on time [15], [16]. CNN can increase the likelihood of accurate classification; when extensive datasets are provided to describe the problem [17], [18].

During the survey, it is seen that many researchers have employed diverse deep learning (DL) algorithms to detect diseases in rice plants. Rahman *et al*. [19] proposed an architecture that addresses the interclass variation pertaining to various stages of specific disease. The experiment uses 1,426 images of rice plant and results in an accuracy of 94.33% with parameter count of 0.8 million. Krishnamoorthy *et al.* [20] proposes a model exclusively for the classification of rice leaf diseases, that addresses effectively vanishing gradient and deterioration. Also, the method incorporates global average pooling to decrease the parameter count. The model uses 5,200 images and the accuracy rate is found to be 95.67%.

Sethy *et al.* [21] conducted an experiment using 11 distinct CNN models using TL approach. The dataset consists of 5,932 images that represent four different types of illnesses affecting rice leaves. The performance of one of the 11 models that implements ResNet50 and SVM is found to be good with F1-score of 98.98%. However, it is observed that the utilization of TL amplifies the intricacy of the model. Deb *et al.* [22] deployed five distinct TL models to analyze a collection of 7,096 images of rice leaves, each representing one of five different illness groups. The models employed in the study include AlexNet, VGG-16, ResNet-18, MobileNet-V2, and InceptionV3. The performance of InceptionV3 demonstrates its efficiency with an accuracy rate of 96.23%.

Saleem *et al.* [23] present an approach that uses mutant particle swarm optimization (MUT-PSO) to detect rice leaf diseases to identify the most optimal CNN architecture. In this study, two datasets that consists of four classes each assess and observed an average accuracy of 96.62%. Chen *et al.* [24] proposed a lightweight inception network for the classification of rice leaf diseases, which incorporates the MobileNet architecture. The study uses a dataset consists of 12 distinct classes and experiment outcomes results with an accuracy of 97.89%. Joshi *et al.* [25] introduced a CNN framework designed specifically for rice leaf images. The architecture involves a series of convolutional layers that are utilized to identify and classify leaf conditions and found that the model came up with an accuracy of 93.75%.

Based on the literature survey, it is found that the existing works are to detect the diseases in rice leaf have used TL and tailored CNN architectures. Typically, TL architectures employ a large number of parameters and yield high levels of accuracy [26]. However, the utilization of bespoke architectures results in a reduction of parameters, albeit at the expense of accuracy [27]. There is a scope to create a CNN model that enhances the accuracy provided by TL models while minimizing the number of parameters. This reduction in parameters leads to less time complexity of the model.

The proposed work aims to create a CNN architecture by combining domain expertise, modern neural network architectures, and the wealth of knowledge encoded existing models. The architecture may distinguish between the various disease manifestations that affect rice crops, that includes bacterial blight, blast, brown spot, and tungro. The significant contributions of the proposed study include, i) the development of a specialized CNN architecture known as rice leaf disease classification (RLDC)-CNN with fewer parameters. ii) the feature extraction process from the input image that utilizes a series of convolutional layers, max pool layers, and activation functions at various stages to introduce non-linear behavior. iii) the model uses Adam, RMSprop, SGD and adaptive gradient (Adagrad) optimizers and compared the level of optimization. iv) the evaluation of the model uses accuracy, precision, recall and F1-Score. v) the results are compared with the state-of-the-art (SOTA) models.

The subsequent sections of this work are organized as follows, section 2 provides a comprehensive analysis of the dataset used for training and evaluation. The simulation findings are explained in section 3. Section 4 focuses on the conclusion, where the summary of the work and prospective directions for further research are suggested.

2. METHOD

This section outlines the sequential procedures involved in the proposed methodology. The procedure commences with gathering the dataset, which is subsequently subjected to data pre-processing. This entails doing operations such as adjusting the size, enhancing, and standardizing the images. Subsequently, an RLDC-CNN model is employed to extract features from the data. Hyper-parameter optimization involves adjusting the values of the epoch count, learning rate, and optimizers. Ultimately, the diseases are categorized according to the retrieved characteristics. The proposed methodology consists of three sections namely, i) data preparation and pre-processing; ii) feature extraction; and iii) hyperparameter optimization. The workflow of the proposed work is shown in Figure 2.

