IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 1, February 2026, pp. 756~765
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i1.pp756-765

Intelligent plant disease detection using twin attention optimal convolutional neural network

Prameetha Pai[1], Namitha S. J.[1], Sowmya T.[1], Amutha S.[2], Nisarga Gondi[1]
[1]Department of Computer Science and Engineering, B.M.S. College of Engineering, Bengaluru, India
[2]Department of Computer Science and Engineering, Dayananda Sagar College of Engineering, Bengaluru, India

Article Info

Article history:

Received Dec 14, 2024
Revised Nov 17, 2025
Accepted Dec 15, 2025

Keywords:

Chaotic slime mould optimization
Deep learning
Improved Gaussian filtering
Neural network
Rice plant leaf disease
Twin attention-convolutional neural network

ABSTRACT

Farming is one of the most important ways for people in India to make a living. Rice is a staple food, and when farmers successfully harvest rice crops, pests often attack them, which costs agriculture a lot of money. There are now a lot of new AI-based ways to help with this problem in rice plants. But those ways don't work very well because they take a long time and make mistakes when sorting things. This article talks about a new hybrid deep learning (DL) method for finding leaf diseases in rice plants. This process has four main steps: pre-processing, segmentation, feature extraction, and classification. A hybrid DL-based twin attention convolutional neural network (CNN) model classifies segmented images into healthy and unhealthy leaves. But this method has the problem of overfitting. An optimization method based on chaotic slime mould (CSM) solves this problem. The proposed method is compared with bidirectional long short-term memory (Bi-LSTM), recurrent neural network (RNN), deep neural network (DNN), and deep belief network (DBN). The suggested method has an overall accuracy of 99.56%, an F-measure of 99.21%, a sensitivity of 99.16%, a specificity of 98.56%, a precision of 99.26%, a mean absolute error (MAE) of 0.004, a mean squared error (MSE) of 0.004, and a root mean square error (RMSE) of 0.06.

Corresponding Author:

Sowmya T.
Department of Computer Science and Engineering, B.M.S. College of Engineering
Bull Temple Road, Basavangudi, Bengaluru 19, Karnataka, India
Email: sowmyat.cse@bmsce.ac.in

1. INTRODUCTION

At first, a specialist did plant disease monitoring and analysis by hand [1]–[5]. This takes a lot of time and work to get through. The signs of disease can usually be seen on the fruits, leaves, and stems. The plant leaf shows the signs of the disease, which is how the illness is found [6], [7]. It is hard to figure out what's wrong with a plant. One needs to look at the leaves and see what they look like. Because of this complexity and the many plants that are grown and their current phyto-pathological problems, even experienced agronomists and plant pathologists often miss some diseases. This results in erroneous conclusions and treatments [8], [9]. An automated computational system for identifying and diagnosing plant diseases would greatly assist agronomists tasked with such diagnoses through leaf examination [10]. Also, the technology could be used with self-driving farm vehicles on big farms to quickly and accurately find plant diseases in the fields by taking pictures all the time [11].

Researchers, especially those from developing countries, are increasingly concentrating on the identification of plant diseases. Another area of research that looks promising is finding plant diseases early [12]. Bacteria, fungi, and viruses are just a few of the things that can make plants sick. These diseases can also hurt the world economy, society, and environment. Because of this, there is a need to find quick and accurate ways to find plant diseases right away. In the past, different ways have been used to find and stop plant diseases in order to cut down on crop losses [13].

Bacteria, viruses, and fungi are some of the most common things that hurt crops. This can be stopped with plant disease detection systems [14]. The farmer picks the right crop based on the weather, the soil type, and how much money it will make. Agricultural businesses started looking for new ways to grow more food because the weather was changing, the population was growing, and politics were unstable. So, scientists are looking for new technologies that are both precise and efficient to boost productivity [15], [16]. Farmers can use precision agriculture in IT to get the data and information they need to make the best choices for getting the most crops. AI applications related to machine learning (ML) have grown a lot in the last few years because of the growth of computational systems, especially graphical processing units (GPU), which leads to the creation of new methods and models like deep learning (DL) [17]. DL employs artificial neural networks (ANN) characterized by multiple processing layers, distinguishing it from the "swallower" topologies utilized in more prevalent neural network methodologies [18]. DL methods in agriculture, especially for figuring out what diseases' plants have, are still new and haven't been around for very long [19], [20].

It takes a long time and is boring to figure out who someone is by looking at them. A computerized screening system was created to make things easier at work. This system automatically checks the plants' health and looks for diseases. The main goal of proposed method is to show a new way to use DL to classify rice plant diseases into healthy and unhealthy groups using twin attention-convolutional neural network (TA-CNN). It also suggests way to optimize using chaotic slime mould (CSM) to reduce loss and overfitting.

