IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 1, February 2026, pp. 707~724
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i1.pp707-724

Autoencoder and GAN-aided plant disease detection in rice and cotton via hybrid feature extraction and decision tree classification

Anandraddi Naduvinamani[1], Jayshri Rudagi[2], Mallikarjun Anandhalli[3]
[1]Department of Electronics and Communication Engineering, S. G. Balekundri Institute of Technology, Affiliated to Visvesvaraya Technological University, Belagavi, India
[2]Department of Electronics and Communication Engineering, Jain College of Engineering, Belagavi, Affiliated to Visvesvaraya Technological University, Belagavi, India
[3]Department of Electronics and Communication Engineering, Central University of Karnataka, Kalaburagi, India

Article Info

Article history:

Received Sep 1, 2024
Revised Nov 13, 2025
Accepted Jan 22, 2026

Keywords:

Decision tree
Generative adversarial networks
Image feature extraction
Machine learning
Plant disease detection
Rice and cotton

ABSTRACT

In agriculture, crop diseases caused by pathogens, including bacteria, viruses, and fungi, pose a significant threat to the effectiveness of agricultural productivity. Some major crops in India such as rice and cotton are adversely impacted, leading to economic loss and loss of production. Timely intervention and sustainable agriculture depend on proper and early identification of diseases. In this paper, we propose a novel plant disease detection framework that integrates generative adversarial network (GAN)-based image denoising with feature extraction and decision tree (DT) classification. The GAN module effectively removes noise from agricultural images, enhancing quality and stability under challenging imaging conditions. Following denoising, a combination of color, texture, and gradient features is extracted to obtain rich and discriminative patterns, which are then used to train a DT classifier for disease identification. Experiments are conducted on benchmark datasets comprising rice and cotton leaf images. The proposed system achieves superior performance, with 98.70% accuracy, 98.20% precision, 97.22% recall, and 98.50% F1-score, outperforming existing methods. These results demonstrate that the GAN-based denoising approach, combined with traditional feature-based classification, offers a robust, efficient, and practical solution for modern agricultural disease monitoring systems.

Corresponding Author:

Anandraddi Naduvinamani
Department of Electronics and Communication Engineering, S. G. Balekundri Institute of Technology
Affiliated to Visvesvaraya Technological University
Belagavi, India
Email: anandreddi02@gmail.com

1. INTRODUCTION

Agriculture has a major impact in ensuring food security, poverty alleviation, and overall development [1]. In the context of India, it holds significant importance in the country's economy, providing employment for a large portion of the population and contributing approximately 15% to the nation's gross domestic product (GDP). India is one of the world's leading producers of various crops, including rice, wheat, sugarcane, cotton, and pulses, further emphasizing it is importance on the global stage. The global population is expected to reach 9.7 billion by 2050 and 11.2 billion by the end of this century, leading to a 1.6% annual increase in the Earth's population and a corresponding rise in demand for plant products [2], [3].

Safeguarding crops against diseases is therefore essential to meet the growing demand for both the quality and quantity of food. Plant diseases alone incur significant economic losses, amounting to around US$220 billion annually worldwide [4].

In the context of India, the Indian Council of Agricultural Research reported that more than 35% of crop production is lost each year due to pests and diseases [5]. These outbreaks not only threaten food security but also have far-reaching economic, social, and environmental consequences. Protecting crops from these threats is vital not only for the agricultural sector but also for the overall well-being of communities and the sustainability of the environment. Plant diseases are considered one of the major concerns in the agricultural domain because they lead to a decline in crop quality and a reduction in overall production. The impacts of these diseases range from minor symptoms causing minimal damage to severe outbreaks that may affect entire regions of cultivated land. This damage results in substantial financial losses and significantly affects the economy, particularly in developing nations that rely on a single crop or just a few crops. Efforts to prevent major agricultural losses have led to the development of various diagnostic methods for plant diseases. Molecular biology and immunology techniques offer accurate detection of disease-causing agents, but these methods often require specialized knowledge and significant financial resources, making them inaccessible to many farmers [6]. Notably, the majority of the world's farms are small, family-operated ventures in developing countries, producing food for a significant portion of the global population. Thus, making such methods accessible to these farmers remains a challenging issue.

Rice remains in the top list of most widely consumed foods globally, with a total consumption of 493.13 million metric tons in 2019-2020 and 486.62 million metric tons in 2018-2019 [7]. These figures reflect a growth in rice consumption over the years. As consumption increases, it is expected that production rates will keep pace. However, inadequate monitoring of farmland has often led to significant rice losses due to disease-related issues. Various diseases commonly affect rice cultivation, causing substantial economic losses. Furthermore, the excessive use of chemicals like bactericides, and fungicides to combat plant diseases has negatively impacted the agro-ecosystem [8]. Figure 1 depicts the trends of total rice consumption worldwide.

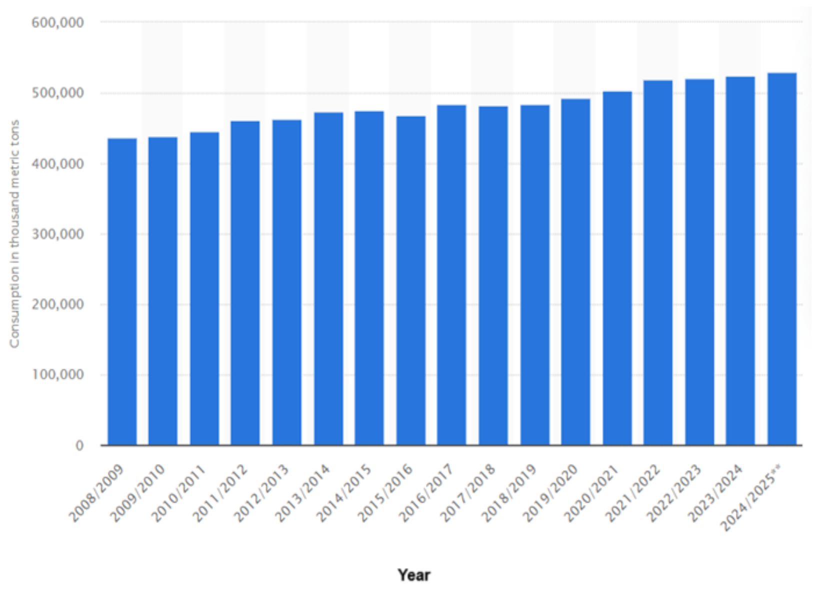

Figure 1. Total rice consumption (worldwide)

The early prediction and forecasting of rice leaf diseases play a significant role in maintaining the quality and quantity of rice production. Timely detection of diseases allows for early intervention, which helps control disease progression and promotes the healthy growth of plants, ultimately improving rice yield and supply [9]. Common rice diseases include sheath blight, bacterial blight, and rice blast, each characterized by distinct symptoms related to texture, color, and shape. These diseases tend to spread rapidly and are easily transmissible. Currently, artificial identification, disease mapping, and automated detection are considered key methods for identifying rice diseases. Figure 2 shows sample images of rice diseases where rice stackburn, leaf smut, leaf scald, false smut, blast, stem rotm white tip, sheath rot, stripe blight, sheath blight, bacterial leaf streak, and brown spot images are depicted in Figures 2(a) to 2(l), respectively.

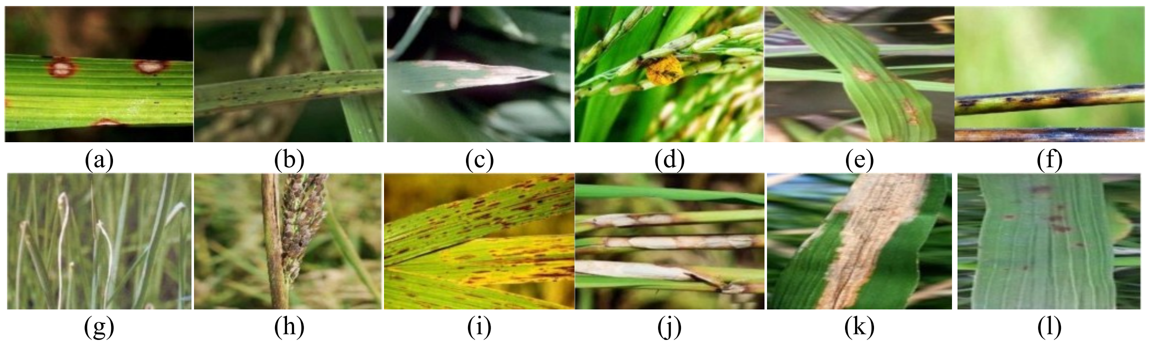

Figure 2. Sample images of rice disease of (a) rice stackburn, (b) rice leaf smut, (c) rice leaf scald, (d) rice false smut, (e) rice blast, (f) rice stem rot, (g) rice white tip, (h) rice sheath rot, (i) rice stripe blight, (j) rice sheath blight, (k) bacterial leaf streak, and (l) rice brown spot

Similarly, cotton is an important crop globally. It is a vital cash crop that supports the livelihoods of millions of people across Asia, Africa, Australia, and the Americas. Its fiber is a key resource, contributing to the income of both farmers and industrialists. Cultivated for thousands of years, cotton has been an essential part of textile production worldwide and is an integral component of farming systems in approximately 60 countries. The leading producers of cotton are China, the United States, and India. In the Indian context, 23% of the cotton produced is exported internationally. However, cotton yields are greatly influenced by crop growth, which can be significantly affected by various diseases. Figure 3 shows sample images of cotton leaf diseases where healthy, leaf spot, nutrient deficiency, powdery mildew, target sport, veticillium wilt, and leaf curl samples are depicted from Figures 3(a) to 3(g), respectively.

