Computational Engineering and Technology Innovations

www.ceti.reapress.com

Comput. Eng. Technol. Innov. Vol. 2, No. 4 (2025) 308–326.

Paper Type: Original Article

Optimized Deep Spectral Generative Adversarial Neural Network (DSGAN2) Framework for Efficient Real-Time Detection of Automatic Rice Leaf Disease

Seyyed Ali Noorkhah[1,*], Zhang Hao[2]

[1] Department of Civil Engineering, Science and Research Branch, Islamic Azad University, 1477893855 Tehran, Iran; a.noourkhah@aihe.ac.ir.

[2] Institute of Education, Guizhou Normal University, Guizhou Province, China; haozhang_guizhou@sina.com.

Citation:

Received: 02 March 2025 *Revised: 12 April 2025* *Accepted: 29 September 2025*	Noorkhah, S. A., & Hao, Zh. (2025). Optimized deep spectral generative adversarial neural network (DSGAN2) framework for efficient real-time detection of automatic rice leaf disease. *Computational engineering and technology innovations, 3*(4), 308–326.

Abstract

Automatic methods which apply computer vision and machine learning to identify rice diseases exists in leaf images, may offer some considerable benefits rather than the manual visual examination methods. In order to detect the rice plant diseases, deep learning techniques are employed, as it supports in lessening the burden of the farmers in securing the crops. Current models are not appropriate for real-time large-scale applications which offer low-resource rural situations; because these current models are often high computational. In this paper we introduced an optimized Deep Spectral Generative Adversarial Neural Network (DSGAN^2) to help in improvement of accuracy and reduction in computational complexity. We performed image pre-processing along with gaussian filter or median filter for noise reduction. Moreover, bilinear interpolation is used to resize the image, efficiently. Most relevant features are effectively preserved using feature selection along with the recursive feature elimination. A simplified U-Net model is applied to facilitate an accurate localization; which goes over the excessive resource demand. Feature selection can be additionally optimized by the Binary Particle Swarm Optimization (BPSO). Effective convergence on optimal weights is attained by the BPSO. As we can see from the obtained classification results, False Positive (FP) and False Negative (FN) are minimized, and high accuracy is ensured by an optimised DSGAN^2, in low-resource conditions**.** Considerable reductions in error rate at 40.8%, APS-DCCNN at 55.2%, Alex Net at 50.4%, and standard Convolutional Neural Networks (CNN) at 49.5% are obtained by the DSGAN^2, compared to the current methods. The proposed method brings out an effective robust method for real-time disease detection in agriculture.

Keywords: Deep Spectral GAN, Smart Agriculture, Principal Component Analysis, Binary Particle Swarm Optimization, Segmentation, U-Net.

1 | Introduction

A branch of Artificial Intelligence called deep learning models. This deep learning models utilizes the multi-layered neural networks, as it helps in identifying the complex patterns in large datasets. It is usually applied in Generative Adversarial Networks (GANs), here, a discriminator and generative neural networks are

Corresponding Author: ja.noourkhah@aihe.ac.ir

https://doi.org/10.48314/ceti.vi.79

Licensee System Analytics. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0).

compared to one another [1], [2]. The reliability of the data samples generated by the generative are evaluated by the discriminator for facilitating constant network improvement. A realistic, high-quality data was generated by the Image Processing tasks, because GAN is effective in IP tasks. On the other hand, this GAN is ineffective for areas like Medical Imaging, natural scene synthesis, and agricultural analysis. For disease detection and disease classification, a unique method was offered by GAN in smart agriculture [3]. Diseased leaves image variations were effectively generated by this approach. In case of limited datasets, the deep learning model training was also enhanced by this method.

An autonomous plant disease detection in agricultural applications has been revolutionized by the deep learning. For image classification, the CNN are usually employed especially for plant disease detection [4]. The visual symptoms like colour changes or leaf lesions were analyzed, as it helps the CNN-based models for effective disease detection and classification of disease. In case of large-scale monitoring, manual monitoring is difficult, so this CNN-based models are highly beneficial. In disease detection, high accuracy was attained by the deep learning models, as it utilizes the pre-processed images of infected leaves. Timely interventions were also facilitated [5]. But, especially in low-resource settings, the application of these models in RT was limited by the demand of large computational resources. Leaf disease usually affects the rice, a stable crop. The production, and food security is greatly impacted by the leaf diseases [6]. Common rice diseases, including Brown Spot, Leaf Blast, and Hispa, are particularly problematic. Photosynthetic activity is reduced, and small brown lesions on leaves is occurred by the brown spot. Then, the elliptical grey spots indicate the leaf blast, finally it will destroy the plant. The beetle infestation causes hisps. the white streaks are formed by the beetle infestation, and the beetles feed on leaf tissues are indicated in *Fig. 1* [7]. The crop production, and crop quality are minimized by these diseases. A direct threat to agricultural economics and food security was posed by these diseases. For effective disease management in rice farming, need for timely and accurate detection is crucial because the scale and frequency of those problems.

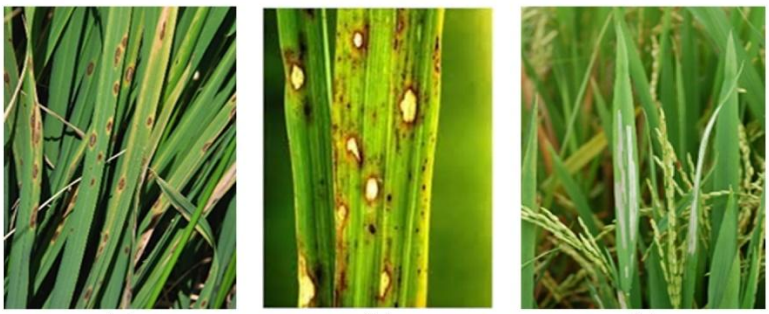


Fig. 1. Common rice leaf diseases: a. Brown spot, b. Leaf blast, and c. Hispa.

Several advantages were offered by implementing deep learning in plant disease detection. Automation, high accuracy, and scalability were some of the advantages. Faster disease detection, Timely intervention, and treatment was facilitated by the suggested method. When compared to the conventional methods, the suggested method is effective. But substantial computational power and large datasets were needed for the resource-intensive models, so that the accuracy is attained [8], [9]. In low-resource farming regions, its practical applications are limited, so in this case, real-time disease detection is crucial. Classification errors are occurred because of the standard models' failure in detecting the diseases which show similar signs. To answer the disease detection challenges, an optimized deep spectral GAN framework designed for resource-constrained environments is used. For noise reduction and bilinear interpolation for efficient resizing, image pre-processing with gaussian filter or median filter was combined in this model [10]. By utilizing the PCA and RFE, the relevant features are selected, and it will support in minimizing the data dimensionality and computational complexity. Irrespective of excessive resource demand, an accurate localization was facilitated by a simplified U-Net model. Disease detection can be a disease detection additionally optimized by the BPSO. Effective convergence on optimal weights was attained by the BPSO. Effective convergence on

optimal weights was attained by the BPSO. From the classifications, in low-resource conditions, the FP and FN were minimized, and high accuracy was ensured by an optimised DSGAN². The accuracy and efficiency in resource-constrained environments were effectively balanced by this method. This method is effective for agricultural applications where timely disease detection is needed.

2|Literature Review

In order to diagnose rice leaf disease, Simhadri et al. suggested a deep learning method similar to transfer learning, ensemble learning and hybrid approaches used in the literature [11]. This review also discusses how effective certain solutions are at solving various problems. This overview covers the details of several models and hyper parameter settings, model fine-tuning techniques, and performance evaluation metrics used in different types of research.

The limits of previous research are also covered in this review, along with potential future paths for developing more reliable and effective methods for detecting rice leaf disease. Using a large dataset of 33,026 images of six distinct rice disease types: brown spot, sheath blight, bacterial stripe disease, false smut, leaf blast, and neck blast, Deng et al. presented a deep learning-based Ensemble Model that integrated sub-models and functioned as the method's core component. The Ensemble Model was validated using a different set of images [12]. The Ensemble Model was constructed using these three submodels.