A thorough and comprehensive examination of the literature regarding contemporary systems was conducted. The survey was done to find out what the current efforts can and can't do, what the problems are, and what has worked. Table 1 shows some of the most important systems that were studied, along with their problems and unique features. These systems are the most important to the objectives of this work.

Table 1. Comparison of existing work

References	Methods	Features	Challenges
Hassan *et al.* [21]	CNN	The time for training process is less	Sometimes the images have noise and leads to misclassification
Ferentinos [22]	CNN	This model has achieved an accuracy of 99.53% and it was more robust in real life application	This model does not identify the existing plant disease.
Kamal [23]	Depth wise separable convolution	This model showed efficient trade-off among accuracy and latency	When the parameters were increased, the cost of computation also increased. Takes more time for large data
Ashtagi *et al.* [24]	Hybrid CNN+RF, CNN+SVM with PSO	Hybrid fusion of DL and ML improves classification accuracy; PSO-based feature selection enhances discriminative power. CNN+RF achieved 95% accuracy, CNN+SVM achieved 93% accuracy	Higher computational complexity; requires careful parameter tuning and quality datasets for optimal generalization
Atila *et al.* [25]	EfficientNet	Achieved better accuracy	Execution time is large when compared to the baseline models.
Panigrahi *et al.* [26]	ML models	This model was more helpful for formers to find the diseases at early stage	This model may not be suitable for all database

2. METHOD

DL is moving quickly, so it can now find plant diseases on it is own. This helps in finding sick plants and figure out what's wrong with them by looking at digital photos. This work shows a DL system that can automatically tell difference between pictures of healthy and sick plant leaves. The first step in getting an image ready for processing is to get rid of noise, add to it, and change it is size. The improved Gaussian filter (IGF) gets rid of the noise in the picture. This makes the colors stand out more and makes the picture clearer. Data can be added to the picture by rotating, clipping, and flipping it to obtain as many pictures as possible after the noise is gone. Next, the image needs to be resized so that it works with the DL model. To get a high level of disease detection accuracy, it is important to break the input image into smaller parts before extracting the features. To do this, binary threshold segmentation (BTS) is used to separate the image of the leaf from the background. Next, feature extraction is discussed, which is a very important part of identifying plant diseases. For this step, a red green blue (RGB) histogram and a grey level co-occurrence matrix (GLCM) are used. The last step is classification process, which uses the twin attention optimal convolutional neural network (TAO-CNN) to figure out if the plant leaf is sick or healthy. The CSM algorithm is used here to fix the TAO-CNN's loss. The TAO-CNN model that was suggested is a good way to do DL that helps with overfitting and makes the classification more accurate. These steps are represented in Figure 1.

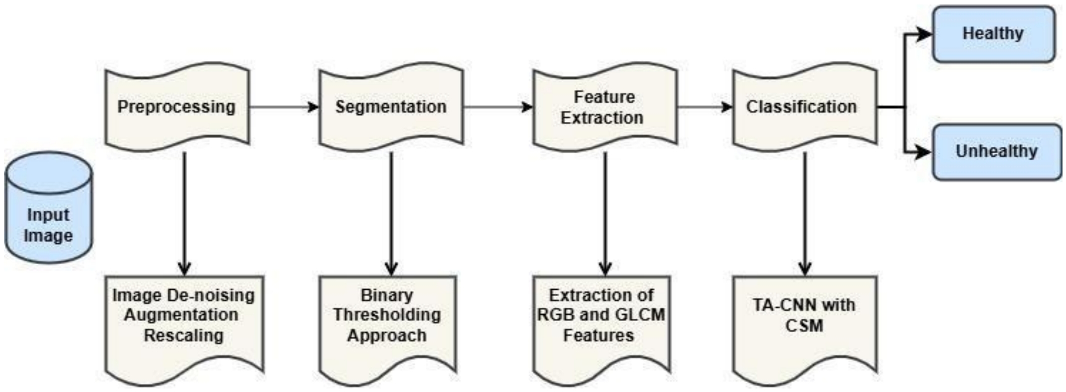


Figure 1. Workflow of the proposed method

2.1. Pre-processing

The raw dataset images have a lot of noise in them, which makes them less accurate. It is necessary to get rid of the noise. People have used Gaussian filtering [27], but it has a lot of false positives in grey values (0–255) and mistakes in pixels that have had noise reduced, which makes it hard to restore images. This problem is solved by the IGF method. This method can check the noise density and use image thresholding to make mistakes less likely. It can also make the window bigger to help find things. Let's look at the picture with the pixel P(x,y) and the grey level G(x,y). The IGF looks at the grey levels of the pixels to decide which ones are noisy. If G(x,y) equals 0 or 255, IGF believes that pixel is noisy. IGF doesn't think it is the noisy pixel if G(x,y) isn't 0 or 255. But in many cases, the suggested IGF method gets the grey level of 0 to 255 wrong and thinks they are noise-free pixels, which makes the picture look blurry. This also makes it hard to tell how loud the noise is, which causes a lot of false alarms. The GF has a better variance component that makes the noise removal process work better. The IGF method uses a fixed weighted mean filtering window to find all the noisy pixels. GF chooses the weights based on which one has more variation. Here is the complete list of steps to get rid of noise:

Let's say that α is the w×w filtering window that is put at P(x,y) for the noise pixel P(x,y). Make w equal to 9. Find the new set V after taking out the noise pixels in α that have pixel values between 0 and 255. The grey level R(x,y) of P(x,y) that was found finds the weighted pixel mean in V. This can be expressed mathematically as:

$$R(x,y) = \frac{\sum_{P(u,v)\in V} \beta(u,v) G(u,v)}{\sum_{P(u,v)\in V} \beta(u,v)} \tag{1}$$

For this case, G(u,v) and β(u,v) are the grey level and weight of P(x,y). The enhanced Gaussian function is represented mathematically by the weight β(u,v), which is:

$$\beta(u,v) = exp\left(-\frac{(u-x)^2+(v-y)^2}{2\sigma^2}\right) \tag{2}$$

In this case, (x,y) and (u,v) are the coordinates for the pixels P(x,y) and P(u,v) that are next to each other in the whole picture. The parameter σ can also be considered as:

$$\sigma = 0.2 + densityofnoise \tag{3}$$

The noise density is the number of pixels that make up the noise divided by the total number of pixels in the image. Data augmentation helps a lot with overfitting problems and makes accuracy better. During data augmentation, the image with the noise removed is randomly moved along the horizontal and vertical axes by a value between −45 and 45. The enhanced Gaussian filter works well to get rid of noise by figuring out how dense it is after these preprocessing steps. Then, thresholding is used to get rid of all but the most important pixel data. So, the pictures that come out are clearer. Images that were pre-processed this way work much better in later steps, such as classification, segmentation, and detection.

2.2. Segmentation using binary thresholding approach

After that, the image goes to the segmentation stage, where the sick part is cut out of the background image. The binary thresholding method (BTS) is employed to perform the segmentation. By setting threshold value, the BTS method can remove unwanted background regions. At first, the pixel in picture that is below the threshold gets a value of 0. A pixel in an image gets a value of 1 if it is value is higher than the threshold.

The best threshold for each image is different because the pixel distribution for each pixel value is different. If all of the test images have the same threshold, the random threshold might not hurt the performance. The line search method cannot identify the best threshold for each image due to the unknown image types in the testing set. This means that labelled data for training, validation, and testing are not available. During training, this problem can be mitigated by searching for the best value.

2.3. Feature extraction

The process of extracting features has two main steps: getting color features and getting GLCM features. To get the important red (R), green (G), and blue (B) color features from the picture of the sick rice plant, an RGB histogram is used. This RGB histogram is great because it is used a lot in TVs, monitors, cameras, and other devices to tell the difference between channels. One of the words used to get leaf color features from the input samples is "RGB histogram". In most of the input samples, the blue channel has fewer features than the red channel. A histogram is a picture made up of pixels that have the same frequency. It helps identify pixels that are different at each feature point. The most important thing the histogram does is get colors from the RGB channels. During extracting features, think of RGB as the hard part that the histogram is broken up into different bins. RGB always has 257 channels, and each RGB cumulative vector has 768 bins.

2.4. Classification using twin attention-convolutional neural network technique

Next, the features that were removed are sent to the classification stage, where they are sorted into pictures of the sick rice plant that are healthy and unhealthy. There are blocks for both spatial and spectral attention in the proposed twin attention architecture. The input patch, which is 11×11×13, is split into two streams so that the deep features can be pulled out of the input samples. This is how the suggested TA-CNN model is set up. The architecture includes a surface multi-layer feature extractor, a max pooling layer, spectral attention blocks, and multi-residual blocks. To start, multi-layer convolution blocks are used to get rid of the deep features that are based on the surface. The feature can be observed in the middle channels with the spectral attention block. The max pooling layer also shrinks the size of the features. The multi-residual block is used to find the most important parts. The last step is when the multi-residual blocks and spectral attention blocks get the deep features that were taken out so they can get hierarchical deep features. In the second stream, spatial attention blocks are used to look for deep spatial features by looking at deep features. The spectral attention block and the max pooling layer are built after the convolution layer. The dense layer gets the flattened features, and the flatten layer flattens the deep features that were taken out. The SoftMax layer then finds out what the final answer is. Figure 2 shows the whole structure of the proposed TA-CNN.

2.4.1. Twin attention mechanism

The CNN's attention block is based on how the human brain uses psychological AM to find important deep features instead of ones that aren't useful. The suggested AM works by using spectral-temporal and spatial-temporal data from input samples to sort rice diseases. The input feature map goes to the convolution block. The kernel size (x,y) is almost the same as the input feature's length and width. The outcome feature map is 1×1×f, where f is the number of filters. The output from the current layer is changed and sent to the multi-layer perceptron (MLP). The MLP has two dense layers with neurons of different sizes.