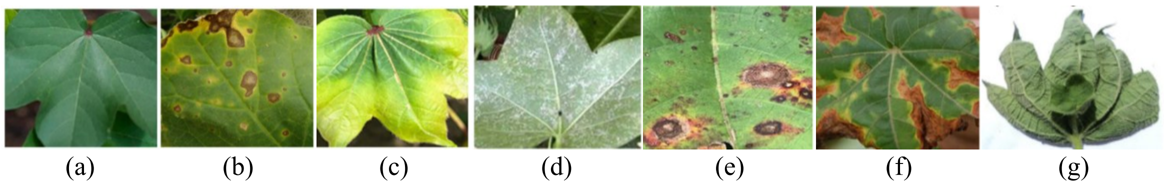

Figure 3. Sample cotton leaf image of (a) healthy, (b) leaf spot, (c) nutrient deficiency, (d) powdery mildew, (e) target spot, (f) vericillium wilt, and (g) leaf curl

In response to these challenges, extensive research has focused on creating accurate and accessible methods for the majority of farmers. Precision agriculture, leveraging cutting-edge technology, optimizes decision-making processes. Modern digital technologies enable real-time data collection, and machine learning (ML) algorithms aid in providing optimal decisions, thereby reducing costs. Nevertheless, there is still room for improvement, particularly in refining decision-support systems capable of transforming vast amounts of data into practical recommendations.

The current technological advancements have suggested to adopt the computer vision-based ML based automated approaches for early detection of plant disease detection. Several methods have been developed based on this concept of image processing [10]. Image processing is a specialized field within signal processing that focuses on extracting valuable information from images. ML, a subset of artificial intelligence (AI), enables automation and provides instructions to perform specific tasks [11]. The main aim of ML is to realize training data and create models that can assist people by facilitating sound decision-making and predicting accurate outcomes based on extensive training data. In the context of plant health, various image properties such as leaf color, extent of damage, leaf area, and texture parameters are utilized for classification purposes. In our project, we have meticulously analyzed different image parameters and features to identify various diseases affecting plant leaves, aiming for the highest accuracy possible. Traditionally, plant disease detection relied on visual inspections of leaves or involved chemical processes conducted by experts. However, this approach demanded a large team of specialists and continuous plant observation, making it costly, especially for large farms. In such scenarios, our proposed system proves invaluable for monitoring vast agricultural fields. By automatically detecting diseases based on observable symptoms on plant leaves, this method becomes not only more straightforward but also significantly more cost-effective. Therefore, we adopt the ML based approach for plant disease detection. The main contribution of this work are as follows: first of all, we focus on image pre-processing stage and present generative adversarial network (GAN) based model for image denoising to improve the image quality. In next stage we focus on feature extraction phase where color, shape and gradient based features are extracted and combined together to formulate the final feature vector. Finally, we adopt decision tree (DT) classifier model to perform the classification of rice and cotton leaf disease.

2. LITERATURE SURVEY

This section presents the overview of existing methods and focus on identifying the drawbacks and challenges in the existing schemes. Vishnoi *et al.* [12] reported that plant leaves are used to detect the various infections in the plants. Authors used computer vision with soft computing methods and suggested to incorporate efficient feature extraction method. In this line of research, Sahu and Pandey [13] presented an optimized approach by using hybrid multiclass support vector machine (SVM) model for plant leaf disease detection. The classification model also uses random forest (RF) model to produce the final hybrid classification. Prior to feature extraction, spatial fuzzy c-means is employed to obtain the segmentation. The segmented region of interest (ROI) is used further for feature extraction. As discussed before, feature extraction is an important phase in these types of computer vision tasks therefore, Ahmad *et al.* [14] focused on developing a new feature extraction method and introduced a new feature extraction method known as local triangular-ternary pattern (LTriTP). In implementing the ternary pattern approach, a dynamic threshold based on the absolute mean value is calculated. This method enables a sensitive analysis of texture information in plant leaf images, generating a wide range of highly pertinent values. As plant diseases can manifest at various orientations on a leaf image, a histogram of the gradient is computed in four directions (0°, 45°, 90°, and 135°) within each triangle. This process helps identify the gradient changes in infected regions compared to healthy areas. Kartikeyan and Shrivastava [15] suggested to adopt image pre-processing approach and introduced hybrid feature extraction approach where discrete wavelet transform (DWT) and gray level co-occurrence matrix (GLCM) attributes are fused to produce the final robust feature vector. Finally, the SVM is used to train the model based on extracted features and labelled images.

Kulkarni *et al.* [16] presented ML based approach which is carried out into multiple stages: first stage demonstrates the pre-processing phase where grayscale conversion, filtering and thresholding steps are employed. The obtained pre-processed image is then processed through the feature extraction phase where morphological, GLCM and color features are extracted and processed through the feature selection process. Finally, a classification model is deployed to train the model by using the selected features. Kumar *et al.* [17] introduced ML based approach for plant disease detection where image contrast enhancement, segmentation and feature extraction steps are employed prior to perform the classification. Image segmentation is done by using K-means clustering where as GLCM model is used for feature extraction. Finally, SVM classifier is used to perform the classification task. Alagumariappan *et al.* [18] presented a ML based approach which considers Hu moments and Haralick texture features for feature analysis and later these features are fed to the extreme learning machine (ELM) and SVM classification. The SVM uses linear and polynomial kernel functions to obtain the final outcome. Pallathadka *et al.* [19] used histogram-based image equalization as pre-processing and principal component analysis (PCA) for feature extraction and finally different classification methods are used for classification. Sarangdhar and Pawar [20] used SVM based regression method for plant disease detection and developed an android application combined with IoT facilities. Jaisakthi *et al.* [21] presented a ML based disease detection approach for disease detection in grapes. In first phase, this approach uses global thresholding and semi-supervised model for segmentation to identify the ROI. Finally, SVM, adaptive boosting (AdaBoost) and RF tree classifiers are used to obtain the classification outcome. Roy *et al.* [22] reported that existing methods suffer from several issues and suggested to incorporate dimensionality reduction methods to improve the classification performance.

Isinkaye *et al.* [23] presented a combination of the variational autoencoder (VAE) and vision transformer (ViT) framework that classifies multi-class plant diseases. The VAE implemented dimensionality reduction in images maintaining important features, and ViT allowed extracting global features. On the dataset PlantVillage (corn, potato, and tomato), their model performs with a 93.2% accuracy rate, which shows an added benefit of robustness and larger scale in disease classification. Prasannakumar and Latha [24] developed contextual mask autoencoder (CMAE) whose optimization was designed on the basis of the dynamic differential annealed optimization algorithm (DDAOA). This plant disease identification (PDI)-CMAE-DDAOA model using this combination of adaptive noise filtering, statistical feature extraction, and cosine similarity enabled hidden Markov model (HMM) to outperform most current models in terms of accuracy, F1-score, and even specificity. This strategy was also focused on the essence of optimization to accurately categorize the plant disease.

Cui *et al.* [25] have improved the maize disease identification with a convolutional block attention module (CBAM) integrated lightweight autoencoder. Preprocessing was done using DWT whereas CBAM enhanced spatial and channel attention of features. The model achieved 99.44% accuracy, and it surpassed the performance of some deep convolutional neural network (CNN) structures (e.g., ResNet-50 and DenseNet201), thus being unexpectedly efficient and interpretable. In segmentation and classification of plant leaf diseases, Abinaya *et al.* [26] designed cascading autoencoder with attention residual U-Net (CAAR-UNet). It is trained with custom dataset and reached the accuracy of 95.26% of mean pixel and 0.7451 of mean intersection over union (IoU), indicating high success in the location and boundary definition of disease which is critical to early diagnosis.