By reducing misconceptions among different diseases, the Ensemble Model decreased the number of misdiagnoses. Six different kinds of rice diseases were identified using the Ensemble Model. The ensemble model is a suitable choice because many rice disease have similar appearances. The client was able to use the Ensemble Model on the web server over a network due to the smartphone app, which made it easy and efficient for the field diagnosis of rice leaf blast, false smut, neck blast, sheath blight, bacterial stripe disease, and brown spot. Yang et al. suggested a Dense Higher-Level Composition Detection Transformer (DHLC-DETR) methodology that integrates the Dense Higher-Level Composition Feature Pyramid Network (DHLC-FPN) into the detection transformer algorithm to effectively detect three diseases: flax spot, rice blast, and sheath blight [13].

First, a feature extraction network is created by combining the proposed DHLC-FPN with Res2Net to replace the detection transformer backbone network. After Res2Net has extracted five feature scales, high-density rank hybrid sampling is used to integrate them using the DHLC-FPN architecture. The transformer then generates class and prediction box predictions using fused features and location encoding. Finally, the Hungarian algorithm applies binary matching to the boxes and the prediction classes. Using data improvement, the DHLC-DETR model increased mean average precision on the IDADP datasets compared to the DETR model.

A disease detection initially, mean average precision for small target detection was improved and the amplitude of the hybrid sampling decreased. The empirical findings demonstrate that the optimised feature extraction structure may considerably increase the model's small target detection accuracy and average detection accuracy, based on the average accuracy of the IDADP rice disease dataset. By using the automatically estimated width of the rice leaves in the images as a size reference for partitioning the original input image, a CNN object detection with an image tiling technique was suggested by Kiratiratanapruk et al. [14].

Leaf width estimation was done using a small CNN, such the 18-layer ResNet architectural model. A new set of partitioned tiled sub-images with consistently sized objects was created to train a rice disease prediction model. This strategy was tested using 4,960 images of eight different rice leaf diseases: orange, red stripe, rice grassy stunt virus, brown spot, narrow brown spot, blast, blight, and streak disease. For the leaf width prediction job assessed across all eight classes, the mean absolute percentage error was 11.18%. We demonstrated the effectiveness of the leaf width prediction model in the experiment.

The mAP of the prediction performance on the YOLOv4 architecture increased from 87.56% to 91.14% when the tiled dataset was used for both training and testing. The research indicates that the suggested image-tiling method increased the effectiveness of rice disease detection. To extract features from complex and accurate detection methods, Rajpoot et al. suggested a VGG-16 transfer learning with Faster R-CNN deep architecture [15]. Three distinct types of plant diseases that impact rice are examined in this study. Brown spot, leaf smut, and bacterial leaf blight are three of the six diseases that can affect rice plants. After completing the transfer learning step, the random forest method categorizes the collected attributes. The radish field is divided into three sections using the RF classifier. The UCI Machine Learning Repository is the source of the pictures of rice plant leaves. When it comes to rice disease image classification, the recommended approach offers an average prediction accuracy.

The comprehensive experiment's findings validate the suggested approach and demonstrate its effectiveness in rice disease detection. A method for feature extraction in disease detection that combines the Gray level Co-occurrence Matrix (GLCM) and Intensity-Level Multi-Fractal Dimension (ILMFD), it has been suggested by Chaudhary and Kumar [16]. This integrated technique's effectiveness was assessed by contrasting it with other classifiers. To determine their accuracy in disease identification, the Artificial Neural Network (ANN), Support Vector Machine (SVM), and Neuro-Genetic Algorithm (Neuro-GA) were explicitly used. The Neuro-GA classifier's accuracy was shown to surpass 90% when GLCM and ILMFD were combined. Interestingly, this integrated technique accurately diagnosed brown spot illness in rice when combined with the SVM classifier. These results demonstrate the superior performance of the GLCM and ILMFD approaches in FE and validating their efficacy.

Agricultural disease identification using the SVM classifier. This study makes a substantial contribution to the field by providing a solid method for precisely diagnosing diseases in rice crops, which helps to manage agricultural activities sustainably. A Faster region-based CNN was created by Bari et al. [17]. In order to detect rice leaf diseases in real-time. An improved RPN architecture introduced by the Faster R-CNN algorithm generates candidate regions by accurately disease detection resigns. The robustness of the suggested deep learning-based method was demonstrated to be effective in the automatic diagnosis of three discriminative rice leaf diseases: rice blast, brown spot, and hispa. The Faster R-CNN model was trained using both publicly available internet datasets and proprietary real-field rice leaf datasets. Furthermore, the model's accuracy rate in recognising a healthy rice leaf was 99.25%. The results demonstrated that the Faster R-CNN model offers an exceptional rice leaf disease detection system that could more precisely detect the most common rice diseases in real-time.

Udayananda et al. suggested a CNN to detect rice plant disease by detecting and classifying images [18]. However, only a few studies have been carried out on detecting illnesses affecting rice plants. This study thoroughly explains the diseases affecting rice plants and the deep learning techniques employed to identify them. A disease detection evaluates various methods used in the literature by weighing their advantages and disadvantages. As a result, this research has identified the most accurate methods at every stage of the image identification process, which will be useful in recognizing rice plant illnesses. Deep feature with SVM and transfer learning techniques have been suggested by Sethy et al. as classification alternatives [19]. The efficacy of small CNN models, such as mobilenetv2 and shuffle-net, was examined. The performance evaluation took into account training time, F1 Score, FP Rate, accuracy, sensitivity, and specificity. Once again, statistical analysis was used to choose the best classification model.

The deep feature of ResNet50 with SVM performs better, with an F1 score of 0.9838. In vgg16, vgg19, and Alex Net, the fc6 layer makes a greater contribution to classification than fc7 and fc8. Furthermore, a number of traditional image classification models, such as Grey Level Co-occurrence Matrix plus SVM, bag-of-feature, local binary patterns plus SVM, and histogram of orientated gradients plus SVM, were compared to the CNN classification's F1 score models. For the purpose of identifying and classifying paddy leaf diseases, Ramesh and Vydeki suggested an optimised deep NN with the Jaya optimization algorithm [20]. Photographs of rice plant leaves are taken from the farm field to acquire pictures of ordinary bacterial blight, brown spots, sheath

rot, and blast illnesses. During pre-processing, RGB photos are transformed into HSV images for background removal. Binary images are retrieved based on hue and saturation to separate the sick and non-diseased portions. A clustering algorithm segments the backdrop, typical section, and ill region. The deep NN-Jaya optimization algorithm is used to classify disorders. A feedback loop is created in the post-processing stage to assess this method's stability. Artificial neural network, DAE, and deep NN are used to analyze and compare the experimental outcomes.

3 | Proposed Methodology

The suggested optimised DSGAN² for Rice Plant Leaf disease detection is examined in this section. To identify and classify signs of rice plant diseases, a number of deep learning frameworks have used visual representations. To detect prevalent leaf diseases in this kind of rice plant, the suggested algorithm is trained. When comparing the test outcomes of the suggested method and other standard methods. The accuracy of the CNN model is enhanced by training an optimized DSGAN2 as a classifier, and it is demonstrated by outcomes of the suggested method. A method was also offered by the suggested method for detecting rice plant disease with high accuracy. The process of rice plant leaf image classification in an image dataset was presented in *Fig. 2*. The leaf image quality was improved by the pre-processing step. This pre-processing step is the initial step, and it has shrinking and noise reduction. For the purpose of detecting colour and region borders accurately, the images are segmented using a U-Net architecture, once pre-processing step is completed. Then, after segmentation, for classification purposes, the most relevant features are then selected by recursive feature elimination. An optimal performance was attained by the PSO, as it modifies the model parameters. These features are then optimized by the PSO. The DSGAN² was employed to classify the optimized features. Then, the performance analysis was employed to achieve the efficacy of the model. Improving the classification accuracy for leaf images that are used in agriculture is considered as an objective of this step by step method.

3.1 | Image Pre-Processing

The process begins with importing leaf images from the dataset containing samples of rice plant leaves affected by various diseases. The accuracy of the analysis is impacted by the noise and variations of the image in resolution.