There are a lot fewer neurons in the first and second layers because of the new features. Before the input feature map is multiplied with the extracted features, a separable convolution layer is applied to each input data. The separable convolution layer multiplies the first stream's output. To do this, point-wise and depth-wise convolution are used. The proposed spatial AM is defined by feature maps that show a middle ground relationship. It is the same as the spectral AM because it only keeps the most important parts of the input feature map. After that, the input feature map is turned into a convolution block with a kernel size of (x,y) and only one kernel for convolution and padding. The outcome feature map has a size of x×y×1. The MLP gets the new result from the layer that is currently active. It has two fully convoluted (FC) layers, each with neurons of different sizes. There are a lot fewer neurons in the first layer than there were before. The second FC layer changes how all the features look, and the separable convolution layer is used on each input data before the input feature map and extracted features are multiplied.

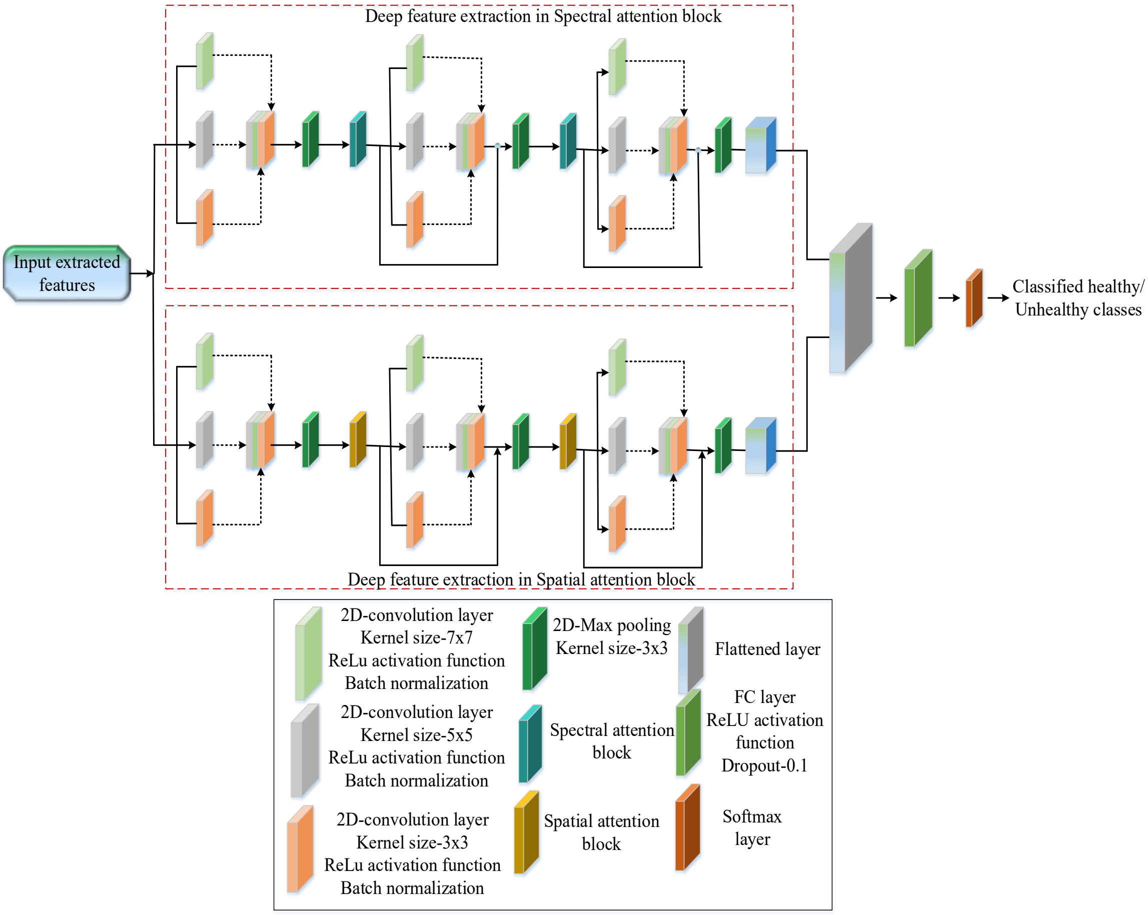


Figure 2. Architecture of the proposed twin attention-convolutional neural network mode

2.4.2. Convolution layer

The convolution layer is the most important part of a CNN network for getting hierarchical features from the input samples. The convolution layer automatically splits the spatial and spectral features. Mathematically, the convolution layer's main job can be understood as:

$$C_M = \beta(w^p C_{M-1}) + B_M \tag{4}$$

In this case, β is the activation function, w is the weighted templates, BM is the bias vector, and M is the total number of layers. The output of the lth feature map at the spatial coordinates (r,s) can be expressed as:

$$C^{rs}{}_{M,l} = \beta\left(B_{M,l} + \sum_c \sum_{x=0}^{X_M-1} \sum_{y=0}^{X_M-1} w^{x,y}{}_{M,l} C^{(r+x)(s+y)}{}_{M-1,c}\right) \tag{5}$$

This time, C is the feature cube that was combined with the current feature cube and is the (M–1)th layer. The filter's length is X, and the filter's width is Y. Using the residual and multiple scaling blocks can make the network model work better and help a lot to stop gradient insufficiency.