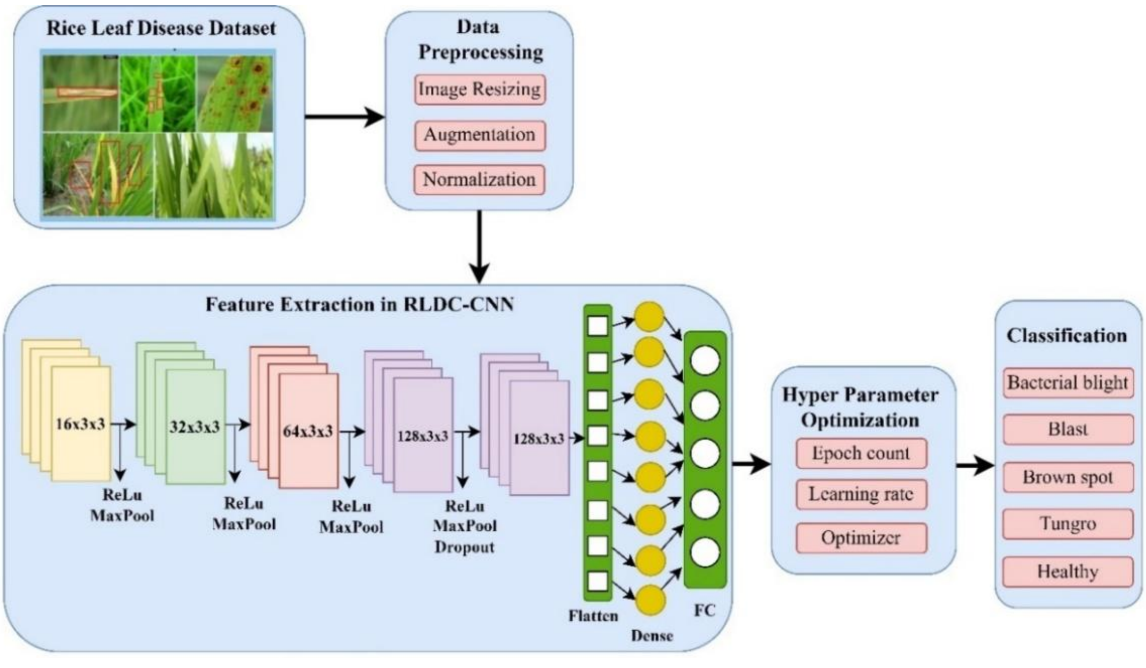


Figure 2. Workflow of the proposed methodology

2.1. Data preparation and preprocessing

The study uses dataset curated from benchmarked Mendeley and Kaggle repository [28], [29]. It encompasses a collection of 7,096 rice leaf images that shows the presence of diseases that may be classified across four classes namely, i) bacterial blight; ii) blast; iii) brown spot; and iv) tungro. Also, the dataset is included with images of healthy leaves. The dataset comprises of images of plants that are grown under different environmental conditions, growth stages, and disease severities.

The images are converted to the size to fit the requirement of the DL pipeline. A series of pre-processing steps are taken to standardize and stabilize the suitability of the dataset for training and evaluation. The size of the images is resized to 224×224 pixels from their original size of 300×300 pixels. The use of data normalization ensures that each pixel value should be in the range between 0 and 1, which leads to quick convergence. The pre-processed dataset is divided into two categories namely, i) the dataset for training and ii) the dataset for testing. Among the 7,096 images considered for conducting the experiment, 80% of the images are used for training and the remaining 20% images are used for testing. Consequently, the dataset is balanced, as the 7,096 images are distributed nearly equitably among the five classes. The distribution of train and test images is illustrated in Figure 3.

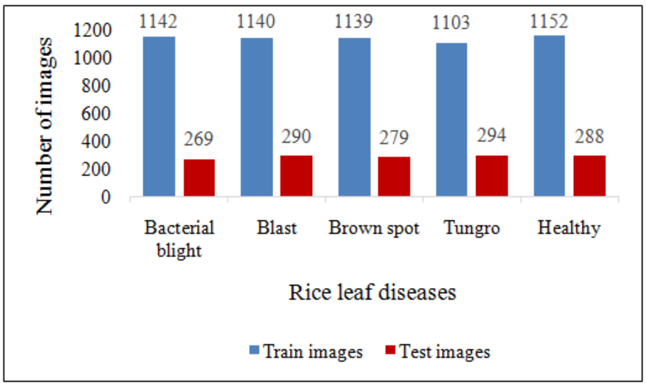


Figure 3. Rice leaf disease dataset description

2.2. Feature extraction

The feature extraction procedure accepts pre-processed data as input and converts the raw data into numerical features, while maintaining the information from the original data set. The RLDC-CNN model is composed of multiple layers that are used to extract features. They are i) convolutional layer; ii) pooling layer; iii) activation function; iv) dropouts; v) flatten layer; and vi) fully connected (FC) layer and is shown in Figure 4.