3.1.1 | Noise Reduction Using Gaussian Filtering

The leaves of rice plant diseases get affected by the diseases like bacterial blast or blight. Discolouration, lesions, or small spots are the symptoms of the diseases. The way the image noise, like random pixel intensity fluctuations or backdrop distractions covers these symptoms.

Smoothing effect was created by gaussian filter via averaging the pixel values with their neighbours' values. The detection of disease symptoms is prevented by the high-frequency noise. These gaussian filter helps in eliminating these high-frequency noises [21]. As you can see in *Fig. 3*, noise can be effectively eliminated by this method. Significant low-frequency factors like lesion edges are maintained by this method. Critical disease symptoms like (spots, lesions, or discolorations) are detected accurately and it was ensured by the filtering process via noise reduction. This contributes the model in emphasizing the disease-related patterns rather than irrelevant image artifacts. This clarity is crucial for accurate segmentation and feature elimination. Image pre-processing which is implemented on MATLAB, is depicted in *Fig. 4*.

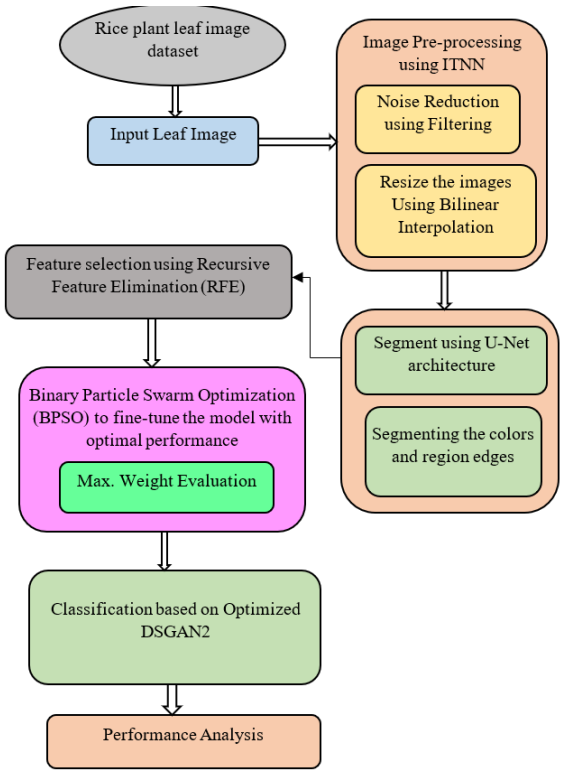


Fig. 2. The overall proposed workflow model.

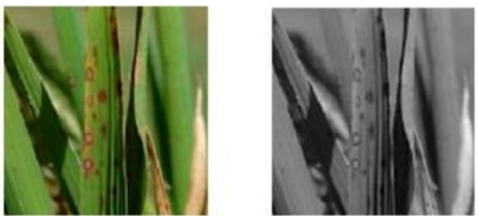


Fig. 3. Noise removal: a. Input image, and b. image obtained after denoising.

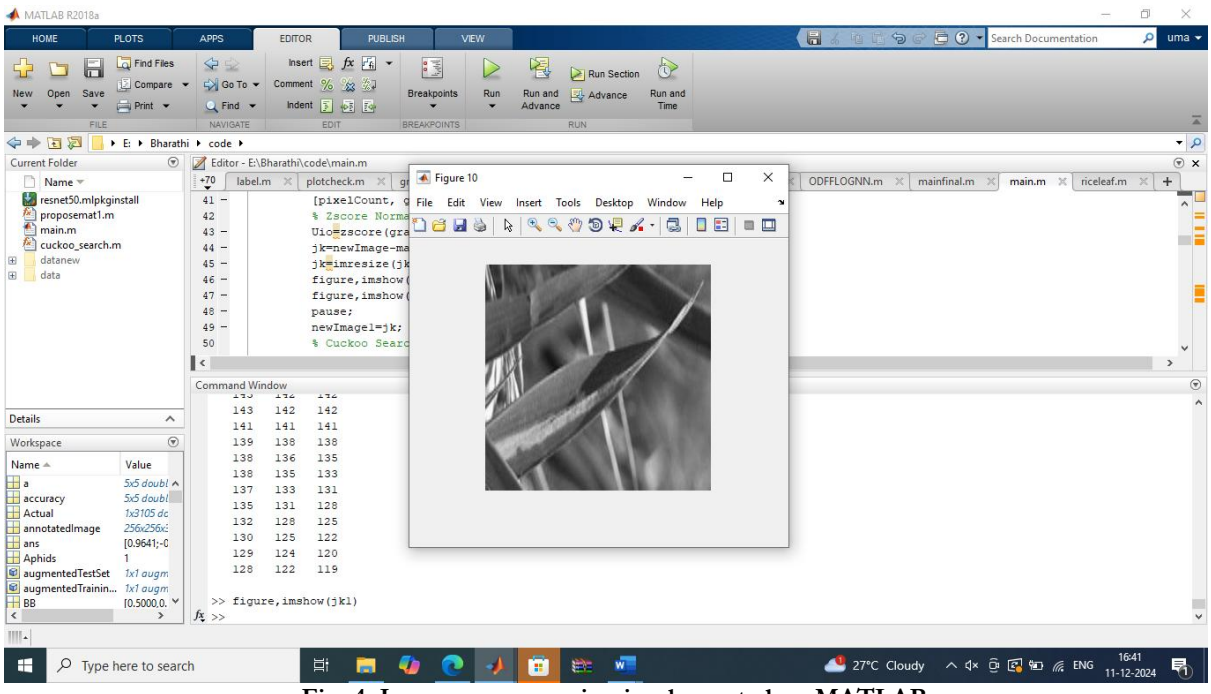


Fig. 4. Image pe-processing implemented on MATLAB.

3.1.2 | Resizing Images Using Bilinear Interpolation

In a dataset, rice leaf images might have varying sizes due to camera distance or resolution differences. Standardizing image size ensures that each input image aligns with the model's requirements, which is particularly important for deep learning models where input dimensions must be consistent. Bilinear interpolation resizes each leaf image to a fixed dimension by calculating new pixel values based on nearby pixels. This technique produces smooth, uniform images that preserve the visual integrity of disease symptoms, such as edges of spots or lesions, without introducing distortions. The model can determine features uniformly when the image dimensions are consistent. For the detection of subtle, localised disease symptoms across images, consistent image dimensions are essential. Regardless of original image size variations, the model accurately generalizing and detecting disease patterns with the help of standardized image dimensions. Together, these pre-processing steps (noise reduction and resizing) prepare rice leaf images by enhancing symptom visibility and ensuring input consistency. This preparation is critical for subsequent segmentation and classification processes, which rely on clear and uniformly scaled images to identify and classify leaf diseases accurately.

3.2 | Segmentation Using U-Net Architecture

Separating the disease affected areas from healthy leaf areas in leaf image datasets is done through this segmentation. This segmentation helps the model in emphasizing the symptomatic areas, and an accurate disease classification is attained. Disease symptoms, such as lesions, spots, or discoloration, are often localized in specific leaf areas. Segmentation allows us to isolate these affected areas, making it easier for the model to analyse and classify the disease based on distinct visual patterns without being distracted by healthy leaf regions. This step lays a strong foundation for feature extraction and disease classification by accurately segmenting diseased areas. U-Net, a widely acclaimed deep learning architecture for image segmentation, is particularly effective in biomedical and agricultural image analysis [22]. Its unique ability to capture fine-grained details, such as the boundaries of lesions or the spread of spots, in stills confidence in its precision. The 'U' shape of U-Net, with its encoder-decoder structure, is instrumental in this process. In order to provide accurate segmentation outputs, the encoder path downsampling the image to collect contextual information, while the decoder path upsampling to restore spatial information. The rice leaf image is passed through multiple convolutional and pooling layers in the encoder path. Each layer reduces the spatial resolution while capturing deeper feature representations, such as the colour and texture differences between diseased and healthy regions.