3. PROPOSED MODEL

This section presents the complete proposed approach where we have used image pre-processing, feature extraction and supervised classification to obtain the final result. The pre-processing module uses GAN approach to enhance the image quality before processing it through the feature extraction model. This obtinaed image is processed through the feature extraction phase where color, texture and gradient features are extracted to formulate the final feature vector. Finally, a classification model is presented to learn the pattern from the attribute and train the model to classify the test data.

3.1. Compreheive overview of proposed model

This work presents a denoising model proposed as sparse autoencoder-based generative adversarial network (Sparse-GAN) to improve the image quality while maintaining the necessary structural features in the process. The generator of this framework is represented as a convolutional sparse autoencoder formed with an encoder-decoder modality. The encoder consists of stack convolutional layers and pooling layers to gradually learn higher level features and reduce the spatial dimensions of the input to a more structural latent representation. The important element is the sparse fully connected latent layer, which adds L1 regularization penalty on the condition that only a part of the neurons is activated. This sparsity scheme is useful in removing useless or noisy features but allows retention of important visual information and hence it performs well on denoising problems. The image is symmetrically reconstructed by the decoder composed of upsampling and convolutional layers with a goal to recover fine-grained details based on the compressed sparse representation. Discriminator is a convolutional binary classifier that judges the realism of generated images and instructs the generator to generate output that cannot be discriminated against a true clean sample. This incorporation of the sparse autoencoder within the GAN model enables the model to achieve accuracy at the pixel-level and at the same time perceptual realism. All the elements of the outlined architecture serve a specific and very important purpose: the encoder compresses noisy data into an efficient code, the sparsity penalty removes noise by making features selective, the decoder renders clean images with retained structure, and the adversarial supervision yields visual quality. These blocks can be used jointly to achieve strong denoising with minimal loss of semantic level to the input images and, hence, render the model exceptionally well to downstream tasks of image analysis and image classification.

3.2. Pre-processing

The main aim of this section is to perform image enhancement to improve the quality of input image. Generally, during the image capturing process, the images get contaminated due to several factors such as low illumination, camera shake, and motion blur. which may impact the quality of analysis. Therefore, this issue needs to be tackled. To overcome this issue, image filtering/denoising is considered as one of the important aspects where the unwanted noisy data is removed from the image. In this work, we adopt deep learning-based mechanism to perform the image filtering task. Therefore, we present a sparse GAN architecture to perform image filtering.

GANs are a class of ML which are widely used in various computer vision related tasks. The GAN models consist of two neural networks (NN) as generator and discriminator which are trained with the help of adversarial training method. While GANs are primarily known for generating new data samples, they can also be utilized for tasks like image denoising. The complete process of GA for image denoising task is as follows.

3.2.1. Generator

The generator takes a noisy image x as input and tries to generate a denoised image $G(x)$. The generator can be represented by a function $G: \mathbb{R}^{W\times H\times C} \to \mathbb{R}^{W\times H\times C}$, where W, H, and C represent the width, height, and number of channels of the images, respectively. Figure 4 presents a generator architecture based on GAN model. The architectural details of generator block are presented in Table 1.

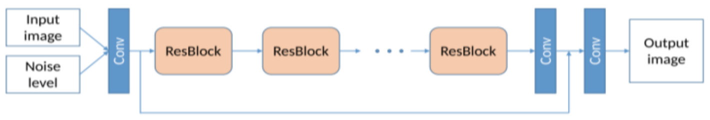

Figure 4. GAN based generator block for denoising

Table 1. Architecture details of generator block

Layer	Type	Kernel size	Stride	Activation	Output size
1	Conv2D	3×3	1	ReLU	256×256×64
2	Conv2D	3×3	1	ReLU	256×256×64
3	Conv2D	3×3	2	ReLU	128×128×128
4	Conv2D	3×3	1	ReLU	128×128×128
5	Conv2D	3×3	2	ReLU	64×64×256
6	Conv2D	3×3	1	ReLU	64×64×256
7	Transposed Conv2D	3×3	2	ReLU	128×128×128
8	Transposed Conv2D	3×3	2	ReLU	256×256×64
9	Conv2D	3×3	1	Tanh	256×256×C

3.2.2. Discriminator

Similarly, discriminator tries to distinguish between real clean images (y) and generated denoised images $(G(x))$. It considers the input image which is generated by generator block and focus on producing the discriminated output according to the training procedure: i) input: either a real clean image y or a generated denoised image $G(x)$ and ii) output: probability score indicating the authenticity of the input image (real or fake). The discriminator can be represented by a function $D: \mathbb{R}^{W\times H\times C} \to [0,1]$ where [0,1] [0,1] represents the probability score output. Figure 5 presents the descriminator architecture to obtain the final denoised image. The architectural details of this discriminator block are presented in Table 2.

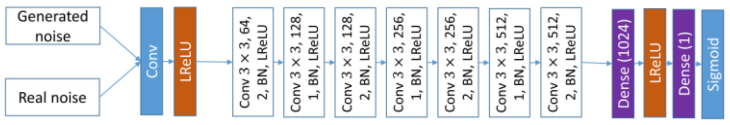
Generated noise
Real noise
Conv
LReLU
Conv 3 × 3, 64, 2, BN, LReLU
Conv 3 × 3, 128, 1, BN, LReLU
Conv 3 × 3, 128, 2, BN, LReLU
Conv 3 × 3, 256, 1, BN, LReLU
Conv 3 × 3, 256, 2, BN, LReLU
Conv 3 × 3, 512, 1, BN, LReLU
Conv 3 × 3, 512, 2, BN, LReLU
Dense (1024)
LReLU
Dense (1)
Sigmoid

Figure 5. Discriminator for distinguishing original clean and generated denoised image

Table 2. Architectural details of discriminator block

Layer	Type	Kernel size	Stride	Activation	Output size
1	Conv2D	4×4	2	LeakyReLU (0.2)	128×128×64
2	Conv2D	4×4	2	LeakyReLU (0.2)	64×64×128
3	Conv2D	4×4	2	LeakyReLU (0.2)	32×32×256
4	Conv2D	4×4	2	LeakyReLU (0.2)	16×16×512
5	Conv2D	4×4	1	LeakyReLU (0.2)	15×15×1
6	Flatten+dense	-	-	Sigmoid	1

3.2.3. Loss function

Loss function establishes the goal which guides the training of both the discriminator network and the generator network. GANs have the goal of ensuring that the samples generated by the generator are indistinguishable to actual data, and the goal of the discriminator is to ensure that real and generated samples are distinguishable. The loss function that is applied to the generator is defined in such a way that it reduces the probability of the discriminator to correctly recognize the generated samples as fake. Thus, the objective of the generator module is to minimize the log probability that the discriminator makes a mistake.

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_y{\sim}p_{data\,(y)}[\log D(y)] + E_x{\sim}p_{data\,(x)}\left[\log\left(1 - D\big(G(x)\big)\right)\right] \tag{1}$$

Where $p_{data}(y)$ is the distribution of real clean images, $p_{data}(x)$ is the distribution of noisy images, $D(y)$ is the output of the discriminator for real clean images and $D(G(x))$ is the output of the discriminator for generated denoised images. Similarly, denoising loss measures the difference between the generated denoised image $(G(x))$ and the clean target image (y). We use mean squared error (MSE) loss for this purpose which is expressed as (2).

$$\mathcal{L}_{MSE}(G) = \mathbb{E}_{x\sim p_{data}(x), y\sim p_{data}(y)}[\|G(x) - y\|_2^2] \quad (2)$$

3.3. Feature extraction

This presents the detailed discussion about proposed approach for feature extraction. This approach is based on three different feature extraction methods and combined the obtained features to formulate the final feature vector. The feature extraction includes color, texture and improved scale-invariant feature transform (SIFT) and histogram of oriented gradients (HOG) feature extraction by using gradient correlations.

3.3.1. Color feature

This subsection presents the more advanced mathematical model for color feature extraction involves the use of color moments. Color moments features represent the statistical measures of the considered image and used to describe the distribution of colors in the input image. Algorithm 1 demonstrates the process to compute the color feature extraction using color moments.

Algorithm 1. Color feature extraction using color moments
Step 1: image pre-processing
1: Input: consider the plant leaf image as input image I with dimension $M \times N$.
2: Pre-process: perform different steps such as resize, normalize and CNN based denoising model.
Step 2: color moment computation
1: Calculate the color moments for each channel. The color moments are calculated as follows:
- First order moment (mean):

$$\mu_i = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N} I(x,y)^i \quad (3)$$

- Second order moments (variance):

$$\sigma_i^2 = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[I(x,y) - \mu_1]^i \quad (4)$$

- Third order moment:

$$v_i = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}[I(x,y) - \mu_1]^i \quad (5)$$

where i represents the moment order (1 represents mean 2 represents variance and 3 represents the skewness).