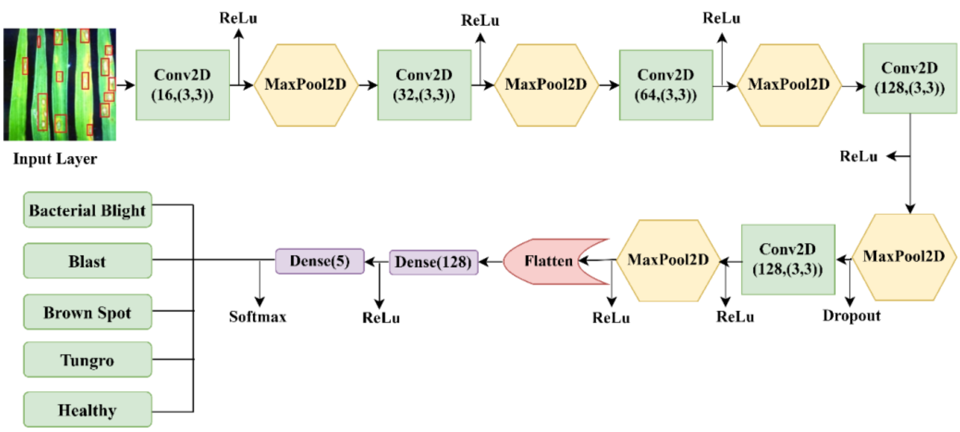


Figure 4. The proposed RLDC-CNN architecture

The proposed RLDC-CNN architecture consists of an input layer, 13 feature extraction layers that includes 5 convolutional layers, 5 maxpooling layer, 1 flatten layer, and 2 FC layers. The workflow starts with the input layer followed by the alternative arrangement of convolutional layer and maxpooling layer and then continues with the flatten layers to reach out the last two dense FC layers that proceed to the classification of images, the output. The training process begins with taking in the input image and then processed through a convolutional layer that consists of 16 kernels, each of size (3×3). The convolution procedure generates 16 feature maps, each of dimensions (222×222). These feature maps are subsequently passes on to the maxpooling layer that extract the vital features and reduces the image dimensions to the size (111×111). The process is repeated by passing the generated feature maps to the next subsequent convolution layers containing 32, 64, 128 kernels that are accompanied by maxpooling layer.

In order to encourage non-linearity, the rectified linear unit (ReLU) activation function is used in combination with all convolution operations. To mitigate overfitting, the proposed work used a dropout layer with the rate of 0.2 and the same is implemented after the fourth convolution operation. This layer randomly deactivates a fraction of neurons during each training iteration and therefore an improvement in the model's resilience shall be observed. The process continues with the flatten layer that converts the combined feature maps that are obtained across the layers to form a single-dimensional vector of size 3,200. The vector is fed as input to the FC layer, which is responsible for combining the low-level features that are learned from the convolutional layers in order to form the high-level feature maps. In this study, the first FC layer is configured with 128 neurons which is followed by an output FC layer with 5 neurons that is equivalent to categories of diseases to be classified. The SoftMax activation function is followed by the output layer that gives the probabilistic outcome that may fit the rice leaf disease in one of the 5 classes. Table 1 represents the configuration summary of RLDC-CNN model that provides an overview of the architecture. The summary includes the details of the layers that determines the output shape and the number of parameters used across layers.

Table 1. RLDC-CNN configuration summary

Layer	Output shape	Parameters
Conv2D	(None,222,222,16)	448
Maxpool2D	(None,111,111,16)	0
Conv2D	(None,109,109,32)	4,640
Maxpool2D	(None,54,54,32)	0
Conv2D	(None,52,52,64)	18,496
Maxpool2D	(None,26,26,64)	0
Conv2D	(None,24,24,128)	73,856
Maxpool2D	(None,12,12,128)	0
Dropout	(None,12,12,128)	0
Conv2D	(None,10,10,128)	147,584
Maxpool2D	(None,5,5,128)	0
Flatten	(None,3200)	0
Dense	(None,128)	409,728
Dense	(None,5)	645
Activation	(None,5)	0
Total Parameters		655,397

2.3. Hyperparameter optimization

This section presents the optimization technique that is implemented in the proposed model. The objective function can be represented as θ that maximizes the accuracy and minimizes the loss. The objective function can be defined using the parameters that include ‘D-Train’, ‘n’, ‘M/k’, where D-Train represents the number of training images, ‘n’ stands for number of epochs and ‘M/k’ is the ratio for every batch ‘k’ of size 32 with the training images ‘M’. Here, the epochs represent the number of times that the model executes the training dataset D-Train. For every epoch, the D-Train is split into batches. Each batch contains the samples that are considered for the updating of the model parameters. The loss function (L) is applied to the batches obtained that represents the discrepancy between the actual label y_i and the predicted label $\hat{y}_i$. ‘L ’ is shown as (1):

$$\mathbf{L} = -\sum_{i=1}^{c} y_i(\log(\hat{y}_i)) \quad (1)$$

The mean of losses of all batches is represented as cost function (C) and is represented as (2):

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n} \sum_{j=1}^{c} [y_{ij} * \log(\hat{y}_{ij})] \quad (2)$$

In general, optimizers adjust the value of parameters to minimize the loss function. The proposed work employs stochastic gradient descent (SGD), Adagrad, RMSprop, and Adam optimizers to minimize the loss function. When RLDC-CNN is completely trained, the weight and bias values are adjusted with respect to LR. The weight and bias values are calculated using the (3) and (4):

$$W_t \leftarrow W_{t-1} - \eta \nabla_W(\text{loss}(W)) \tag{3}$$

$$b_t \leftarrow b_{t-1} - \eta \nabla_W(\text{loss}(b)) \tag{4}$$

where, W_{t-1} and b_{t-1} are the values of the weight and bias obtained during the previous execution, η is the learning rate (LR), ∇_W represents the rate of change in the derivative with respect to weight.

The proposed RLDC-CNN model updates the objective function θ after the completion of 30 training epochs with LR of 0.001. Where θ contains the minimized arguments for weight, bias, loss and cost can be applied on unseen dataset D-Test. The learning parameters used for the implementation of RLDC-CNN is given in Table 2.