I. A crucial component is lied at the heart of the U-Net; the bottleneck layer. This layer plays a pivotal role in capturing high-level disease-related features, such as the boundaries of lesions or the spread of spots. It provides a compact representation of disease symptoms, ensuring no crucial detail is overlooked in the segmentation process. The decoder path gradually up samples the feature maps, restoring the image's spatial resolution while focusing on regions marked as potentially diseased. Skip connections between corresponding layers in the encoder and decoder paths help retain essential details from earlier layers, making Segmentation more precise.

II. The final output of the U-Net is a binary or multi-class segmentation mask where each pixel is classified as part of a disease-affected region or a healthy region. This mask highlights infected spots, lesions, or discoloration on the rice leaf.

The segmented image now contains only the disease-related regions from the rice plant leaf image, with healthy leaf areas either masked or ignored. This focused output enables downstream processes, like feature selection and classification, to analyze only the disease-specific parts, improving the accuracy of diagnosis and classification. Using U-Net for Segmentation in rice leaf disease detection enhances the model's ability to distinguish between healthy and diseased areas. This provides precise input for subsequent classification steps and ultimately leads to more reliable disease identification.

3.3 | Feature Selection Using Recursive Feature Elimination

Once segmentation is complete, each segmented image has a large set of features from the dataset, possibly including colour intensity, texture patterns, edge contours, or spectral information. However, not all of these features will be equally relevant for identifying disease symptoms. Some might add noise or unnecessary complexity, which can degrade the model's performance [23]. By using recursive feature elimination, we aim to improve model accuracy, reduce computational complexity, and prevent overfitting by retaining only the most impactful features. Let's denote the initial set of features extracted from the segmented images as *Eq. (1)*;

$$F = \{f_1, f_2, f_3, \dots f_n\}, \quad (1)$$

where n is the total number of features generated from segmentation. These features could represent colour intensities, texture patterns, or other properties relevant to identifying disease symptoms in rice leaves. Recursive feature elimination works by training a model iteratively and ranking features by their importance in making predictions. A base model, often a simple classifier like a SVM, is trained on the full feature set F. The model's objective is to minimize a loss function L such as the Mean Squared Error (MSE) or cross entropy loss, depending on the classification setup. The initial model training can be expressed as *Eq. (2)*;

$$\min_{\theta} L\left(Y, \hat{Y}(F;\theta)\right). \quad (2)$$

where θ represents the model parameters, Y is the true label vector, and $\hat{Y}(F;\theta)$ is the predicted label vector based on the feature set F. Once the model is trained, each feature f_i is assigned an importance score. In RFE, this score is based on how much the model's performance would degrade if that feature were removed. For an SVM, this can be quantified by the absolute weight $|w_i|$ of each feature in the decision function. In general, the importance of feature f_i, $I(f_i)$ can be represented as *Eq. (3)*;

$$I(f_i) = |w_i|, \quad (3)$$

where w_i is the weight assigned to feature f_i in the model.

Recursive feature elimination removes the least important feature(s) in each iteration. Suppose we eliminate one feature per iteration (though recursive feature elimination can also remove a small subset). Let be the feature with the smallest importance score can be expressed as *Eq. (4)*;

$$f_{min} = \arg\min_{f_i \in F} I(f_i). \quad (4)$$

Then, f_{min} is removed from F, resulting in a reduced feature set as *Eq. (5)*;

$$F \leftarrow F/(f_{min}), \quad (5)$$

with the reduced feature set F, the model is re-trained, and new importance scores $I(f_i)$ are calculated for the remaining features. This re-training step allows recursive feature elimination to reassess the importance of each feature in the context of the reduced set, as the removal of certain features may change the relative importance of the others. The model optimization now focuses on the updated feature set F can be expressed as *Eq. (6)*;

$$\min_{\theta} L\left(Y, \hat{Y}(F;\theta)\right). \quad (6)$$

The elimination process is repeated iteratively. In each iteration, the feature set is updated by removing the least important feature(s) from the previous set, retraining the model, and recalculating feature importance. This iterative process can be formally represented as;

Start with $F(0) = F$.

For each iteration t.

I. Train the model with $F^{(t-1)}$.

II. Calculate $I(f_i)$ for all $f_i \in F^{(t-1)}$.

III. Remove $f_{min} = \text{argmin}_{f_i \in F^{(t-1)}} I(f_i)$.

IV. Update $F^t = F^{(t-1)} \setminus \{f_{min}\}$.

RFE continues until a stopping criterion is met.

I. Fixed Number of Features: Stop when a predefined number of features k remain, where k is chosen based on validation performance.

II. Performance Threshold: Stop if removing additional features would lead to a significant drop in the model's performance on a validation set.

III. The final subset of features, $F^*$is then passed to the next stage of the workflow for further refinement in *Eq. (7)*.

$$F = F^{(T)}, \quad (7)$$

where, T is the iteration at which the stopping criterion is satisfied.

Recursive feature elimination reduces noise and dimensionality, leading to a simpler and more interpretable model from the rice leaf image dataset. By retaining only, the essential features, recursive feature elimination improves computational efficiency, reduces overfitting, and enhances classification accuracy in the downstream processes (PSO and DSGAN2). Ultimately, recursive feature elimination fine tunes the feature set to ensure the model focuses on the most relevant data, boosting the robustness of rice leaf disease detection.

4|Binary PSO (BPSO) using Feature with the Closest Weight

Binary Particle Swarm Optimization (BPSO) can be effectively used to select features for rice plant leaf disease detection by finding the closest weights (binary values indicating feature inclusion or exclusion) that optimize classification accuracy. In BPSO, each particle in the swarm represents a binary solution vector for feature selection, where each bit corresponds to a feature, with 1 meaning the feature is selected and 0 meaning it is not. For example, if we have 10 diseases images features, a particle could be represented as [1,0,1,0,1,0,0,1,0,1].

Each particle i has.

Position vector $X_i = x_{i1}, x_{i2}, x_{i3}, \ldots . x_{iN}$.

Velocity vector $V_i = v_{i1}, v_{i2}, v_{i3}, \ldots . v_{iN}$which influences the probability of each bit flipping from 0 to 1 or vice versa.

The velocity of each particle determines the likelihood of a feature's inclusion in the next iteration. The velocity for each bit is expressed as *Eq. (8)*;

$$v_{ij}(t+1) = \omega . v_{ij}(t) + c_1 \cdot r_1 \cdot \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 \cdot r_2 \cdot \left(g_{ij}(t) - x_{ij}(t)\right). \quad (8)$$

where:

ω is the inertia weight that controls exploration and exploitation.

c_1 and c_2 are acceleration constants for cognitive and social components.

r_1 and r_2 are random numbers in [0,1].

p_{ij} is the personal best position of particle i.

g_{ij} is the global best position among all particles.

In BPSO, instead of continuous movement, a particle's position x_{ij} is updated by applying a sigmoid function to v_{ij} to determine the probability of each bit being set to 1 can be expressed as *Eq. (9)*;

$$S(v_{ij}) = \frac{1}{e + e^{-v_{ij}}}. \quad (9)$$

A random threshold is then applied in *Eq. (10)*,

$$x_{ij}(t+1) \equiv \begin{cases} 1, & \text{if } r = s\left(v_{ij}(t+1)\right), \\ 0, & \text{otherwise}, \end{cases} \quad (10)$$

where, r is a random number in [0, 1]. If r is less than $S(v_{ij})$ the feature is included ($x_{ij} = 1$) otherwise, it is excluded ($x_{ij} = 0$).

The fitness function evaluates the feature subset for accuracy in detecting rice plant leaf diseases. This can involve training a classifier with the selected features and calculating its accuracy on a validation set. The fitness F_i for particle i is given by *Eq. (11)*;

$$F_i = \text{accuracy}(X_i) - \alpha \cdot |X_i|, \quad (11)$$

where:

(X_i) accuracy is the classifier accuracy using the feature subset defined by X_i.

$|X_i|$ is the number of selected features.