When a rice plant loses it is function, it's hard to tell what kind of disease it has. The multi-feature system works well because the TA-CNN loss function is lower. The loss function always looks at the negative frames, which lowers the true positive rate. During training, there will be a small difference in how the whole network is handled. This imbalance causes the extra factor to cross the original parameter that has gone bad. The loss can beat the gradient of images that aren't evenly spaced.

2.4.3. Loss function reduction using chaotic slime mould optimization technique

The losses in TA-CNN make accuracy performance much worse. That's why this study suggests the CSM optimization method. The standard slime mould (SM) method [28] depends on how SM moves back and forth. A mathematical study finds the best way to find food that has a lot of room to explore and use. But this method takes longer to reach a solution and is more likely to get stuck in local optima. Combining chaotic mapping with SM optimization speeds up the search and makes better use of localized solutions.

3. RESULTS AND DISCUSSION

This study focusses on disease classification using an Indian dataset [29] collected from Gandhinagar, Gujarat, containing around 120 unhealthy classes. Because the dataset only has unhealthy classes, a publicly available rice leaf dataset [30] with 4,000 healthy classes was made. The classification process uses 4,120 classes in the end. Some of these classes are healthy, and some are not. The hyperparameters of the proposed method are shown in Table 2.

Table 2. Hyper parameters of the proposed method

Hyper parameters	Values
Optimizer	CSM optimizer
Learning rate	0.001
Epochs	100
Loss function	Binary cross entropy
Batch size	60
Dropout	0.3

The proposed method's confusion matrix is shown in Figure 3. A total of 116 samples is examined for the healthy categories. There are 115 samples that are correctly identified as healthy classes and one that is incorrectly identified as an unhealthy class. There are 115 samples in the unhealthy classes. All 115 samples are correctly put into the unhealthy classes in this case.

Figure 4 shows how long it takes for each method to work. The graph shows that the proposed method takes less time than current methods. The current bidirectional long short-term memory (Bi-LSTM), recurrent neural network (RNN), deep neural network (DNN), and proposed model take 401.23, 421.36, 399.41, 413.52, and 251.32 ms, respectively. The current methods take a long time because they don't work well with big datasets when they're being trained. The proposed hybrid method yields intriguing results with minimal time complexity. It also shows performance that is almost the same when the amount of training data increases.

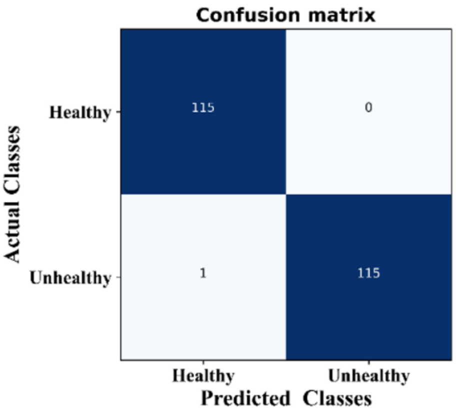


Figure 3. Confusion matrix of the proposed method

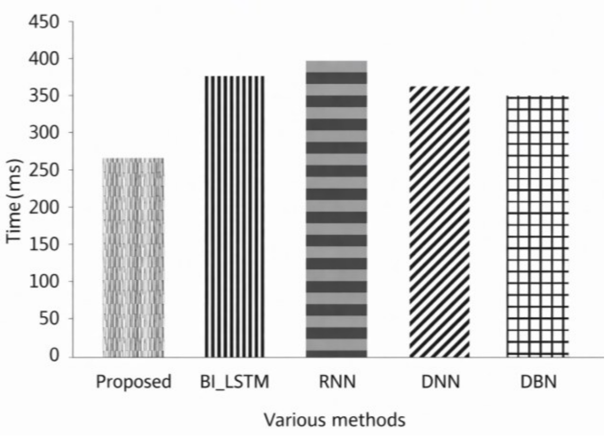
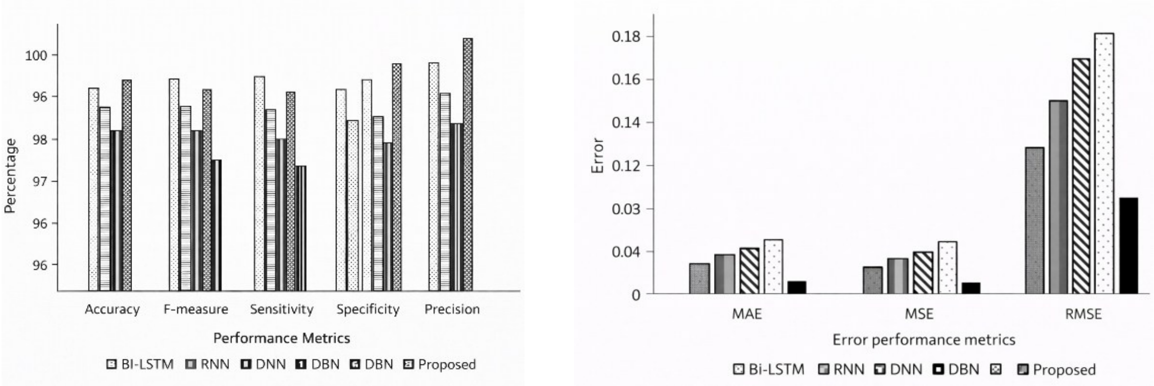