Table 2. RLDC-CNN learning parameters

S.No.	Parameters	Value
1	Optimizer	SGD, Adagrad, RMSprop, Adam
2	Loss function	Sparse categorical crossentropy
3	Learning rate	0.001
4	Batch size	32
5	Epochs	30

3. RESULTS AND DISCUSSION

This study investigates the effects of customized CNNs for RLDC. It is found from the existing works given in section 1 that the models use TL models are not addressing the influence of optimizers that may increase the efficiency of the model. The proposed RLDC-CNN model outperforms the TL models in terms of accuracy and parameter reduction. The infrastructure setup for conducting the experiment includes, an Intel-core i5-82650 CPU operating at a frequency of 1.60 GHz with 8 GB of RAM. Google Colaboratory GPU is utilized as hardware accelerator in the Python framework that uses TensorFlow and Keras.

The RLDC-CNN model is trained using the dataset of size 5,676, which is said to have balanced proposition of images with infections that can be classified across five classes, viz., bacterial blight; blast; brown spot; tungro; and healthy. The efficacy of the proposed model is assessed to gauge the overall performance and quality, through the consideration of various levels of severity and circumstances. The optimizers are individually combined with the model to analyze the performance that entails a dataset of 1,420 images taken from D-Test. The proposed model is evaluated using the metrics, accuracy; precision; recall; and F1-score is represented using the (5) to (8) as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{7}$$

$$F_1 - \text{Score} = 2 \text{ X } \frac{\text{Precision X Recall}}{\text{Precision + Recall}} \tag{8}$$

where, (TP) represents true positive, (TN) is true negative, (FP) indicates false positive, (FN) specifies false negative.

The analysis of the RLDC-CNN is described in the following subsections. Section 3.1 addresses the performance of the proposed model implementing one optimizer at a time. Section 3.2 discusses the efficiency of the proposed model with other pre-trained models. Section 3.3 presents the comparison of RLDC-CNN model and existing literatures.

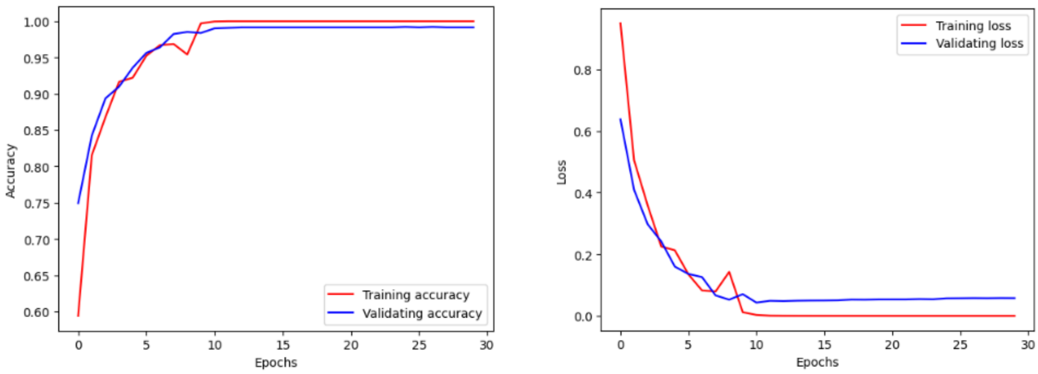
3.1. Performance evaluation of RLDC-CNN with different optimizers

The proposed RLDC-CNN model implements the optimizers namely, Adam; RMSprop; SGD; and Adagrad. The implementation combines the proposed model and one of the optimizers per time to get the optimized results. It is observed from the results that the Adam optimizer achieves the highest accuracy of 99.15%. It incorporates the most advantageous characteristics of other optimizers, specifically Adagrad and RMSprop, which is observed to yield improved convergence of the model. The performance of the RLDC-CNN model using different optimizers is presented in Table 3.

Table 3. RLDC-CNN model with different optimizers

Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Adam	99.15	99.15	99.13	99.13
RMSprop	98.31	98.3	98.3	98.29
SGD	97.96	97.96	97.92	97.93
Adagrad	79.72	80.8	79.57	80.18

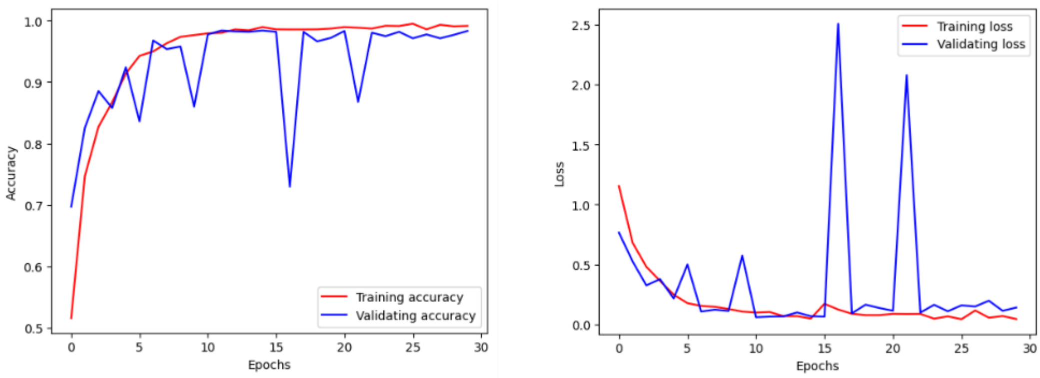
Figures 5 to 8 shows the accuracy and loss plots, which represents the information of the convergence of the RLDC-CNN model that uses different optimizers. It is evident from Figure 5(a) that the Adam optimizer excels in terms of accuracy when compared to other optimizers. The model exhibits a convergence rate that is exceedingly rapid, with an accuracy of 99%. Figure 5(b) demonstrates that the Adam optimizer reduces the model's loss in the 10th epoch and that the results do not exhibit any form of overfitting.