α is a penalty coefficient to discourage excessive feature selection, improving the model's generalization. After evaluating the fitness, update each particle's personal best p_i if its current fitness F_i is better than previous best and global best g if any particle's fitness exceeds the current global best fitness. Repeat Steps 2-5 until convergence criteria are met (e.g., maximum iterations or no improvement in global best). The resulting global best position vector g will indicate the optimal feature subset for disease detection. In order to identify the most informative features for rice leaf disease detection, BPSO uses iterative optimisation with binary positions and probabilistic velocity updates. This approach helps balance the model's accuracy and feature compactness, ensuring effective disease classification with minimal features.

5 | Optimized Deep Spectral Generative Adversarial Neural Network (DSGAN^2)

DSGAN^2 combines the advantages of spectral information processing and GANs to create an optimized model that accurately classifies images based on disease symptoms [24]. GANs are typically used for data generation but, in this case, the architecture has been adapted and optimized for classification, leveraging both real and generated data features for improved disease detection. A GAN consists of two main components:

I. Generator GGG: This network generates synthetic data samples (fake images) that mimic real data.

II. Discriminator disease detection decoder: This network distinguishes between real and fake data samples.

The decoder and generator in DSGAN^2 are specially tuned to extract spatial and spectral features from rice leaf images. The model's ability to detect subtle disease patterns is improved.

The generator G in DSGAN^2 is designed to extract spectral features from input images. It learns to generate spectral representations that mimic really diseased and healthy rice leaf data. By focusing on spectral characteristics, the generator can capture wavelength-specific features, such as colour and texture variations, that are relevant to identifying different types of leaf diseases. The G objective is to create synthetic images $\hat{X}$ that are close to the real samples X in spectral distribution. This process can be represented by *Eq. (12)*;

$$\hat{X} = G(z; \theta_G), \quad (12)$$

where, z is a random noise vector and θ_G are the parameters of the generator network. The aim is to make $\hat{X}$ indistinguishable from real images in terms of spectral features. The decoder is trained to differentiate between real images X (actual rice leaf images) and fake images $\hat{X}$ (generated by GGG). A probability $D(X)$, which indicates whether an input image is real (near to 1) or fake (close to 0), is output by the D. The D objective is to maximize the probability of correctly classifying real and fake images. This objective can be expressed as *Eq. (13)*;

$$\max_{\theta_D} \mathbb{E}_{X\sim p_{real}}[\log D(X)] + \mathbb{E}_{X\sim p_{fake}}\left[\log\left(1 - D(\hat{X})\right)\right], \quad (13)$$

where, p_{real} is the distribution of real images and p_{fake} is the distribution of images generated by G. The generator, on the other hand, is optimized to fool the discriminator. It aims to minimize the probability that D correctly classifies $\hat{X}$ as fake. This adversarial objective for the generator is expressed as *Eq. (14)*;

$$\max_{\theta_D} \mathbb{E}_{X\sim p_{fake}}\left[\log\left(1 - D(\hat{X})\right)\right]. \quad (14)$$

Through this adversarial process, both G and D improve iteratively. G learns to produce more realistic spectral images, while D becomes better at detecting fakes. In DSGAN2, spectral and spatial constraints are added to the loss functions to ensure that generated images contain both accurate spectral and spatial information. This is important in disease classification, where spatial patterns (e.g., lesion shapes and color distribution) play a critical role. The overall generator loss function can thus be formulated as *Eq. (15)*;

$$L_G = \mathbb{E}_{X\sim p_{fake}}\left[\log\left(1 - D(\hat{X})\right)\right] + \lambda_s L_{Spectral} + \lambda_c L_{Classification}, \quad (15)$$

where, $L_{Spectral}$ is a loss term ensuring the generator produces images with realistic spectral information. $L_{Classification}$ is a loss term related to classification accuracy on generated images, λ_s and λ_c are hyperparameters controlling the importance of spectral and classification losses.

PSO is used to fine-tune the weights in both the generator and discriminator to achieve the closest weight configurations for optimal classification performance. This optimization improves the accuracy and robustness of DSGAN2, ensuring the model captures the most relevant features for rice leaf disease classification. Once trained, DSGAN^2 uses the discriminator's classification capabilities to predict disease classes. The discriminator is repurposed as a classifier, leveraging the features learned during adversarial training. When a new input image X_{new} is fed into the model, the discriminator assigns it a probability score for each disease class.

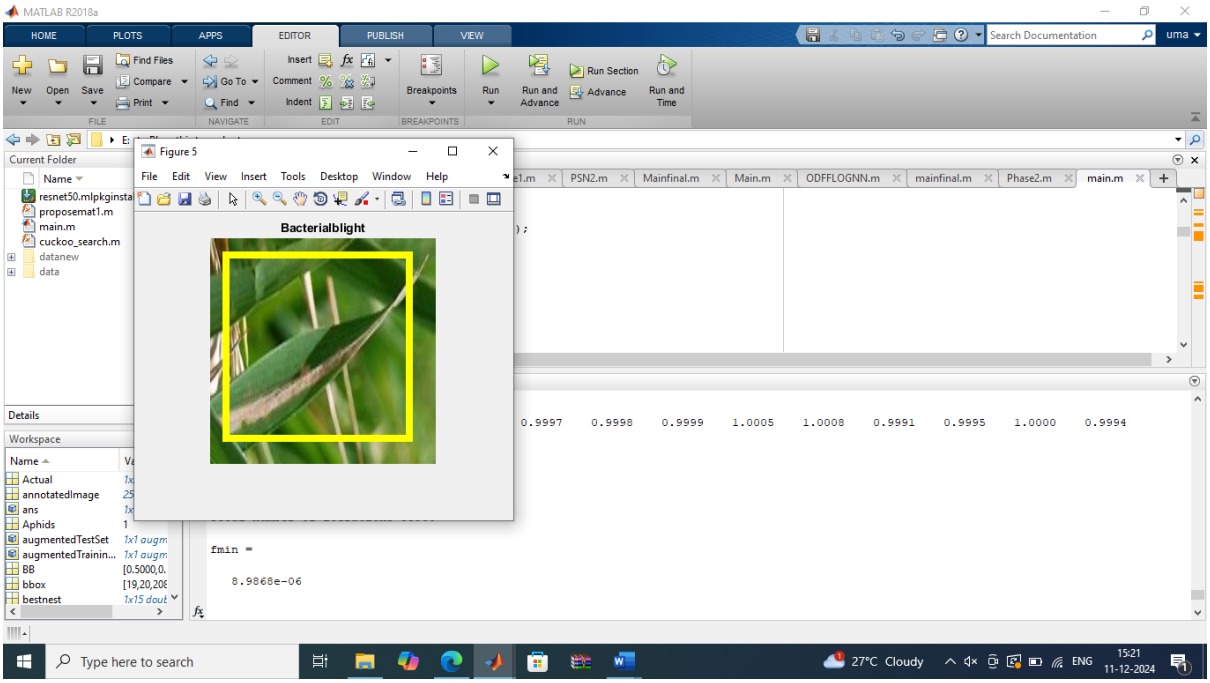
Let $P(y|X_{new})$ represent the probability of class y given the new image X_{new} expressed as *Eq. (16)*;

$$P(y|X_{new}) = D(X_{new}), \quad (16)$$

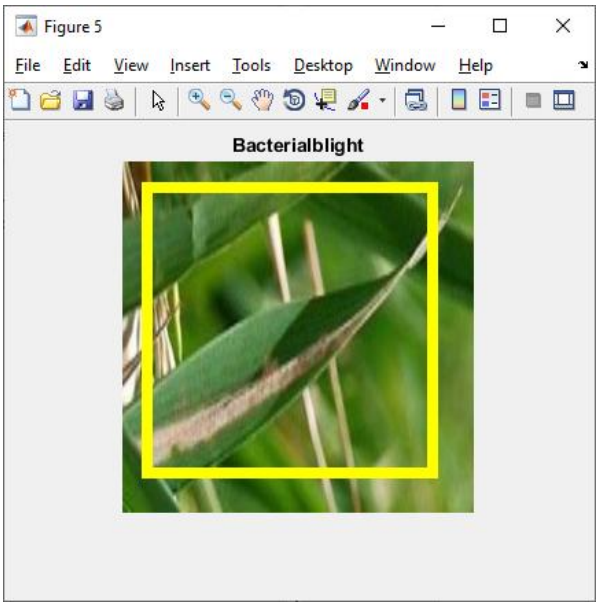
where D now represents the modified discriminator that outputs disease class probabilities instead of a binary real/fake decision. The overall objective function for training DSGAN^2 is a combination of the adversarial loss, spectral consistency loss, and classification loss can be expressed as *Eq. (17)*;

$$L_{DSGAN2} = \min_{\theta_G}\max_{\theta_D}\left(\mathbb{E}_{X\sim p_{real}}[\log D(X)] + \mathbb{E}_{X\sim p_{fake}}\left[\log\left(1 - D(\hat{X})\right)\right]\right) + \lambda_s L_{Spectral} + \lambda_c L_{Classification}. \quad (17)$$