Step 3: feature vector formulation
- In order to create the final feature vector, we arrange all features as (6).

$$F = [\mu_L, \mu_L^2, v_L, \mu_a, \sigma_a^2, v_a, \mu_b, \sigma_b^2, v_b\ ] \quad (6)$$

3.3.2. Texture feature extraction

Texture feature extraction from images often involves analyzing patterns and variations in pixel intensities. One widely used method for texture feature extraction is the GLCM. GLCM calculates the occurrences of pixel pairs with specific values and distances in an image, capturing texture patterns. Algorithm 2 shows the process to extract the texture feature vector.

Algorithm 2. Texture feature extraction using GLCM
Step 1: image pre-processing
1: Input: consider the plant leaf image as input image I with dimension $M \times N$.
2: Pre-process: perform different steps such as resize, normalize and CNN based denoising model.
Step 2: GLCM computation
- Choose a set of displacement vectors (d_x, d_y) to define pixel pairs' distances and angles (e.g., (1,0) (1,0) for horizontal, (1,1) (1,1) for diagonal).
- For each pixel (x, y) in the image, calculate the GLCM values by considering the pixel pairs $(I(x,y), I(x + dx, y + dy))$ within the specified displacement.
- Count the occurrences of these pixel pairs and construct a GLCM for each displacement vector.
- Normalize the GLCM matrices to obtain probabilities of occurrence.

Step 3: feature extraction
- Contrast: measures the local variations in the image $\sum_i \sum_j (i-j)^2 \times GLCM(i,j)$.

– Entropy: measures the randomness or complexity of the texture:

$$Entropy = -\sum_i \sum_j GLCM(i,j) \times \log(GLCM(i,j)) \quad (7)$$

– Homogeneity: measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal:

$$Homogenity = \sum_i \sum_j \frac{GLCM(i,j)}{1+|i-j|} \quad (8)$$

– Correlation: measures the linear dependency between the pixel pairs in the GLCM:

$$Correlation = \sum_i \sum_j \frac{(i-\mu)(j-\mu)\times GLCM(i,j)}{\sigma^2} \quad (9)$$

where μ is the mean and σ is the standard deviation of the GLCM.

3.3.3. Gradient based feature extraction for improving the HOG and SIFT feature

This section describes the proposed mechanism for improved HOG and SIFT feature extraction for plant disease detection by using ML approach. This model can be considered as an extension of HoG or SIFT from 1st to 2nd order statistics. According to this approach the gradients of image are represented sparsely corresponding to their magnitude and orientations. Let us consider that image region is presented as I and its position vector is represented as $r = (x,y)^t$. The gradient of this image is represented as $\left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)^t$ which can be expressed in terms of magnitude at each pixel as $n = \sqrt{\frac{\partial I^2}{\partial x} + \frac{\partial I^2}{\partial y}}$ and its corresponding orientation angle can be expressed as $\theta = \arctan\left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$. In this approach, the obtained orientations θ is coded into D orientation bins with the help of assigning weights to the nearest bins. This is defined as sparse vector which is expressed as $f(\in R^D)$ and known as gradient orientation vector. With the help of gradient vector f and magnitude n, the N^{th} order autocorrelation gradient function in local neighbour can be expressed as (10).

$$R(d_0, \dots, d_N, a_1, \dots, a_N) = \int_I w[n(r), n(r+a_1), \dots, n(r+a_N)] f_{d_0}(r) f_{d_1}(r+a_1) \dots f_{d_N}(r+a_N) dr \quad (10)$$

Where a_i represents the displacement vector from the considered reference point r, f_d represents the d^{th} element of f and w is the scalar weighting function. This mathematical expression demonstrates two types of gradient correlation which are spatial and orientational correlation. The spatial correlation is derived from the displacement vector and orientational correlation is obtained from product of elements of f_{d_i}. Based on this, the gradient based local correlation are computed as (11) and (12).

$$0^{th}\ order\ R_{N=0}(d_0) = \sum_{r\in I} n(r) f_{d_0}(r) \quad (11)$$

$$1^{st}\ order\ R_{N=1}(d_0, d_1, a_1) = \sum_{r\in I} \min[n(r), n(r+a)] f_{d_0}(r) f_{d_1}(r+a) \quad (12)$$

The above-mentioned 0^{th} order gradient represents the histogram of gradients used in SIFT and HOG whereas the 1st order correlation can be considered as joint histogram of orientation pairs. Moreover, the 1st order orientation features also characterize the image contours.

3.4. Classification model

This work has adopted the supervised classification approach to obtain the final classification. In this work, we have adopted DT classification for multiclass supervised classification. Algorithm 3 demonstrates the classification algorithm. According to this approach, X is considered as a feature matrix with n samples and m features which is represented as $X = [x_1, x_2, \dots x_n]$ where x_i is the feature vector of i^{th} sample. Similarly, y is the vector of labels with n elements which is expressed as $y = [y_1, y_2, \dots, y_n]$ where y_i is the label of i^{th} sample. These vectors are arranged as $D = (X, y)$ to formulate the dataset and T is the tree model.

```
Algorithm 3. DT algorithm
Step 1: initialization: create a node $N$ at the root of the tree.
Step 2: splitting criteria: select the best feature $F$ to split the data at node $N$. This is done by evaluating a splitting criterion such as Gini impurity or Entropy. Let $T_F$ be the set of possible feature values for feature $F$.
```

Step 3: split the data: partition the dataset D into subsets $D_1, D_2, \dots, D_K$ based on the values in T_F.
Step 4: recursive splitting: for each subset D_i, create a child node N_i and repeat steps 2-3 until a stopping criterion is met (e.g., maximum depth and minimum samples per leaf).
Step 4: assign labels: if a stopping criterion is met, assign a label y_i to node N_i based on the majority class in D_i. If all samples in D_i have the same class, N_i becomes a leaf node.
Step 5: prediction: to classify a new sample x, traverse the tree from the root, following the appropriate branches based on the feature values of x. This process is repeated until the leaf node is reached. Finally, the leaf node class is assigned as predicted class of x.

The Gini index is used in this work as the splitting criterion in the decision-making process of the model. The Gini index is used to measure the impurity of a dataset and it is also used to determine the best split at each node It can be expressed as (13).

$$Ginni\ (D) = 1 - \sum_{i=1}^{k} (p_i)^2 \tag{13}$$

Where, k is the number of classes, and p_i is the proportion of samples in class i in dataset D. Similalry, the entropy measure is applied as a classification criterion in this work. Entropy measures the uncertainty or disorder in the dataset and can be applied to determine the best attribute to split using the one that offers the greatest information gain. It can be expressed as (14).

$$Entropy\ (D) = -\sum_{i=1}^{k} p_i \log_2(p_i) \tag{14}$$

Where, k is the number of classes, and p_i is the proportion of samples in class i in dataset D.

4. RESULTS AND DISCUSSION

This section presents the outcome of proposed approach and compares it performance with existing approaches to shows the robustness of proposed model. First of all, we describe theexperiemntal setup followed by the dataset details and later performance measurement parameters are described. Finally, comparative analysis is presented.

4.1. Experimental setup

Experiments were performed in the same framework on a PC powered by Windows 11, equipped with Intel Core i7, 16 GB of RAM and NVIDIA RTX 3060 (6 GB VRAM). It was implemented on Python 3.10, and the main deep learning frameworks comprise TensorFlow 2.11 and Keras in building and training the models. Other libraries that were utilized included NumPy, Pandas, OpenCV, Scikit-learn, Matplotlib, and Albumentations to manipulate data, perform data augmentation, and visualize the data. Input images were scaled to the size of 256×256 pixels and the models were trained up to 100 epochs with a batch size of 8 using Adam optimization algorithm with an initial learning rate of 0.001 which was adjusted dynamically through learning rate scheduler. The loss function was categorical cross-entropy. To avoid the issue of overfitting, early stopping was performed, which occurred in the range of 60-80 epochs.

The proposed GAN-based denoising architecture includes a generator and a discriminator aimed at improving noisy agricultural images. The generator trains the noisy images to the denoised results using six convolutional layers with rectified linear unit (ReLU) activation, skip connections, and two transposed convolution layers to upsample images, and then Tanh activation is used to obtain the denoised image in the final layer. The discriminator includes five convolutional blocks with a LeakyReLU activation and a batch normalization followed by a sigmoid output to classify between real and generated image, and dropout (0.3) is added to avoid overfitting. The networks are trained, with the help of both adversarial loss and MSE loss, optimized using Adam optimizer (learning rate=0.0002, batch size=16) through 100 epochs. Image size is reduced to 256×256 pixels during dataset preprocessing, which is then augmented by using rotation, flipping, and brightness manipulation to enhance generalization.