(a) (b)

Figure 5. Training and validation plots of RLDC-CNN with Adam optimizer (a) accuracy and (b) loss

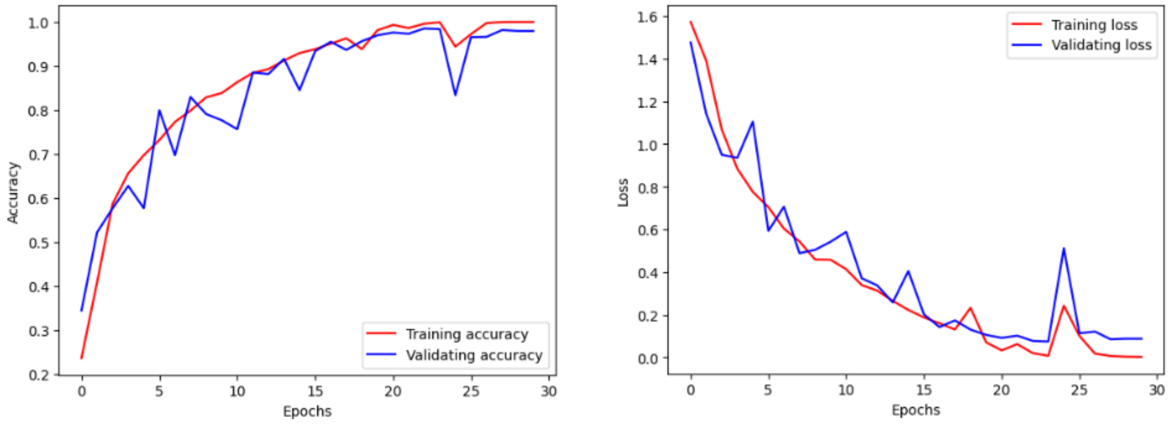
The impact of the RMSprop optimizer is illustrated in Figure 6. Figure 6(a) exhibits the model's ability to attain an accuracy of 98%. Additionally, the model's loss is illustrated in Figure 6(b), which contains numerous peaks. Nevertheless, the peaks observed in the plots suggest that a regularization is warranted, which could potentially intensify the model's complexity.



(a) (b)

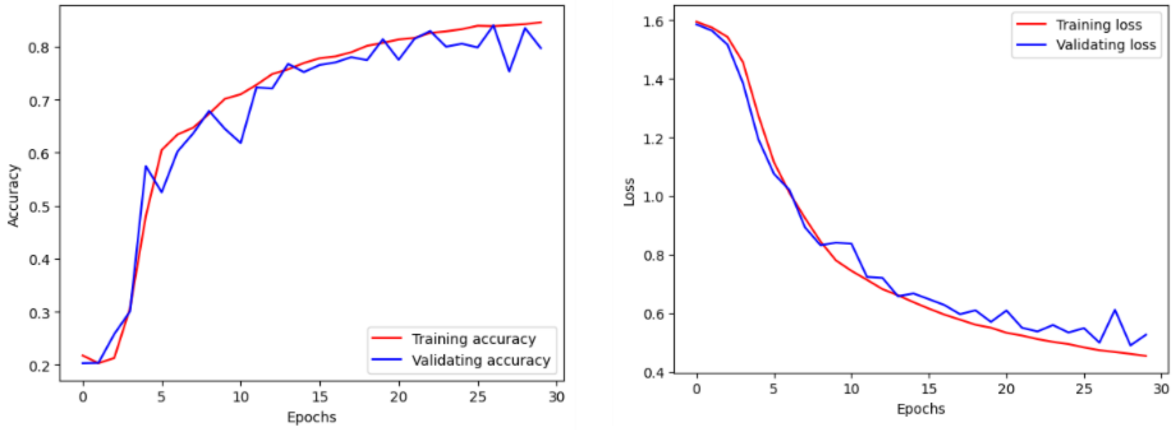
Figure 6. Training and validation plots of RLDC-CNN with RMSprop optimizer (a) accuracy and (b) loss

The behaviour of SGD and Adagrad is illustrated in Figures 7 and 8, and it is evident that the model converges after a considerable period. The accuracy of the SGD optimizer, which requires additional time to converge, is depicted in Figure 7(a). Figure 7(b) illustrates the SGD optimizer's loss, which is nearly converging at the 30th epoch, but still exhibits peaks. Figure 8(a) illustrates the Adagrad optimizer's accuracy, which is significantly lower than that of other optimizers due to its sluggish learning rate. Additionally, Figure 8(b) demonstrates that a substantial loss is observed when compared with the Adam optimizer.