This combined loss function allows DSGAN^2 to accurately classify rice plant leaf diseases by balancing adversarial training with spectral and spatial fidelity. DSGAN^2 is optimized for high-dimensional, complex data like rice leaf images. By incorporating spectral information, DSGAN^2 captures disease-relevant patterns more effectively than traditional models. The sample output shown on *Fig. 5*. The adversarial setup also allows DSGAN^2 to learn robust features, improving its generalization capability and accuracy in real-world disease detection scenarios.



a.



b.

Fig. 5. Image classification; a. Classified image on MATLAB screens, and b. Classification output of the proposed work.

6 | Result and Discussion

Using the trained features of the Rice Plant image dataset, the suggested Optimised DSGAN^2 implementation outcomes achieve the recommended performance and outcomes. The accuracy, consistency, reproducibility, and recall of the measures taken at this stage are evaluated by performance evaluation. Performance values are the sum of positive and negative values. The metrics being tested are sensitivity, specificity, false speed, recall, classification detection accuracy, and precision accuracy. A set of images of processed vegetation used to assess the effectiveness of the suggested approach, and it is described in *Table 1*.

Every set of training and test photos is classified to assess leaf texture detection. The formulas for recall, accuracy, and precision are expressed as *Eqs. (18)-(20).*

$$\text{Accuracy} = \frac{T_P}{T_P + T_N} \times 100. \quad (18)$$

$$\text{Precision} = \frac{T_P}{T_P + F_P} \times 100. \quad (19)$$

$$\text{Recall} = \frac{T_P}{T_P + F_N} \times 100. \quad (20)$$

Table 1. Performance evaluation of precision.

No. of Images	APS-DCCNN	Alex Net	CNN	DSGAN2	Optimized DSGAN2
25	40	48	55	60	66
50	47	53	58	67	72.8
100	54	68	73	76.9	79.12
150	62	74	80	85	90
200	76	80	87	93	96

Fig. 6 illustrates the precision rates of five image processing models APS-DCCNN, Alex Net, CNN, DSGAN2, and Optimized DSGAN2 across different numbers of images (25, 50, 100, 150, and 200). Generally, precision increases as images increase, indicating better model performance with more data. DSGAN2 and Optimized DSGAN2 exhibit the highest precision at each image count, with Optimized DSGAN2 achieving slightly higher scores as the dataset grows, reaching around 96% precision at 200 images. APS-DCCNN, while initially lagging behind DSGAN2-based models, shows steady improvement and competes closely at higher image volumes. However, Alex Net and CNN consistently show lower precision than the other methods, particularly at lower image counts, where a significant margin outperforms them. By 200 images, Alex Net and CNN catch up somewhat but remain less precise than the DSGAN2 models and APS-DCCNN. Overall, Optimized DSGAN2 stands out as the most accurate model, with DSGAN2 and APS-DCCNN also demonstrating strong performance, especially in larger datasets.

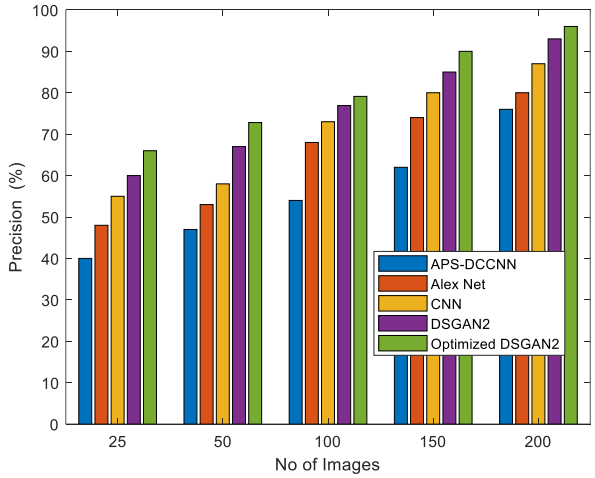


Fig. 6. Comparison of precision score with existing methods.

Fig. 7 and *Table 2* illustrate the performance of various image processing methods APS-DCCNN, Alex Net, CNN, DSGAN2, and Optimized DSGAN2 measured by recall across different image quantities (25, 50, 100, 150, and 200). Across all image volumes, the APS-DCCNN and Optimized DSGAN2 consistently achieve higher recall rates, with the Optimized DSGAN2 showing slightly improved performance over APS-DCCNN

at larger image counts. As image quantity increases, recall generally improves for all methods, demonstrating the positive impact of more data on model performance. Specifically, Optimized DSGAN2 and APS-DCCNN exceed 95% recall at 200 images, while DSGAN2 also performs competitively but slightly lower. Alex Net and CNN, while achieving moderate recall rates, consistently underperform compared to the other advanced models, indicating less effectiveness in high recall tasks. Overall, Optimized DSGAN2 emerges as the best-performing method, especially at high image counts, followed closely by APS-DCCNN, which also maintains robust recall across different scales. This analysis suggests that Optimized DSGAN2 and APS-DCCNN are well-suited for tasks requiring high recall in larger datasets.

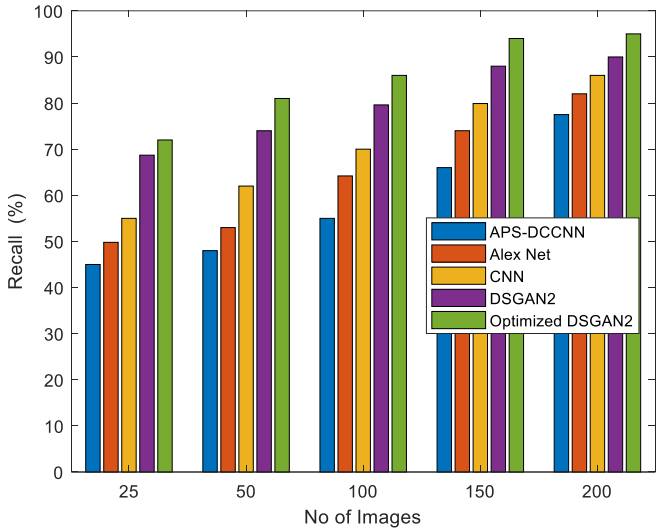


Fig. 7. Comparison of recall score with existing methods.

Table 2. Performance evaluation of recall.

No. of Images	APS-DCCNN	Alex Net	CNN	DSGAN2	Optimized DSGAN2
25	45	49.8	55	68.7	72
50	48	53	62	74	81
100	55	64.2	70	79.6	86
150	66	74	79.9	88	94
200	77.5	82	86	90	95

Fig. 8 and *Table 3*, illustrate the specificity (%) of various models APS-DCCNN, Alex Net, CNN, DSGAN2, and our proposed Optimized DSGAN2 across increasing image counts. Specificity here reflects each model's ability to correctly identify negative instances, with higher percentages indicating fewer False Positives (FPs). Our Optimized DSGAN2 consistently achieves the highest specificity across all image sets, reaching nearly 94.8% at 200 images, showcasing its robustness in distinguishing true negatives effectively. In comparison, APS-DCCNN shows fluctuating and generally lower specificity, especially at higher image counts, indicating a relative weakness in handling complex datasets. Alex Net and CNN deliver moderate performance but consistently outperform the DSGAN2-based models. DSGAN2 shows strong specificity, though it remains slightly below our optimized version. This trend suggests that the Optimized DSGAN2 is particularly effective in reducing FPs, which is crucial in agricultural applications where precise classification is essential. This improvement in specificity underscores the enhanced reliability of our proposed optimized model over traditional neural network approaches.