Figure 4. Time complexity under different methods

Figures 5 show how well the predictions and the mistakes worked. Figure 5(a) shows the prediction-based performance, while Figure 5(b) shows the error performances. The results show that the suggested method works well to tell the difference between healthy and unhealthy groups. Table 3 shows how well different methods worked by using performance metrics, and Table 4 shows how well different metrics worked with different batch sizes.



(a) (b)

Figure 5. Performance comparison under: (a) prediction-based performance and (b) error performances

Table 3. Obtained outcome of different techniques under performance metrics

Metrics	Methods				
	Bi-LSTM	RNN	DNN	DBN	Proposed
Accuracy	99.13	98.26	97.40	96.53	99.56
F-measure	98.79	97.92	97.05	96.21	99.21
Sensitivity	98.73	97.87	97	96.14	99.16
Specificity	98.14	97.27	96.41	95.58	98.56
Precision	98.84	97.97	97.1	96.28	99.26
Mean absolute error (MAE)	0.0086	0.017	0.025	0.034	0.0043
Mean squared error (MSE)	0.0086	0.017	0.025	0.034	0.0043
Root mean square error (RMSE)	0.093	0.13	0.16	0.18	0.065

Table 4. Comparative analysis of the batch size under different metrics

Models	Batch size	Accuracy (%)	F1 score (%)	Precision (%)	Recall (%)	Specificity (%)
Proposed	20	96.5	96.32	96.3	96.56	96.7
	40	97.3	97	96.5	97.3	97.15
	60	99.3	98.45	98.35	98.5	99.01
	80	98.3	97.3	97.13	97	98.3
	100	97.9	96.21	96.8	96.3	97.6
BI-LSTM	20	95.9	94.4	95.8	94.7	94.3
	40	95.6	95.11	95.21	96.1	95.2
	60	97.3	96.52	97.15	97.75	95.35
	80	95.2	94.2	94.24	94.14	94.5
	100	94	93.6	94.12	93.2	92.6
RNN	20	93.8	95.6	93.5	93.7	94.6
	40	94.3	96.4	95.14	94.4	95.4
	60	95.6	97.36	95.26	95.16	96.16
	80	94.2	94.4	94	94.4	94.4
	100	93.5	92.14	92.4	92.8	93.3
DNN	20	92.5	93.36	92.2	92.5	93.4
	40	93.2	94.5	93.5	93.6	94.4
	60	95.2	96.2	95.8	94.5	95.51
	80	94.6	93.16	94.16	92.16	93.16
	100	93.3	93.28	94.23	92	92.45

4. CONCLUSION

The existence of an automated computational system for detecting rice plant diseases is significantly influenced by fluctuating environmental conditions and temporal complexities. This study presents an innovative hybrid DL methodology for the classification of rice plant leaf diseases. The suggested method has an overall accuracy of 99.56%, an F-measure of 99.21%, a sensitivity of 99.16%, a specificity of 98.56%, a precision of 99.26%, a MAE of 0.004, a MSE of 0.004, and a RMSE of 0.06. The proposed method is very helpful for finding diseases in many crops, and it doesn't take long. But the suggested method doesn't say how bad the disease is for the rice crop. The researchers need to find better ways to figure out how bad the disease is that affects the leaves of rice plants in the future. The proposed work needs to be better in order to find more plant diseases and see how well the model works.

FUNDING INFORMATION

The authors assert that they did not receive any financial compensation, grants, or assistance during the composition of this paper.

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
Prameetha Pai	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Namitha S. J.	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		
Sowmya T.	✓		✓	✓		✓		✓	✓	✓	✓		✓	
Amutha S.	✓	✓		✓		✓	✓		✓			✓		
Nisarga Gondi	✓		✓		✓		✓	✓		✓		✓		

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

The authors claim that there are no conflicts of interest.