(a) (b)

Figure 7. Training and validation plots of RLDC-CNN with SGD optimizer (a) accuracy and (b) loss

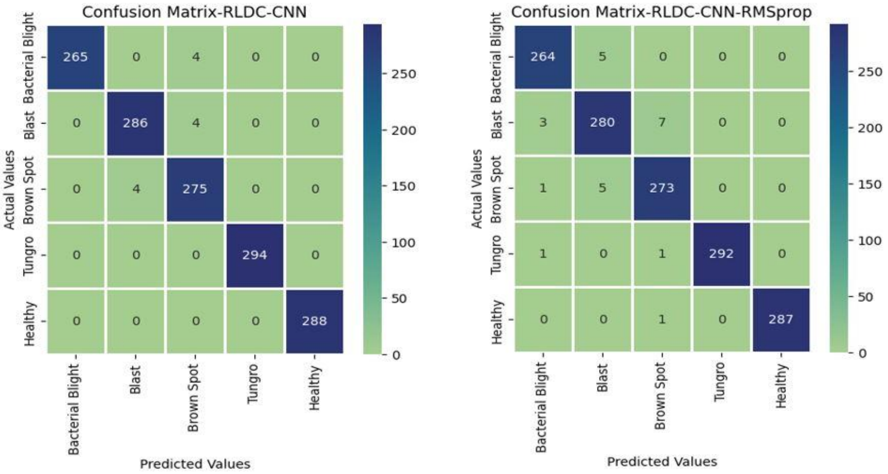


(a) (b)

Figure 8. Training and validation plots of RLDC-CNN with Adagrad optimizer (a) accuracy and (b) loss

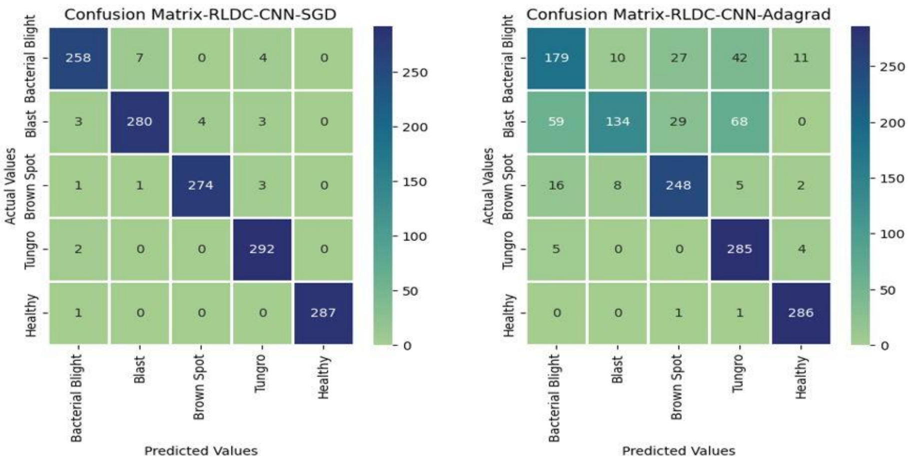
Figures 9 and 10 shows the confusion matrix that exhibit the performances of the RLDC-CNN model with the optimizers: Adam, RMSprop, SGD, and Adagrad. From the confusion matrix in Figure 9(a), it can be shown that Adam optimizer performs better than the other optimizers, with the fewest misclassifications. The confusion matrix in Figure 9(b) illustrates the performance of RLDC-CNN with RMSprop optimizer in distinguishing blast disease from other diseases, indicating difficulties in accurate classification.

The confusion matrix displayed in Figure 10(a) clearly indicates that there are approximately 28 instances of misclassifications. Which is higher compared to the results obtained with the adam optimizer shown in Figure 9(a). Figure 10(b) demonstrates the Adagrad optimizer. Which has 288 misclassifications, indicating that this optimizer performs poorly.



(a) (b)

Figure 9. Confusion matrix (a) RLDC-CNN-Adam and (b) RLDC-CNN-RMSprop



(a) (b)

Figure 10. Confusion matrix (a) RLDC-CNN-SGD and (b) RLDC-CNN-Adagrad

3.2. Performance evaluation of RLDC-CNN with different pre-trained models

The proposed RLDC-CNN model combines each one of the optimizers has shown that the model performs better with Adam optimizer. In this section, a study is made to assess the performance of the proposed RLDC-CNN model with Adam optimizer when compared with pre-trained models Viz., DenseNet121, VGG16, and ResNet50. The study involves 7,096 images of rice leaf diseases to conduct the experiment of the factors namely, i) the number of parameters and ii) accuracy. Table 4 shows the effectiveness of the proposed model that results in an accuracy of 99.15% considering a minimum of 0.65 million parameters when comparing the parameters with other optimizers.

Table 4. Performance comparison RLDC-CNN with various pre-trained models

Model	Parameters (in Million)	Accuracy (%)
RLDC-CNN+Adam	0.65	99.15
DenseNet121	8.49	98.94
VGG16	15.44	97.82
ResNet50	26.49	96.48

3.3. Comparative study of RLDC-CNN with existing CNN models

Table 5 presents a comparative analysis of the RLDC-CNN model with other models addressed in the literature, as discussed in section 1. The superiority of the RLDC-CNN is apparent, as it surpasses the other models in both parameter usage and accuracy. The existing models utilize pre-trained architectures such as Inception ResNetV2, InceptionV3, and ResNet50 as part of their DL architectures. It is important to mention that pre-trained models need a substantial number of parameters, that may lead to high computational and memory requirements. This may become a serious issue when attempt to implement in embedded systems. The proposed RLDC-CNN architecture is carefully designed to improve the accuracy with minimum number of layers that results in good performance with optimized parameters.