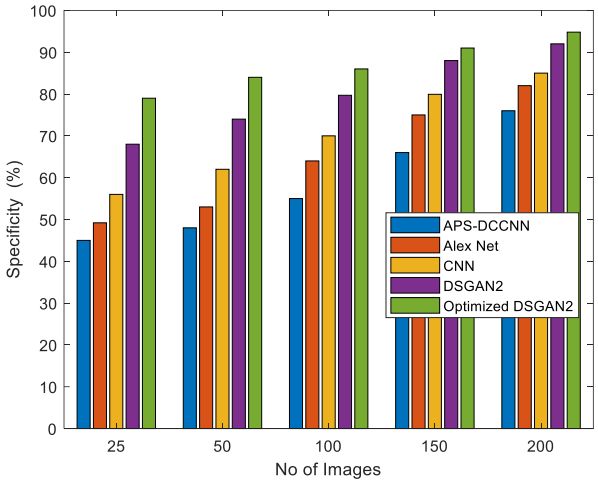


Fig. 8. Comparison of Specificity value with Existing methods.

Table 3. Performance evaluation of specificity.

No. of Images	APS-DCCNN	Alex Net	CNN	DSGAN2	Optimized DSGAN2
25	45	49.2	56	68	79
50	48	53	62	74	84
100	55	64	70	79.7	86
150	66	75	79.93	88	91
200	76	82	85	92	94.8

Fig. 9 and *Table 4* compare the detection accuracy of various models APS-DCCNN, Alex Net, CNN, DSGAN2, and our proposed Optimized DSGAN2 across different image counts for rice plant detection. Our Optimized DSGAN2 consistently demonstrates the highest accuracy, reaching nearly 99.8% as the dataset size increases to 200 images. In contrast, APS-DCCNN shows the lowest performance, particularly at more minor image counts, where it trails behind other methods. Alex Net and CNN achieve moderate accuracy, yet DSGAN2-based models consistently outperform them. DSGAN2 shows strong results, with accuracy steadily increasing, though it remains slightly lower than the optimized version. The clear trend is that Optimized DSGAN2 excels in low and high image quantities, indicating its robustness and reliability in complex classification tasks, even with larger datasets. This highlights the effectiveness of our optimized architecture in enhancing detection accuracy and outperforming conventional neural network models.

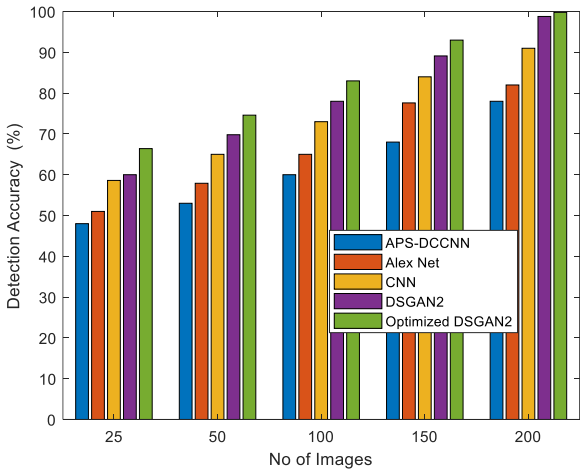


Fig. 9. Comparison of detection accuracy with Existing methods.

Table 4. Performance evaluation of detection accuracy.

No. of Images	APS-DCCNN	Alex Net	CNN	DSGAN2	Optimized DSGAN2
25	48	51	58.6	60	66.4
50	53	57.9	65	69.8	74.6
100	60	65	73	78	83
150	68	77.6	84	89.12	93
200	78	82	91	98.8	99.8

Fig. 10 and *Table 5*, illustrate the false rate (FR) for different models applied to rice plant images, focusing on minimizing classification errors. The suggested Optimized DSGAN² achieves the lowest FR across all image counts, demonstrating superior performance in error reduction. Specifically, DSGAN² records a false rate of 40.8%, significantly lower than APS-DCCNN at 55.2%, Alex Net at 50.4%, and standard CNN at 49.5%. As the number of images increases, our optimized model consistently outperforms the other approaches, highlighting its robustness in managing larger datasets and minimizing errors in the training and testing phases. This suggests that the Optimized DSGAN², with its enhanced generative architecture, is an effective solution for complex agricultural image classification tasks, reducing error rates more effectively than traditional models.

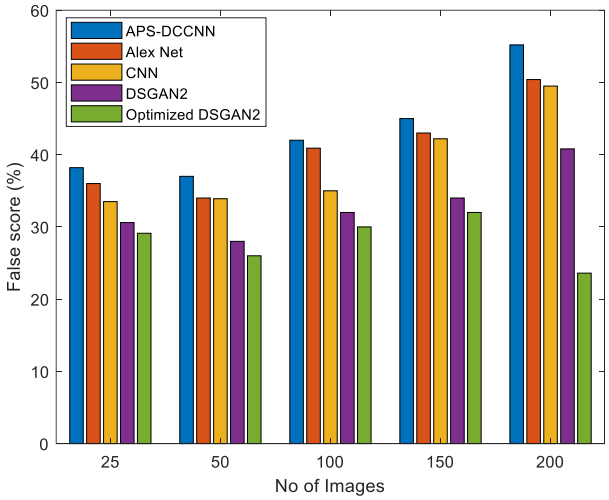


Fig. 10. Comparison of False score with Existing methods.

Table 5. Performance evaluation of false score.

No. of Images	APS-DCCNN	Alex Net	CNN	DSGAN2	Optimized DSGAN2
25	38.2	36	33.5	30.6	29.12
50	37	34	33.9	28	26
100	42	40.9	35	32	30
150	45	43	42.2	34	32
200	55.2	50.4	49.5	40.8	23.6

The confusion matrix (*Fig. 11*) shows strong performance in classifying rice diseases, with high accuracy for each category. Bacterial Blight is accurately predicted 233 times, with minor misclassifications (2 as Tungro and 1 as Hispa). Leaf Blast has the highest correct predictions (333), with very few errors. Brown Spot, Tungro, and Hispa are also accurately predicted, with 261, 224, and 210 correct classifications, respectively. Misclassifications are minimal, indicating a robust model for disease identification across all categories. The diagonal dominance reflects high precision, while off-diagonal entries suggest limited confusion among disease types.

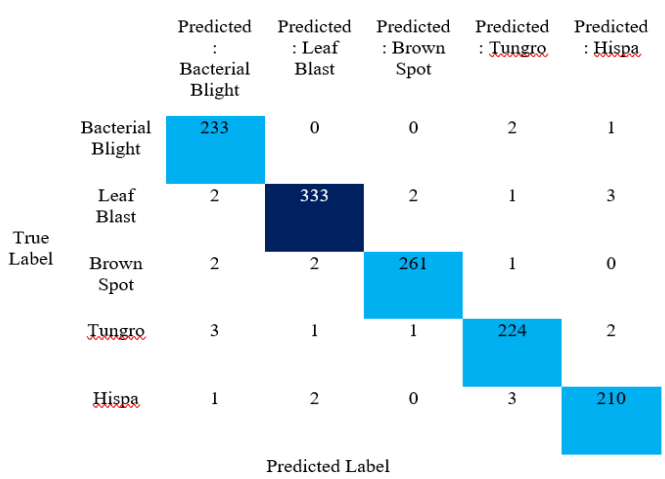


Fig. 11. Confusion matrix.

7 | Conclusion

Finally, the suggested DSGAN2 framework demonstrates a significant advancement in automated rice leaf disease detection, effectively balancing accuracy and computational efficiency for deployment in resource-limited settings. By integrating image pre-processing techniques and Recursive Feature Elimination (RFE), the system enhances feature relevance, while BPSO ensures optimal feature selection, further refining model performance. The simplified U-Net structure provides precise disease localization with reduced resource requirements, making the framework well-suited for real-time applications in low-resource rural areas. Achieving an error rate of 40.8%, DSGAN2 surpasses traditional models like APS-DCCNN, Alex Net, and standard CNN, proving both its accuracy and robustness in minimizing FPs and negatives. Overall, this approach promises to support farmers in proactive disease management, promoting healthier crops and improved yield through accessible and efficient technology tailored to agricultural needs.