In order to achieve the blance between accuracy and perceptual fidelity, the overall objective function of proposed model is formed as weighte combination of adversarial loss and MSE loss. The adversarial loss helps to encourage the generator to produce realistic denoised image while MSE loss ensures the structural consistency and pixel level similalrtiy.

$$\mathcal{L}_{total}(G, D) = \lambda_{adv}\mathcal{L}_{GAN}(G, D) + \lambda_{MSE}\mathcal{L}_{MSE}(G) \tag{15}$$

Where $\mathcal{L}(G, D)$ is the adversarial loss which is defined as (16).

$$\mathcal{L}_{GAN}(G,D) = \mathbb{E}_{y\sim p_{data}(y)}[\log D(y)] + \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(1 - D\big(G(x)\big)\right)\right] \quad (16)$$

$\mathcal{L}_{MSE}(G)$ represents the mean squared reconstruction loss which is represented as (17).

$$\mathcal{L}_{MSE}(G) = \mathbb{E}_{x,y\sim p_{data}}[\|G(x) - y\|_2^2] \quad (17)$$

λ_{adv} and λ_{mse} represents the weighting coefficents which are used to balance the perceptual and pixel-wise losses.

4.2. Dataset details

This section presents the brief discussion about the dataset used in this work where we have used rice leaf disease and cotton leaf disease dataset. The rice dataset consists of images of rice leaves which are affected by several diseases such as bacterial leaf blight, brown spot, and others. Similarly, the cotton leaf data include leaf curl, bacterial blight, and more.

4.2.1. Rice plant dataset

The rice leaf disease dataset is obtained by comprehensive compilation of information based on the three common rice disease which considers bacterial blight, brown spot, and leaf smut. This dataset is designed to support researchers, agronomists, and ML experts in analysing, diagnosing, and possibly forecasting the appearance of these diseases by utilizing a range of attributes and parameters. The dataset was compiled from various secondary sources, including several well-known online repositories. Table 3 provides a detailed overview of these sources, which include Mendeley [27], Kaggle [28], UCI [29], and GitHub [30]. Specifically, 1,584, 40, 40, and 192 images of bacterial blight were sourced from Mendeley, Kaggle, UCI, and GitHub, respectively. For leaf smut, 40 images were obtained from Kaggle and another 40 from UCI. Additionally, images of rice blast infection were gathered from Mendeley and GitHub. Altogether, the dataset comprises 3,535 images of diseased rice leaves. Images were collected under diverse lighting conditions, varying backgrounds, and with different camera devices. The images were resized to a uniform size of 224×224 during preprocessing. To ensure effective training and unbiased evaluation, the dataset was randomly partitioned into: 70% training, 15% validation, and 15% testing. This stratified split was applied per class to maintain class balance across all subsets.

Table 3. Dataset details for rice plant

Disease type	Online repository			
	GithHub	Kaggle	UCI	Mendeley
Bacterial blight	192	40	40	1,584
Leaf smut		40	40	
Rice blast	159			1,440

4.2.2. Cotton leaf disease

A dataset featuring images of both healthy and diseased cotton leaves and plants has been sourced from Kaggle.com, specifically from a competition organized by D3V [31]. This dataset includes four categories of images: fresh cotton leaves, fresh cotton plants, diseased cotton leaves, and diseased cotton plants, totaling 2,310 images. For instance, images labeled “2” and “3” represent fresh and diseased samples, respectively, within this dataset. Image resolutions is obtained as 256×256 pixels, and images were taken in natural lighting across different times of day, introducing variation in illumination and background. The dataset was partitioned as follows: 70% training, 15% validation, and 15% testing.

4.3. Performance evaluation parameters

This work mainly focused on plant image denoise and plant leaf disease classification by using computer vision and ML based solutions. Therefore, we discuss image denosing performance evaluation and classification performance measurement methods in this section. The image denoising performance is measured in terms of peak signal-to-noise ratio (PSNR) and MSE. On the other hand, the classification performance is measured in terms of accuracy, sensitivity, specificity, and F1-score. The first subsection describes the methods used to measure the denoising performance and next subsection describes the parameters used to evaluate the classification performance.

4.3.1. Image denoising performance measurement parameters

The image denoising performance is measured by comparing the filtered image with the actual input image given to the filtering module. The PSNR is the common metric used to assess the performance of these tasks. It compares the quality of denoised image to the original image. The higher the PSNR value, the better the quality of the denoised image, as it indicates less distortion and loss of information. It can be computed as follows (18).

$$PSNR = 10\ \log_{10}\left(\frac{255^2}{MSE}\right) \tag{18}$$

Similalry, MSE is used to measure the average of the squares of the errors or deviations between the denoised and the original image. It can be expressed as (19).

$$MSE = \frac{\sum_i \sum_j \left(Y(i,j) - \hat{Y}(i,j)\right)^2}{M \times N} \tag{19}$$

Besides the above-mentioned metrics we also utilize the area under the receiver operating characteristic curve (AUC-ROC) to give more blended interpretation especially when we have an imbalanced dataset. AUC-ROC checks the capability of the model to distinguish classes at different threshold values. The ROC It is a graphical display of the true positive rate (sensitivity) and the false positive rate (1-specificity). The closer the AUC value is to 1 the more the model is able to discriminate between positive and negative classes. AUC becomes especially helpful in validation of skewed class distributed classification models since it is insensitive to class imbalances and allows better comparisons of classifiers.

4.3.2. Classification performance

This section describes the methods used to measure the overall performance of the plant disease classification model. The performance of the classifier mode is evaluated in terms of accuracy, sensitivity, specificity, and F1-score which are obtained with the help of confusion matrix. The standard representation of confusion matrix is presented in Table 4.

Table 4. Binary class confusion matrix

	Positive	Negative	Total
Positive	T_p	F_p	$T_p + F_p$
Negative	F_N	T_N	$F_N + T_N$
Total	$T_P + F_N$	$T_P + T_N$	

With the help of this confusion matrix, we compute other performance measurement parameters including accuracy, sensitivity, specificity and F1-score. These parameters are computed with the help of true positive, false positive, true negative false negative and false negative. The computation can be expressed as (20)-(23).

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{20}$$

$$Sensitivity = \frac{T_P}{T_P + F_N} \tag{21}$$

$$Specificity = \frac{T_N}{T_N + F_P} \tag{22}$$

$$F - measure = \frac{2 \times T_P}{2 \times T_P + F_N + F_P} \tag{23}$$

4.3.3. Denoising performance analysis

This section presents the outcome of proposed denoising approach and compares it is performance with state-of-art standard filtering approaches. We have considered two different types of noises which are Gaussian and speckle noise which are added at different levels to the original image. In this work, we have combined both the dataset and added this noise with different level of parameters. Table 5 shows the RMSE performance for Gaussian noise for varied noise levels and performance of different filtering methods.

Based on the RMSE performance, we measured the PNSR performance for varied levels of Gaussian noise. In this experiment, we have varied the noise ratio as 10, 20, 30, 40, and 50% to evaluate the robustness for different levels of noise. Table 6 shows the obtained performance and it is comparative analysis with existing filtering methods.

Table 5. RMSE performance for Gaussian noise

Filtering method	10%	20%	30%	40%	50%
Mean	25.30	52.20	77.50	92.30	107.50
Median	26.30	51.50	78.25	97.50	110.25
Wiener	27.50	52.80	73.80	91.20	104.30
Wavelet	28.50	53.50	74.60	93.50	107.55
Curvelet	150.30	180.45	210.50	210.15	243.20
Proposed model	17.90	23.80	27.80	31.50	33.20

Table 6. PSNR performance for Gaussian noise

Filtering method	10%	20%	30%	40%	50%
Mean	20.25	14.80	12.50	9.5	7.47
Median	19.80	15.60	11.50	9.80	7.13
Wiener	19.60	15.90	12.30	9.20	7.46
Wavelet	19.30	12.80	13.50	8.5	7.45
Curvelet	8.25	7.20	7.10	8.1	4.25
Proposed model	36.80	35.50	31.20	28.50	26.95

This experimental analysis shows the increased noise levels affects the denoising performance which may lead to increase the misclassification and reduce the classification accuracy. However, the proposed approach reported the significant increase in PSNR and reduction in RMSE. Further, we extend this experiment for speckle and measured the RMSE and PSNR performance for varied levels of noise. Table 7 shows the RMSE performance for speckle noise scenario.