INFORMED CONSENT

This study did not involve individuals or any personal identification information requiring informed consent.

ETHICAL APPROVAL

This paper does not include individuals or animals; there has been no research involving human subjects. The authors did not seek approval from any institutional review board.

DATA AVAILABILITY

The data that support the results of this study were gathered from publicly available repositories and used to train and test the suggested TAO-CNN for detecting plant diseases. The PlantVillage dataset, which is publicly available, was used in this study. To enhance model's performance, additional data were incorporated, image sizes were adjusted, and the dataset was standardized. The corresponding author, [PP], can provide all scripts used for data preparation and model training, along with trained model weights and evaluation results, upon request. These materials are intended solely for research that is not for school or business.

REFERENCES

[1] S. K. Upadhyay and A. Kumar, "A novel approach for rice plant diseases classification with deep convolutional neural network," *International Journal of Information Technology*, vol. 14, no. 1, pp. 185–199, Feb. 2022, doi: 10.1007/s41870-021-00817-5.

[2] Y. Borhani, J. Khoramdel, and E. Najafi, "A deep learning based approach for automated plant disease classification using vision transformer," *Scientific Reports*, vol. 12, no. 1, Jul. 2022, doi: 10.1038/s41598-022-15163-0.

[3] V. K. Shrivastava, M. K. Pradhan, and M. P. Thakur, "Application of pre-trained deep convolutional neural networks for rice plant disease classification," in *Proceedings-International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, 2021, pp. 1023–1030, doi: 10.1109/ICAIS50930.2021.9395813.

[4] M. Sharma, C. J. Kumar, and A. Deka, "Early diagnosis of rice plant disease using machine learning techniques," *Archives of Phytopathology and Plant Protection*, vol. 55, no. 3, pp. 259–283, 2022, doi: 10.1080/03235408.2021.2015866.

[5] T. S. Poornappriya and R. Gopinath, "Rice plant disease identification using artificial intelligence approaches," *International Journal of Electrical Engineering and Technology*, vol. 11, no. 10, pp. 392–402, 2020, doi: 10.17605/OSF.IO/R5JXA.

[6] S. Babu, M. Maravarman, and R. Pitchai, "Detection of rice plant disease using deep learning techniques," *Journal of Mobile Multimedia*, vol. 18, no. 3, pp. 757–770, 2022, doi: 10.13052/jmm1550-4646.18314.

[7] 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.

[8] J. Chen, D. Zhang, A. Zeb, and Y. A. Nanehkaran, "Identification of rice plant diseases using lightweight attention networks," *Expert Systems with Applications*, vol. 169, 2021, doi: 10.1016/j.eswa.2020.114514.

[9] 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.

[10] T. Tosawadi, T. Kasetkasem, W. Laungnarutai, T. Phatrapomnant, and I. Kumazawa, "Automatic rice plant disease evaluation method based on anomaly detection and deep learning," in *ECTI-CON 2021-2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology: Smart Electrical System and Technology*, 2021, pp. 900–903, doi: 10.1109/ECTI-CON51831.2021.9454737.

[11] K. Kumar K and E. Kannan, "Detection of rice plant disease using AdaBoostSVM classifier," *Agronomy Journal*, vol. 114, no. 4, pp. 2213–2229, 2022, doi: 10.1002/agj2.21070.

[12] N. Krishnamoorthy, L. V. N. Prasad, C. S. P. Kumar, 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.

[13] B. R. Pushpa, A. Ashok, and A. V. S. Hari, "Plant disease detection and classification using deep learning model," in *Proceedings of the 3rd International Conference on Inventive Research in Computing Applications, ICIRCA 2021*, 2021, pp. 1285–1291, doi: 10.1109/ICIRCA51532.2021.9544729.

[14] B. Patel and A. Sharaff, "Rice crop disease prediction using machine learning technique," *International Journal of Agricultural and Environmental Information Systems*, vol. 12, no. 4, 2021, doi: 10.4018/IJAEIS.20211001.oa5.

[15] B. S. Bari *et al.*, "A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework," *PeerJ Computer Science*, vol. 7, 2021, doi: 10.7717/PEERJ-CS.432.

[16] R. Sowmyalakshmi *et al.*, "An optimal classification model for rice plant disease detection," *Computers, Materials and Continua*, vol. 68, no. 2, pp. 1751–1767, 2021, doi: 10.32604/cmc.2021.016825.

[17] R. Sharma, V. Kukreja, and V. Kadyan, "Hispa rice disease classification using convolutional neural network," in *2021 3rd International Conference on Signal Processing and Communication, ICPSC 2021*, 2021, pp. 377–381, doi: 10.1109/ICSPC51351.2021.9451800.