Authors' Contributions

X.: Writing-original draft, Methodology, Conceptualization, and Formal Analysis Y.: Research Design, Data Curation, Visualization, Writing-Review & Editing, and Validation. Z.: Software, Validation, formal analysis, and investigation. The authors have read and agreed to the published version of the manuscript.

Data Availability

Sufficient data is available upon request.

Funding

This research received no external funding.

Conflict of Interest

No potential conflict of interest was reported by the authors.

Consent for Publication

All authors have provided their consent for the publication of this manuscript.

Ethics Approval and Consent to Participate

This article does not involve studies with human participants or animals conducted by any authors.

References

[1] Kaur, A., Guleria, K., & Trivedi, N. K. (2021). Rice leaf disease detection: A review. In *2021 6th international conference on signal processing, computing and control (ISPCC)* (pp. 418-422). IEEE. https://doi.org/10.1109/ISPCC53510.2021.9609473

[2] Pothen, M. E., & Pai, M. L. (2020). *Detection of rice leaf diseases using image processing* [presentation]. 2020 fourth international conference on computing methodologies and communication (ICCMC) (pp. 424–430). https://ieeexplore.ieee.org/abstract/document/9076527/1000

[3] Marandi, S. M. C., Sahoo, A., & Mohanty, S. (2023). *Using a deep neural network* [presentation]. Computing, communication and learning: first international conference, cocole 2022, warangal, india, october 27--29, 2022, proceedings (p. 231). https://books.google.com/books

[4] Nalini, S., Krishnaraj, N., Jayasankar, T., Vinothkumar, K., Britto, A. S. F., Subramaniam, K., & Bharatiraja, C. (2021). Paddy leaf disease detection using an optimized deep neural network. *Computers, materials & continua*, *68*(1), 1117. https://doi.org/10.32604/cmc.2021.012431

[5] Babu, S., Maravarman, M., & Pitchai, R. (2022). Detection of rice plant disease using deep learning techniques. *Journal of mobile multimedia*, *18*(3), 757–770. https://doi.org/10.13052/jmm1550-4646.18314

[6] Bhattacharya, S., Mukherjee, A., & Phadikar, S. (2020). A deep learning approach for the classification of rice leaf diseases. *Intelligence enabled research: dosier 2019* (pp. 61–69). Springer. https://doi.org/10.1007/978-981-15-2021-1_8

[7] Tawde, T., Deshmukh, K., Verekar, L., Reddy, A., Aswale, S., & Shetgaonkar, P. (2021). Identification of rice plant disease using image processing and machine learning techniques. *2021 international conference on technological advancements and innovations (ICTAI)* (pp. 102-108). IEEE. https://doi.org/10.1109/ICTAI53825.2021.9673214

[8] Aggarwal, M., Khullar, V., & Goyal, N. (2023). Exploring classification of rice leaf diseases using machine learning and deep learning. *2023 3rd international conference on innovative practices in technology and management (ICIPTM)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICIPTM57143.2023.10117854

[9] Patidar, S., Pandey, A., Shirish, B. A., & Sriram, A. (2020). Rice plant disease detection and classification using deep residual learning. *International conference on machine learning, image processing, network security and data sciences* (pp. 278-293). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-15-6315-7_23

[10] Saini, A., Guleria, K., & Sharma, S. (2023, August). Multiclass classification of rice leaf disease using deep learning based model. *2023 3rd Asian conference on innovation in technology (ASIANCON)* (pp. 1-6). IEEE. https://doi.org/10.1109/ASIANCON58793.2023.10270229

[11] Simhadri, C. G., Kondaveeti, H. K., Vatsavayi, V. K., Mitra, A., & Ananthachari, P. (2025). Deep learning for rice leaf disease detection: A systematic literature review on emerging trends, methodologies and techniques. *Information processing in agriculture*, *12*(2), 151–168. https://doi.org/10.1016/j.inpa.2024.04.006

[12] Deng, R., Tao, M., Xing, H., Yang, X., Liu, C., Liao, K., & Qi, L. (2021). Automatic diagnosis of rice diseases using deep learning. *Frontiers in plant science*, *12*, 701038. https://doi.org/10.3389/fpls.2021.701038

[13] Yang, H., Deng, X., Shen, H., Lei, Q., Zhang, S., & Liu, N. (2023). Disease detection and identification of rice leaf based on improved detection transformer. *Agriculture*, *13*(7), 1361. https://www.mdpi.com/2077-0472/13/7/1361

[14] Kiratiratanapruk, K., Temniranrat, P., Sinthupinyo, W., Marukatat, S., & Patarapuwadol, S. (2025). Automatic detection of rice disease in images of various leaf sizes. *IET image processing*, *19*(1), e13301. https://doi.org/10.1049/ipr2.13301

[15] Rajpoot, V., Tiwari, A., & Jalal, A. S. (2023). Automatic early detection of rice leaf diseases using hybrid deep learning and machine learning methods. *Multimedia tools and applications*, *82*(23), 36091–36117. https://doi.org/10.1007/s11042-023-14969-y

[16] Chaudhary, S., & Kumar, U. (2024). Automated detection and classification of rice crop diseases using advanced image processing and machine learning techniques. *Traitement du signal*, *41*(2), 739. https://www.academia.edu/download/121028548/129612.pdf

[17] Bari, B. S., Islam, M. N., Rashid, M., Hasan, M. J., Razman, M. A. M., Musa, R. M., … & Majeed, A. P. P. A. (2021). A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ computer science*, *7*, e432. https://doi.org/10.7717/peerj-cs.432

[18] Udayananda, G., Shyalika, C., & Kumara, P. (2022). Rice plant disease diagnosing using machine learning techniques: a comprehensive review. *SN applied sciences*, *4*(11), 311. https://doi.org/10.1007/s42452-022-05194-7

[19] Sethy, P. K., Barpanda, N. K., Rath, A. K., & Behera, S. K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Computers and electronics in agriculture*, *175*, 105527. https://doi.org/10.1016/j.compag.2020.105527

[20] Ramesh, S., & Vydeki, D. (2020). Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm. *Information processing in agriculture*, *7*(2), 249–260. https://doi.org/10.1016/j.inpa.2019.09.002

[21] Annamalai, M., & Muthiah, P. B. (2022). An early prediction of tumor in heart by cardiac masses classification in echocardiogram images using robust back propagation neural network classifier. *Brazilian archives of biology and technology*, *65*, e22210316. https://www.scielo.br/j/babt/a/FQCjzbjPWGtVm9DbfNmXXGg/?format=pdf&lang=en

[22] Vijaya Kishore, V., & Kalpana, V. (2020). Effect of noise on segmentation evaluation parameters. In *Soft computing: theories and applications: proceedings of socta 2019* (pp. 443–453). Springer. https://doi.org/10.1007/978-981-15-4032-5_41

[23] Islam, A., Islam, R., Haque, S. M. R., Islam, S. M. M., & Khan, M. A. I. (2021). Rice leaf disease recognition using local threshold based segmentation and deep CNN. *International journal of intelligent systems and applications*, *11*(5), 35. https://www.researchgate.net/profile/S_M_Rafizul_Haque/publication/355031837_Rice_Leaf_Disease_Recognition_using_Local_Threshold_Based_Segmentation_and_Deep_CNN/links/6158a67ae7bb415a5d50c8bd/Rice-Leaf-Disease-Recognition-using-Local-Threshold-Based-Segmentation-and-Deep-CNN.pdf

[24] Kalpana, V., Vijaya Kishore, V., & Praveena, K. (2019). A common framework for the extraction of ILD patterns from CT image. *Emerging trends in electrical, communications, and information technologies: proceedings of icecit-2018* (pp. 511–520). Springer. https://doi.org/10.1007/978-981-13-8942-9_42