Table 7. RMSE performance for speckle noise

Filtering method	10%	20%	30%	40%	50%
Mean	18.25	24.50	27.30	29.92	32.21
Median	25.30	35.80	45.80	46.22	51.05
Wiener	20.50	28.30	36.30	34.20	36.31
Wavelet	17.25	20.40	25.10	24.84	26.63
Curvelet	135.30	120.25	135.20	152.82	155.05
Proposed model	16.20	23.25	23.50	25.55	30.58

For this experimental setup, we measure the performance in terms of PSNR and compared the obtained performance with existing schemes. In this experiment also, the noise levels are varied from 10% to 50% to evaluate the performance of proposed approach. Table 8 shows the PSNR analysis. According to this experiment, the proposed approach outperformed existing methods however, the increase in noise levels affects the filtering performance but proposed model is able to achieve substantial increase in PSNR performance. Mean filtering is sensitive to outliers and can blur edges and fine details in the image. It is not effective for images with non-uniform noise or impulse noise. On the other hand, the median filtering can remove impulse noise effectively, but it may not work well for Gaussian or other types of noise. It preserves edges better than mean filtering but can still cause blurring, especially for images with fine structures. Wiener filtering assumes a specific noise model and requires knowledge of the power spectral density of both the signal and the noise. It is sensitive to the accuracy of the noise model and may not perform well if the noise characteristics deviate from the assumed model. Wavelet filtering can effectively remove noise and preserve edges, but it may introduce artifacts known as "ringing" artifacts around sharp edges in the image. The choice of wavelet basis and decomposition level significantly impacts the filtering results. Curvelet filtering is computationally intensive and requires more complex algorithms than wavelet filtering. While it can handle highly directional features better than wavelets, it may not always provide significant advantages for images with general noise. Further, we compared the performance of proposed model with state-of-art methods in terms of PSNR for valried noise levels.

The proposed denoising method is measured in PSNR and SSIM and compared to the classical and deep learning-based algorithms, including BM3D, UDWT, DnCNN, FFDNet, IRCNN, LSLA-2, and the original GAN model. Based on Table 9, it may be noted that traditional techniques can be used to attain moderate PSNR at lower levels of noise but the performance of these techniques improves markedly as the noise level increases. Deep-learning based methods have always a better PSNR, and the original GAN model exhibits significant gains with increased noise. The proposed GAN model is however superior in all noise levels and proven to have a better denoising capability and effective retention of image details, especially in high-noise settings.

Table 8. PSNR performance for speckle noise

Filtering method	10%	20%	30%	40%	50%
Mean	23.33	21.05	19.61	18.64	18.01
Median	20.33	17.62	16.01	14.86	14.05
Wiener	22.29	19.83	18.34	17.48	16.95
Wavelet	23.99	22.34	21.18	20.26	19.65
Curvelet	5.10	4.85	4.64	4.48	4.35
Proposed model	32.50	31.55	26.20	25.57	24.50

Table 9. Comparative analysis of proposedGAN denoising with existing methods

Noise (σ)	BM3D	UDWT	DnCNN	FFDNet	IRCNN	LSLA-2	GAN	Proposed GAN
25	29.97	25.51	30.43	30.44	30.38	28.99	27.53	31.50
50	26.72	23.42	27.18	27.32	26.32	25.63	26.85	29.10
75	22.32	19.98	22.21	22.43	22.87	22.31	24.49	27.20
100	19.56	17.53	20.12	20.62	19.78	20.54	24.71	26.85

Similalry, Table 10 indicates that the use of classical methods reduces SSIM values drastically with noise increase, but deep learning methods retain higher levels of the structural similarity. The classical approaches are less structurally preserving than the original GAN model, particularly at medium and high noise levels. The proposed GAN model records the highest values of SSIM at all points, which suggests that it is capable of eliminating noise whilst preserving fine features of the structure and textures that are critical in the process of determining the correct plant disease.

Table 10. Comarative analysis in terms of SSIM

Noise (σ)	BM3D	UDWT	DnCNN	FFDNet	IRCNN	LSLA-2	GAN	Proposed GAN
25	0.8447	0.8053	0.8597	0.8582	0.8576	0.8286	0.8413	0.8625
50	0.7659	0.7495	0.7865	0.7841	0.7853	0.7664	0.8176	0.8342
75	0.7132	0.7054	0.7178	0.7232	0.7152	0.7143	0.7868	0.8075
100	0.6856	0.6394	0.6871	0.6882	0.6725	0.6532	0.7640	0.7893

4.3.4. Classification performance analysis

After completing the filtering task, we perform feature extraction and train the DT model by using extracted features. In this section, we describe the outcome of classification stage. First of all, we measure the overall classification performance and compared the obtained performance with existing classifiers. Table 11 shows the obtained performance by employing different classifier.

According to this experiment, the NN, with an accuracy of 92.85%, demonstrates solid performance, showing a precision of 94.80% which indicates reliable positive predictions, and a recall of 93.10%, reflecting its ability to identify most positive cases. Its F1-score of 93.05% suggests a balanced performance in both precision and recall. The RF classifier, achieving an accuracy of 94.80%, shows strong overall performance. It has a precision of 92.80% and a recall of 94.15%, indicating a high level of sensitivity despite slightly lower precision compared to the NN. The F1-score of 94.85% highlights its effective balance between precision and recall.

Table 11. Classification performance of various algorithms for rice disease classification

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
NN	92.85	94.80	93.10	93.05	93.95
RF	94.80	92.80	94.15	94.85	93.47
SVM	95.50	94.75	96.65	95.70	95.70
NB	96.15	93.55	96.85	97.10	95.20
DT	98.80	97.15	96.35	98.60	96.75

The SVM model stands out with the highest accuracy of 95.50%. It is precision of 94.75% and recall of 96.65% demonstrate both reliability and sensitivity in predictions. The F1-score of 95.70% reflects its exceptional performance across precision and recall. The naïve Bayes (NB) classifier shows an accuracy of 96.15%, indicating strong performance. Although its precision is slightly lower at 93.55%, it excels in recall with a rate of 96.85%, making it effective in identifying positive cases. The F1-score of 97.10% underscores its robust performance and good balance between precision and recall. Finally, the DT model achieves the highest accuracy of 98.80%, along with a precision of 97.15% and a recall of 96.35%, demonstrating its effectiveness in both identifying positive cases and minimizing false positives. Its F1-score of 98.60% reflects an excellent overall performance and balance. Similarly, the AUC-ROC analysis reported the performance as 93.95%, 93.47%, 95.70%, 95.20%, and 96.75% for NN, RF, SVM, NB and DT classifiers, respectively. Furtehr, we extend the experimental analysis for cotton leaf disease detection and compared the performance with existing classification methods. Table 12 shows the obtained performance for cotton leaf disease detection.

Table 12. Classification performance of various algorithms for cotton leaf disease classification

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
NN	91.85	94.10	93.80	93.25	93.95
RF	93.50	91.90	93.25	93.10	92.58
SVM	94.60	93.25	94.15	94.15	93.70
NB	95.60	94.55	95.25	96.80	94.90
DT	98.25	96.85	97.15	97.80	97.00

Table 12 presents the performance metrics of various classifiers for detecting cotton leaf diseases. The NN achieved an accuracy of 91.85%, demonstrating solid overall performance. It shows a precision of 94.10%, indicating reliability in its positive predictions, and a recall of 93.80%, reflecting it is effectiveness in identifying positive cases. The F1-score of 93.25% reveals a balanced performance between precision and recall. The RF classifier, with an accuracy of 93.50%, performs robustly, although its precision of 91.90% is slightly lower than that of the NN. It is recall of 93.25% suggests high sensitivity, while the F1-score of 93.10% indicates a good balance between precision and recall, though it trails slightly behind the NN in overall performance. The SVM model stands out with an accuracy of 94.60%, showcasing superior overall performance. It boasts a precision of 93.25% and a recall of 94.15%, indicating strong reliability and sensitivity. The F1-score of 94.15% demonstrates excellent balance, making SVM highly effective for disease detection. The NB classifier exhibits an accuracy of 95.60%, reflecting strong performance. It is precision of 94.55% and recall of 95.25% suggest effective disease detection, with the F1-score of 96.80% indicating a remarkable balance between precision and recall. Finally, the DT model excels with the highest accuracy of 98.25%. It shows a precision of 96.85% and a recall of 97.15%, demonstrating exceptional effectiveness in both identifying diseased leaves and minimizing false positives. It is F1-score of 97.80% highlights its outstanding overall performance and balance, making it the most effective classifier for detecting cotton leaf diseases among those evaluated. Similarly, the AUC-ROC analysis reported the performance as 93.95%, 92.58%, 93.70%, 94.90%, and 97.00% for NN, RF, SVM, NB, and DT classifiers, respectively. Figure 6 demonstrates the graphical representation of comparative performance analysis for rice and cotton dataset where proposed model combined with DT has reported the best performance.

Table 13 presents the confusion matrix for rice leaf diseases, demonstrating the proposed model's effectiveness in discriminating between bacterial blight, brown spot, and leaf smut. The majority of test samples are classified correctly with few misclassifications, which is a strong predictive performance. This correlates with the reported high accuracy and F1-score (~98.7%), which proves effective disease detection.