Table 5. Performance comparison RLDC-CNN with existing models in literature

Authors	DL architecture	Parameters (Million)	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Krishnamoorthy *et al.* [20]	Inception ResnetV2	55.9	95.67	96.5	96.5	96.5
Sethy *et al.* [21]	ResNet 50+SVM	25.6	98.25	98.25	98.26	98.26
Deb *et al.* [22]	Inception V3	23.9	96.23	96.4	96.2	96.2
Proposed methodology	RLDC-CNN	0.65	99.15	99.15	99.13	99.13

4. CONCLUSION

Rice is the primary food source for a majority of the world's population. Various diseases impact the rice plant at different phases of its growth. The utilization of CV in conjunction with DL architecture can effectively identify certain disorders. This research presents RLDC-CNN, a novel light weight customized DL model to classify the rice leaf diseases. The model is evaluated using 7096 images taken from Mendeley and Kaggle repositories. The architecture of the model is made up of series of convolutional and max pool layers that are arranged alternatively. These layers in the model are used for feature extraction. In order to introduce non-linearity, the model incorporates the ReLU activation function. The dropout layer is used to deactivate certain neurons that may results in a simple model with great efficiency. To ensure optimization, the model uses Adam Optimizer which has a learning rate of 0.001 and sparse categorical entropy loss function. The optimization process is performed over 30 epochs, with each epoch consisting of a batch size of 32. This methodology facilitates the effective convergence and the use of minimum number of parameters. The RLDC-CNN model achieved an accuracy rate of 99.15% while utilizing a relatively small parameter count of 0.65 million. The proposed model is compared with various pre-trained models that include DenseNet121, VGG16, and ResNet50, and the proposed RLDC-CNN model outperforms other compared models. The proposed model would definitely help the farmers to identify the rice leaf diseases, which may lead to choose the appropriate procedure to cure the same. The accurate and timely diagnosis shall increase the productivity at very higher rate. The study's significant constraint is that variations in illumination or the presence of shadows from one image to another do not influence the dataset. It has the potential to diminish accuracy. In future studies, it would be beneficial to incorporate datasets including images of this nature. Moreover, it can be extended to include explainable AI, which would provide farmers with precise advice based on discovered ailments and effective solutions to resolve the problem.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Parthasarathi Jayaraman	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Muthulakshmi Palaniyandi	✓	✓				✓			✓	✓	✓	✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Mendeley Data at https://data.mendeley.com/datasets/fwcj7stb8r/1, reference number [28]; and Kaggle dataset link at https://www.kaggle.com/datasets/leonardoaruizv/paddy-doctor-dataset-lr, reference number [29].

REFERENCES

[1] D. I. Patrício and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: a systematic review," *Computers and Electronics in Agriculture*, vol. 153, pp. 69–81, 2018, doi: 10.1016/j.compag.2018.08.001.

[2] R. Sujatha, J. M. Chatterjee, N. Z. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocessors and Microsystems*, vol. 80, 2021, doi: 10.1016/j.micpro.2020.103615.

[3] S. K. Noon, M. Amjad, M. A. Qureshi, and A. Mannan, "Use of deep learning techniques for identification of plant leaf stresses: A review," *Sustainable Computing: Informatics and Systems*, vol. 28, 2020, doi: 10.1016/j.suscom.2020.100443.

[4] Y. Wang, H. Wang, and Z. Peng, "Rice diseases detection and classification using attention based neural network and bayesian optimization," *Expert Systems with Applications*, vol. 178, 2021, doi: 10.1016/j.eswa.2021.114770.

[5] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, "Recent advances in image processing techniques for automated leaf pest and disease recognition – a review," *Information Processing in Agriculture*, vol. 8, no. 1, pp. 27–51, 2021.

[6] T. Pham and S. Dao, "Plant leaf disease classification based on feature selection and deep neural network," in *Handbook of Deep Learning in Biomedical Engineering: Techniques and Applications*, 2020, pp. 155–189. doi: 10.1016/B978-0-12-823014-5.00010-7.

[7] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosystems Engineering*, vol. 172, pp. 84–91, 2018, doi: 10.1016/j.biosystemseng.2018.05.013.

[8] M. Nandhini, K. U. Kala, M. Thangadarshini, and S. Madhusudhana Verma, "Deep learning model of sequential image classifier for crop disease detection in plantain tree cultivation," *Computers and Electronics in Agriculture*, vol. 197, 2022, doi: 10.1016/j.compag.2022.106915.

[9] L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by Deep Learning - A Review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.

[10] Z. Gao, Z. Luo, W. Zhang, Z. Lv, and Y. Xu, "Deep learning application in plant stress imaging: a review," *AgriEngineering*, vol. 2, no. 3, pp. 430–446, 2020, doi: 10.3390/agriengineering2030029.