[18] K. K. Jena, S. K. Bhoi, D. Mohapatra, C. Mallick, and P. Swain, "Rice disease classification using supervised machine learning approach," in *5th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2021*, 2021, pp. 328–333, doi: 10.1109/I-SMAC52330.2021.9641054.

[19] S. Aggarwal *et al.*, "Rice disease detection using artificial intelligence and machine learning techniques to improvise agro-business," *Scientific Programming*, vol. 2022, no. 1, 2022, doi: 10.1155/2022/1757888.

[20] T. Daniya and S. Vigneshwari, "Amalgamation of rice plant disease classification and detection with KNN and butterfly algorithm," *Innovations in Computer Science and Engineering*, 2022, pp. 127–135, doi: 10.1007/978-981-16-8987-1_14.

[21] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of plant-leaf diseases using CNN and transfer-learning approach," *Electronics*, vol. 10, no. 12, 2021, doi: 10.3390/electronics10121388.

[22] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[23] K.C. Kamal, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Computers and Electronics in Agriculture*, vol. 165, 2019, doi: 10.1016/j.compag.2019.104948.

[24] R. Ashtagi, S. M. Jaybhaye, S. Mohite, V. Musale, S. Chinchmalatpure, and R. V. Bidwe, "Fusion of AI techniques: a hybrid approach for precise plant leaf disease classification," *Journal of Electrical Systems*, vol. 20, no. 1s, pp. 850–861, 2024, doi: 10.52783/jes.836.

[25] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using efficientnet deep learning model," *Ecological Informatics*, vol. 61, 2021, doi: 10.1016/j.ecoinf.2020.101182.

[26] K. P. Panigrahi, H. Das, A. K. Sahoo, and S. C. Moharana, "Maize leaf disease detection and classification using machine learning algorithms," *Advances in Intelligent Systems and Computing*, vol. 1119, pp. 659–669, 2020, doi: 10.1007/978-981-15-2414-1_66.

[27] J. Khearallah and K. Rijab, "Optimizing the performance of wireless sensor network based on software defined network and gaussian filter," *Engineering and Technology Journal*, vol. 40, no. 2, pp. 379–385, 2022, doi: 10.30684/etj.v40i2.2221.

[28] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020, doi: 10.1016/j.future.2020.03.055.

[29] vbookshelf, "Rice leaf diseases dataset," *Kaggle*, Accessed: Feb, 12. 2024. [Online]. Available: https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases

[30] Bahri, "Rice-leaf," *Kaggle,* Accessed: Feb, 12. 2024. [Online]. Available: https://www.kaggle.com/datasets/bahribahri/riceleaf

BIOGRAPHIES OF AUTHORS

Prameetha Pai holds a master's degree from B.M.S. College of Engineering, Bangalore and a bachelor's degree from NMAM Institute of Technology, Karkala. She is also pursuing her research at Dayananda Sagar College of Engineering Bangalore which is affiliated to VTU Belgaum. She is currently working as assistant professor in the Department of Computer Science and Engineering, B.M.S College of Engineering, Bangalore. She has over 12 years of teaching experience. She has about 10 publications in national and international journal. Her area of research is precision agriculture and deep learning. She can be contacted at email: prameethapai.cse@bmsce.ac.in.

Namitha S. J. holds Master of Technology in Computer Networking and Engineering and a Bachelor of Engineering (B.E.) in Computer Science and Engineering, She is currently working as assistant professor in the department of Computer Science and Engineering, B.M.S College of engineering, Bangalore, with over 12 years of teaching experience, her focused researched fields are computer networks, mobile application development, image processing, graph theory, active member of The Institution of Engineers (India) (IEI). She has an excellent knowledge in user interface design techniques for a good web design. She can be contacted at email: namithasj.cse@bmsce.ac.in.

Sowmya T. holds a master's degree from Bangalore Institute of Technology, Bangalore and a bachelor's degree from Bangalore Institute of Technology, Bangalore. She is also pursuing her Ph.D. at BMS College of Engineering Bangalore which is affiliated with VTU Belgaum. Her research interests are cyber security, machine learning, and IoT. She has industry experience of 3 years and teaching experience of 10 years and is currently working as assistant professor in the Department of Computer Science and Engineering, B.M.S. College of Engineering. She can be contacted at email: sowmyat.cse@bmsce.ac.in.

Amutha S. is professor of Department of Computer Science and Engineering, Dayananda Sagar College of Engineering, Bengaluru. She obtained her Ph.D. degree from VTU University. She has more than 26 years of teaching experience. She has published in 27 journals and conferences. Her research interests include data analytics, computer vision, and cloud computing. She can be contacted at email: amutha-cs@dayanandasagar.edu.

Nisarga Gondi holds a bachelor degree from BMS College of Engineering, Bangalore and is pursuing master's degree from Carnegie Mellon University. His research interests are machine learning, deep learning and information security. He has one publication in IEEE conference. He can be contacted at email: nisargagondi@gmail.com.