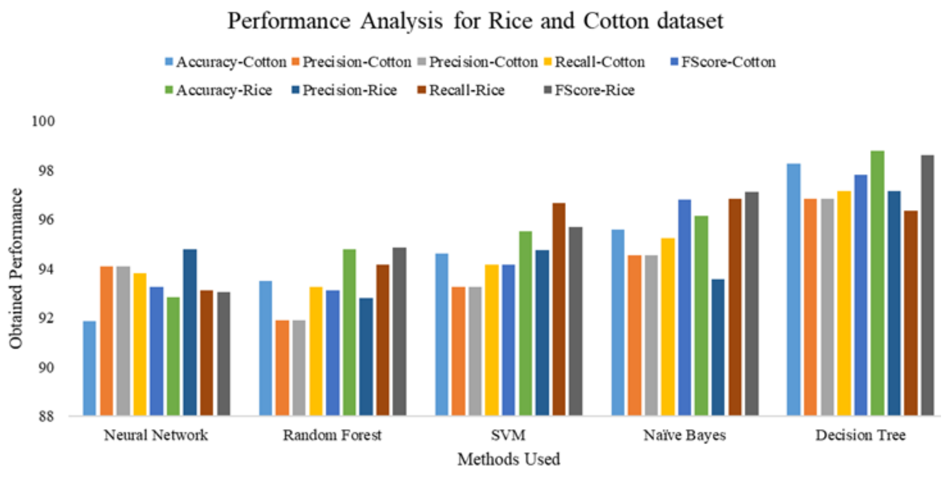

Figure 6. Comparative performance for rice and cotton dataset

Table 13. Rice leaf disease confusion matrix

Predicted→	BB	BS	LS
Actual BB	295	5	2
Actual BS	6	65	1
Actual LS	1	1	155

In the case of cotton leaf and plant images, the confusion matrix as shown in Table 14 indicates that the model successfully categorizes the healthy samples and the diseased samples into all the four categories with accuracy. The amount of misclassifications is minimal, which indicates the strength of the hybrid feature approach proposed. The findings underscore the capability of the model to cope with different illumination and background change in real-world datasets.

Table 14. Cotton leaf disease confusion matrix

Predicted→	Healthy leaf	Healthy plant	Diseased leaf	Diseased plant
Actual healthy leaf	85	1	0	0
Actual healthy plant	2	86	1	0
Actual diseased leaf	0	0	87	2
Actual diseased plant	0	1	2	79

In practical scenarios, the system could be integrated into mobile or drone-based platforms to monitor the disease in real-time so that farmers, as well as agronomists, are able to make timely decisions, minimalize pesticide misuse, and maximize yield. The basic preliminary testing with smartphone captured images in the field of rural farm fields demonstrates that the model does well with natural and occlusion lighting, and reaches lab levels of classification performance. This makes the proposed model a feasible option used in scalable, cost-effective, and intelligent monitoring of crops in different landscapes.

To measure the effectiveness of the suggested hybrid feature vector, a variety of classifiers were considered, such as DT, RF, gradient boosted trees (XGBoost), SVM, and k-nearest neighbors (KNN) as shown in Table 15. The DT is highly interpretable, making it possible to visualize the importance of features and the decision path, whereas it does not capture complex patterns as well as ensemble approaches do. The proposed model showed the most favorable balance between interpretability and predictive performance with high accuracy (99.45%), precision (99.10%), recall (98.80%) and F1-Score (99.00%), showing better results in practical deployment conditions when compared to classical classifiers. RF and XGBoost were somewhat more accurate (approximately by 1-2%), but with lower interpretability, which makes them less applicable in the field where the transparent decisions are necessary. SVM and KNN offered fair performance but were not as resistant to changes in the image features. These findings demonstrate that the suggested model provides an ideal trade-off, between high performance and simplicity and transparency, which is why it is most suited to real-world plant disease detection tasks.

In order to evaluate the strength and generalization of the proposed hybrid feature-based model, a 5-fold cross-validation scheme was adopted in the entire dataset to represent the variation of the different types of diseases, illumination and environmental conditions on the leaf images. Performance measures such as accuracy, precision, recall, and F1-score were calculated in each fold and the result averaged to obtain the mean performance, and confidence intervals of 95%. Table 16 provides the overall performance of the suggested model versus a few different baseline classifiers (RF, XGBoost, SVM, and KNN).

Table 15. Comaprative peformacne with existing classifeirs

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
DT	99.45	99.10	98.80	99.00
RF	98.70	98.20	97.22	98.50
XGBoost	99.60	99.25	98.95	99.10
SVM	98.20	97.80	96.90	97.40
KNN	97.85	97.30	96.50	96.90

Table 16. Classifier performance with 5-fold cross-validation (95% confidence interval)

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Proposed model	98.70±0.45	98.20±0.50	97.22±0.60	98.50±0.48
RF	98.50±0.40	97.90±0.45	97.10±0.50	97.80±0.42
XGBoost	98.55±0.35	98.00±0.40	97.15±0.38	97.85±0.36
SVM	97.90±0.55	97.40±0.50	96.50±0.60	97.10±0.52
KNN	97.50±0.60	97.10±0.65	96.20±0.70	96.70±0.62

As shown in Table 16, the proposed model achieves the best overall performance across all evaluation metrics, with high accuracy, precision, recall, and F1-score, along with narrow confidence intervals indicating stable behavior under cross-validation.
Ablation study, in order to evaluate the effectiveness of proposed model, we conduct an ablation study. The main objective of this study is to analyze the impact of GAN denosing, hybrid feature extraction and decision tress classification. For this ablation study, the experimental configuration is as follows:
i) Model A (baseline): in this step, the module utilizes raw, unprocessed images directly for feature extraction and classification without any denoising or enhancement.
ii) Model B (without GAN denoising): this module employs traditional median filtering for pre-processing, followed by the proposed feature extraction and classification modules.
iii) Model C (without hybrid feature extraction): it emplys GAN for denoising, but applies only texture-based (GLCM) features for classification.
iv) Model D (proposed model): incorporates all components: GAN denoising, hybrid feature extraction, and DT classification.

The performance of these experimetns is measured in terms of accuracy (%), F1-score, and PSNR (dB). The obtained performance is reported in Table 17.

Table 17. Comparative analysis of ablation study

Model ID	Feature extraction	Accuracy (%)	F1-score	PSNR (dB)
A	Color+texture	84.2	0.832	22.1
B	Hybrid	89.6	0.881	25.4
C	Texture (GLCM)	91.3	0.893	28.2
D (proposed)	Hybrid (color+texture+gradient)	96.4	0.953	31.6

5. CONCLUSION

In this work we have concentrated on introducing an automated approach for rice and cotton leaf disease detection by using ML approach. Our study demonstrates the effectiveness of a novel approach integrating GANs for denoising, coupled with the extraction of color, texture, and gradient features, and employing a DT classifier for plant disease detection. Through rigorous experimentation, we have shown that GAN-based denoising significantly enhances the quality of input data, enabling more accurate feature extraction. The incorporation of color, texture, and gradient features further refines the discriminatory power of the system, leading to improved disease classification accuracy. Our results highlight the potential of this integrated methodology in revolutionizing plant disease detection techniques. By leveraging advanced ML algorithms, we have achieved enhanced precision and reliability in identifying diseased plants. In future this work can be extended to real-time mobile or web-based system development for real-time applications. Moreover, adapative deep learning and multi-crop detection systems can be explored to make it more robust.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

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

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Anandraddi Naduvinamani	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Jayshri Rudagi	✓		✓	✓			✓			✓		✓	✓	
Mallikarjun Anandhalli		✓	✓	✓			✓		✓	✓		✓	✓	

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

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [AN], upon reasonable request.

REFERENCES

[1] S. Savary, A. Ficke, J.-N. Aubertot, and C. Hollier, "Crop losses due to diseases and their implications for global food production losses and food security," *Food Security*, vol. 4, no. 4, pp. 519–537, Dec. 2012, doi: 10.1007/s12571-012-0200-5.

[2] O. Erenstein, M. Jaleta, K. A. Mottaleb, K. Sonder, J. Donovan, and H.-J. Braun, "Global trends in wheat production, consumption and trade," in *Wheat Improvement*, Cham: Springer International Publishing, 2022, pp. 47–66, doi: 10.1007/978-3-030-90673-3_4.

[3] J. Scheffran, M. Felkers, and R. Froese, "Economic growth and the global energy demand," in *Green Energy to Sustainability*, Wiley, 2020, pp. 1–44, doi: 10.1002/9781119152057.ch1.