[11] M. E. H. Chowdhury *et al.*, "Automatic and reliable leaf disease detection using deep learning techniques," *AgriEngineering*, vol. 3, no. 2, pp. 294–312, 2021, doi: 10.3390/agriengineering3020020.

[12] R. Sangeetha and M. Mary Shanthi Rani, "A novel method for plant leaf disease classification using deep learning techniques," *Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication*, pp. 631–643, 2021, doi: 10.1007/978-981-16-0289-4_46.

[13] S. Panigrahi, A. Nanda, and T. Swarnkar, "A survey on transfer learning," *Smart Innovation, Systems and Technologies*, vol. 194, pp. 781–789, 2021, doi: 10.1007/978-981-15-5971-6_83.

[14] S. S. Kumar and B. K. Raghavendra, "An efficient approach for coffee leaf disease classification and severity prediction," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 5, pp. 702–716, 2023, doi: 10.22266/ijies2023.1031.59.

[15] G. Geetharamani and J. A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Computers and Electrical Engineering*, vol. 76, pp. 323–338, 2019, doi: 10.1016/j.compeleceng.2019.04.011.

[16] S. Rozlan and M. Hanafi, "Efficacy of chili plant diseases classification using deep learning: a preliminary study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 3, pp. 1442–1449, 2022, doi: 10.11591/ijeecs.v25.i3.pp1442-1449.

[17] W. H. Zeng, H. Li, G. Hu, and D. Liang, "Identification of maize leaf diseases by using the SKPSNet-50 convolutional neural network model," *Sustainable Computing: Informatics and Systems*, vol. 35, 2022, doi: 10.1016/j.suscom.2022.100695.

[18] M. Aparna and B. S. Rao, "A novel automated deep learning approach for Alzheimer's disease classification," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 1, pp. 451–458, 2023, doi: 10.11591/ijai.v12.i1.pp451-458.

[19] 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, doi: 10.1016/j.biosystemseng.2020.03.020.

[20] N. Krishnamoorthy, P. L. V. Narasimha, K. C. S. Pavan, B. Subedi, H. B. Abraha, and V. E. Sathishkumar, "Rice leaf diseases prediction using deep neural networks with transfer learning," *Environmental Research*, vol. 198, 2021, doi: 10.1016/j.envres.2021.111275.

[21] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Deep feature based rice leaf disease identification using support vector machine," *Computers and Electronics in Agriculture*, vol. 175, 2020, doi: 10.1016/j.compag.2020.105527.

[22] M. Deb, K. G. Dhal, R. Mondal, and J. Gálvez, "Paddy disease classification study: a deep convolutional neural network approach," *Optical Memory and Neural Networks (Information Optics)*, vol. 30, no. 4, pp. 338–357, 2021, doi: 10.3103/S1060992X2104007X.

[23] M. A. Saleem, M. Aamir, R. Ibrahim, N. Senan, and T. Alyas, "An optimized convolution neural network architecture for paddy disease classification," *Computers, Materials and Continua*, vol. 71, no. 2, pp. 6053–6067, 2022, doi: 10.32604/cmc.2022.022215.

[24] J. Chen, W. Chen, A. Zeb, S. Yang, and D. Zhang, "Lightweight inception networks for the recognition and detection of rice plant diseases," *IEEE Sensors Journal*, vol. 22, no. 14, pp. 14628–14638, 2022, doi: 10.1109/JSEN.2022.3182304.

[25] P. Joshi, D. Das, V. Udutalapally, M. K. Pradhan, and S. Misra, "RiceBioS: identification of biotic stress in rice crops using edge-as-a-service," *IEEE Sensors Journal*, vol. 22, no. 5, pp. 4616–4624, 2022, doi: 10.1109/JSEN.2022.3143950.

[26] S. Nalini *et al.*, "Paddy leaf disease detection using an optimized deep neural network," *Computers, Materials and Continua*, vol. 68, no. 1, pp. 1117–1128, 2021, doi: 10.32604/cmc.2021.012431.

[27] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation," *Information Processing in Agriculture*, vol. 7, no. 4, pp. 566–574, 2020, doi: 10.1016/j.inpa.2019.11.001.

[28] P. K. Sethy, "Rice leaf disease image samples," *Mendeley Data, ver. 1*, 2020, doi: 10.17632/fwcj7stb8r.1.

[29] L. A. Ruiz, "Paddy doctor dataset LR," *Kaggle*. 2022. [Online]. Available: https://www.kaggle.com/datasets/leonardoaruizv/paddy-doctor-dataset-lr

BIOGRAPHIES OF AUTHORS

Parthasarathi Jayaraman hold a master's degree in computer science. At present, he is pursuing Ph.D. in the Department of Computer Science, SRM Institute of Science and Technology, Kattankulatur, Chennai. His research focuses on deep learning. He can be contacted at email: sarathiresearch@gmail.com.

Muthulakshmi Palaniyandi hold Ph.D. and a master's degree in computer science. She is a Professor at the SRM Institute of Science and Technology who works in the Department of Computer Science. Her research focuses on performance analysis in distributed and parallel computing, cyber physical systems. She can be contacted at email: muthulap@srmist.edu.in.