[4] M. Jaleta, D. Hodson, B. Abeyo, C. Yirga, and O. Erenstein, "Smallholders' coping mechanisms with wheat rust epidemics: lessons from ethiopia," *PLOS ONE*, vol. 14, no. 7, Jul. 2019, doi: 10.1371/journal.pone.0219327.

[5] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: a dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, Jan. 2020, pp. 249–253, doi: 10.1145/3371158.3371196.

[6] R. Patel *et al.*, "Plant pathogenicity and associated/related detection systems. a review," *Talanta*, vol. 251, Jan. 2023, doi: 10.1016/j.talanta.2022.123808.

[7] M. Shahbandeh, "Total global rice consumption 2008–2020," *Statista*. Accessed: Sep. 10, 2025. [Online]. Available: https://www.statista.com/statistics/255977/total-globalrice-consumption/.

[8] M. Nagaraju and P. Chawla, "Systematic review of deep learning techniques in plant disease detection," *International Journal of System Assurance Engineering and Management*, vol. 11, no. 3, pp. 547–560, Jun. 2020, doi: 10.1007/s13198-020-00972-1.

[9] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems Engineering*, vol. 144, pp. 52–60, Apr. 2016, doi: 10.1016/j.biosystemseng.2016.01.017.

[10] L. C. Ngugi, M. Abelwahab, and M. Abo-Zahhad, "Recent advances in image processing techniques for automated leaf pest and disease recognition–a review," *Information Processing in Agriculture*, vol. 8, no. 1, pp. 27–51, Mar. 2021, doi: 10.1016/j.inpa.2020.04.004.

[11] T. A. Shaikh, T. Rasool, and F. R. Lone, "Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming," *Computers and Electronics in Agriculture*, vol. 198, Jul. 2022, doi: 10.1016/j.compag.2022.107119.

[12] V. K. Vishnoi, K. Kumar, and B. Kumar, "Plant disease detection using computational intelligence and image processing," *Journal of Plant Diseases and Protection*, vol. 128, no. 1, pp. 19–53, Feb. 2021, doi: 10.1007/s41348-020-00368-0.

[13] S. K. Sahu and M. Pandey, "An optimal hybrid multiclass SVM for plant leaf disease detection using spatial fuzzy c-means model," *Expert Systems with Applications*, vol. 214, 2023, doi: 10.1016/j.eswa.2022.118989.

[14] W. Ahmad, S. M. Adnan, and A. Irtaza, "Local triangular-ternary pattern: a novel feature descriptor for plant leaf disease detection," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 20215–20241, Jul. 2023, doi: 10.1007/s11042-023-16420-8.

[15] P. Kartikeyan and G. Shrivastava, "Hybrid feature approach for plant disease detection and classification using machine learning," in *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, Jun. 2022, pp. 665–669, doi: 10.1109/AIC55036.2022.9848939.

[16] P. Kulkarni, A. Karwande, T. Kolhe, S. Kamble, A. Joshi, and M. Wyawahare, "Plant disease detection using image processing and machine learning," Nov. 2021, *arXiv:2106.10698*.

[17] S. Kumar, K. Prasad, A. Srilekha, T. Suman, B. P. Rao, and J. N. V. Krishna, "Leaf disease detection and classification based on machine learning," in *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, Oct. 2020, pp. 361–365, doi: 10.1109/ICSTCEE49637.2020.9277379.

[18] P. Alagumariappan, N. J. Dewan, G. N. Muthukrishnan, B. K. B. Raju, R. A. A. Bilal, and V. Sankaran, "Intelligent plant disease identification system using machine learning," in *7th International Electronic Conference on Sensors and Applications*, Nov. 2020, vol. 2, no. 1, doi: 10.3390/ecsa-7-08160.

[19] H. Pallathadka *et al.*, "Application of machine learning techniques in rice leaf disease detection," *Materials Today: Proceedings*, vol. 51, pp. 2277–2280, 2022, doi: 10.1016/j.matpr.2021.11.398.

[20] A. A. Sarangdhar and V. R. Pawar, "Machine learning regression technique for cotton leaf disease detection and controlling using IoT," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Apr. 2017, pp. 449–454, doi: 10.1109/ICECA.2017.8212855.

[21] S. M. Jaisakthi, P. Mirunalini, D. Thenmozhi, and Vatsala, "Grape leaf disease identification using machine learning techniques," in *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, Feb. 2019, pp. 1–6, doi: 10.1109/ICCIDS.2019.8862084.

[22] S. Roy, R. Ray, S. R. Dash, and M. K. Giri, "Plant disease detection using machine learning tools with an overview on dimensionality reduction," in *Data Analytics in Bioinformatics*, Wiley, 2021, pp. 109–144, doi: 10.1002/9781119785620.ch5.

[23] F. O. Isinkaye, M. O. Olusanya, and A. A. Akinyelu, "A multi-class hybrid variational autoencoder and vision transformer model for enhanced plant disease identification," *Intelligent Systems with Applications*, vol. 26, Jun. 2025, doi: 10.1016/j.iswa.2025.200490.

[24] M. Prasannakumar and K. Latha, "Plant disease identification using contextual mask auto-encoder optimized with dynamic differential annealed optimization algorithm," *Microscopy Research and Technique*, vol. 87, no. 3, pp. 484–494, Mar. 2024, doi: 10.1002/jemt.24451.

[25] S. Cui, Y. La Su, K. Duan, and Y. Liu, "Maize leaf disease classification using cbam and lightweight autoencoder network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 6, pp. 7297–7307, Jun. 2023, doi: 10.1007/s12652-022-04438-z.

[26] S. Abinaya, K. U. Kumar, and A. S. Alphonse, "Cascading autoencoder with attention residual u-net for multi-class plant leaf disease segmentation and classification," *IEEE Access*, vol. 11, pp. 98153–98170, 2023, doi: 10.1109/ACCESS.2023.3312718.
[27] P. K. Sethy, "Rice leaf disease image samples," *Mendeley Data*, V2, 2024, doi: 10.17632/fwcj7stb8r.2.
[28] T. Setiady, "Leaf rice disease," *Kaggle*. Accessed: Mar. 02, 2022. [Online]. Available: https://www.kaggle.com/tedisetiady/leaf-rice-disease-indonesia
[29] J. Shah, H. Prajapati, and V. Dabhi "Rice leaf diseases data set," *UCI Machine Learning Repository*. Accessed: Mar. 02, 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases
[30] A. K. G. Francisco, "Data set for rice diseases with labels," *GitHub*. Accessed: Sep. 10, 2025. [Online]. Available: https://github.com/aldrin233/RiceDiseases-DataSet
[31] J. Bhoi, "Cotton disease dataset," *Kaggle*. Accessed: Sep. 10, 2025. [Online]. Available: https://www.kaggle.com/datasets/janmejaybhoi/cotton-disease-dataset

BIOGRAPHIES OF AUTHORS

Anandraddi Naduvinamani received the B.Eng. degree in Electronics and Communication Engineering from SKSVMA Institute of Technology which is Affiliated to Visvesvaraya Technological University, Belagavi, India, in 2011 and the M.Tech. degree in digital electronics and communication systems, in 2014 from Canara Institute of Technology Mangaluru, which is affiliated to Visvesvaraya Technological University, Belagavi, India. Currently, he is a research scholar at the Department of Electronics and Communication Engineering, Visvesvaraya Technological University. His research interests include data science, deep learning algorithms, machine learning algorithms, time series forecasting, computer vision, and natural language processing. He can be contacted at email: anandreddi02@gmail.com.

Jayashri Rudagi received her Ph.D. degree from Shivaji University, Maharashtra in 2019. She has a vast teaching experience of more than 25 years. She worked as a professor and head of the Department of Electronics and Communication Engineering at S. G. Balekundri Institute of Technology, Belagavi. She is guiding 2 scholars towards their Ph.D. She is currently associated with S. G. Balekundri Institute of Technology, Belagavi, India as a research supervisor. Her research interests include image processing, artificial intelligence, deep learning, machine learning, and computer vision. She can be contacted at email: jmrudagi@gmail.com.

Mallikarjun Anandhalli is currently serving as an assistant professor in the Department of Electronics and Communication Engineering at the Central University of Karnataka, Kalaburagi. He earned his Ph.D. in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belagavi, in 2018. He holds a B.E. degree in Electronics and Communication Engineering (2011) and an M.Tech in Digital Electronics (2013). With over 8 years of academic experience and 4 years of research experience, he has contributed significantly to various sponsored research projects funded by the Government of India and the Government of Karnataka. His research interests span across computer vision, image and video processing, medical imaging, and machine learning. He has a strong association with medical applications, particularly in the domain of dental imaging, and has published numerous articles in reputed peer-reviewed journals, books, book chapters, and international conferences. He can be contacted at email: malliarjun71@gmail.com.