Multiclass Classification of Paddy Leaf Diseases Using Random Forest Classifier

Saminathan K[1], Sowmiya B[1,*], and Chithra Devi M[2]

[1]A.V.V.M. Sri Pushpam College, PG and Research Department of Computer Science, Affiliated to Bharathidasan University, Poondi, Thanjavur, Tamil Nadu, India;
Email: arksami@avvmspc.ac.in (S.K.), m.chithradevi@gmail.com (C.D.M)
[2]Queens College of Arts and Science for Women, Affiliated to Bharathidasan University, Pudukkottai, India
*Correspondence: sowmiyabaskar@gmail.com (S.B.)

***Abstract*—With increase in population, improving the quality and quantity of food is essential. Paddy is a vital food crop serving numerous people in various continents of the world. The yield of paddy is affected by numerous factors. Early diagnosis of disease is needed to prevent the plants from successive stage of disease. Manual diagnosis by naked eye is the traditional method widely adopted by farmers to identify leaf diseases. However, when the task involves manual disease diagnosis, problems like the hiring of domain experts, time consumption, and inaccurate results will arise. Inconsistent results may lead to improper treatment of plants. To overcome this problem, automatic disease diagnosis is proposed by researchers. This will help the farmers to accurately diagnose the disease swiftly without the need for expert. This manuscript develops model to classify four types of paddy leaf diseases bacterial blight, blast, tungro and brown spot. To begin with, the image is preprocessed by resizing and conversion to RGB Red, Green and Blue (RGB) and Hue, Saturation and Value (HSV) color space. Segmentation is done. Global features namely: hu moments, Haralick and color histogram are extracted and concatenated. Data is split in to training part and testing part in 70:30 ratios. Images are trained using multiple classifiers like Logistic Regression, Random Forest Classifier, Decision Tree Classifier, K-Nearest Neighbor (KNN) Classifier, Linear Discriminant Analysis (LDA),Support Vector Machine (SVM) and Gaussian Naive Bayes. This study reports Random Forest classifier as the best classifier. The Accuracy of the proposed model gained 92.84% after validation and 97.62% after testing using paddy disordered samples. 10 fold cross validation is performed. Performance of classification algorithms is measured using confusion matrix with precision, recall, F1-score and support as parameters.**

***Keywords*—paddy leaf diseases, preprocessing, segmentation, feature extraction, classification, machine learning, random forest**

I. Introduction

In India, the main source of employment is in the agricultural industry and its associated sectors. India is the world’s second-largest rice producer [1]. India needs to meet the rising food needs of the country. Rice produce must be increased by the development of cultivars, coordinated harvesting, and advancements in irrigation management [2]. Producing enough food to fulfill societal need is now possible with the support of developed technologies. Yet, we need to improve food crop security and protection to the fullest [3]. Plants are vulnerable to diseases. The diseases are caused by bacteria, fungi and virus. The diseases reduce the quality and quantity of yield. Manual disease identification takes a lot of time, needs specialized knowledge, and is impractical in big farms. It is quite challenging too [4]. In order to speed up crop diagnosis, plant leaf disease detection systems should be automated [5]. As vision-based technology can swiftly identify illnesses, machine learning and deep learning approaches are being adopted for disease automation. Although deep learning produces promising results, it demands a big dataset and takes more time when compared to machine learning [6]. Deep learning requires high performance computer, large amounts of computation and memory which may not always feasible. In such cases, simpler models that can be trained on less powerful hardware may be a more practical choice [7]. Though various parts of plants show disease symptoms, leaves play a dominant role in identifying diseases [8]. Image processing approach follows certain steps such as preprocessing, segmentation, feature extraction and classification [9].

Crop productivity will increase the quality of yield with early disease detection mechanism. The aim of the paper is to propose an automated framework which recognizes diseases with the least amount of expenditure and without the expertise knowledge. The rest of the paper is organized as follows. Section II covers current state of research in the same field as well as brief review of disease detection mechanism. Section III depicts the proposed system with detailed explanation of each step-in disease classification process. Section IV comprises results and discussion. Section V concludes the paper.

II. Literature Review

Aggarwal *et al.* [2] reviews about state-of-the-art algorithms in various stages of disease detection in rice

Manuscript received December 22, 2022; revised January 28, 2023; accepted April 13, 2023.

doi: 10.18178/joig.11.2.195-203

with advanced artificial intelligence and machine learning. Additionally, they provide current year publications and citations broken down by year and nation to aid academics working in the same field.

Harakannanavar *et al.* [3] employ image resizing, Histogram Equalization (HE) technique for image enhancement. K-means clustering applied for segmentation. The contour tracing technique extracts boundary of leaf affected areas. Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA) and Gray Level Co-occurrence Matrix (GLCM) methods extract informative features from leaf. Classification done by SVM, KNN and Convolutional Neural Network (CNN) with 88%, 97% and 99.6% accuracy on tomato disordered samples.

Kartikeyan *et al.* [4] reviews various machine learning algorithms like SVM, Artificial Neural Network (ANN), KNN, Fuzzy KNN and conclude SVM as the best recommended algorithm for classification. Its futuristic work includes hybrid algorithms development with the help of Particle Swarm Optimization (PSO) with SVM, ANN and KNN, genetic algorithms, Ant colony and Cuckoo optimization with mobile application development.

Zamani *et al.* [5] have a vision to automate disease diagnosis by machine learning and image processing techniques. Median filter and K-means algorithm does noise removal and segmentation. PCA is used for feature extraction. Classification uses Radial Basis Function-SVM (RBF-SVM), SVM, Random Forest (RF) and ID3. RBF-SVM outperforms in classification.

Dhar *et al.* [6] uses global and local features as its novelty. Rice, apple, cherry, corn and other leaf varieties are used. Images are resized, filtered using median filter. Feature extraction done using Gist and Local Binary Pattern (LBP). Gist assesses well for smaller feature-sized image. The LBP feature categorizes images with various lighting and environmental changes like occlusion and illusion. Extended LBP is utilized here. SVM, KNN and AdaBoost classifiers are used and SVM records with highest accuracy of 99.7% for cherry leaf disease. SVM is found to be better than KNN and AdaBoost except for rice and tomato. KNN is good for rice leaf disease classification and AdaBoost works fine for Tomato leaf disease classification. Confusion matrix and Receiver Operating Characteristics (ROC) curve analyses classifier's result. The True Positive Rate (TPR) and False Positive Rate (FPR) are plotted on the ROC curve.

Kaur *et al.* [8] proposed disease recognition using ensemble classification. K-means clustering is applied; Followed by Law's textural mask, Gray Level Co-occurrence Matrix (GLCM), LBP, Gabor and Scale-Invariant Feature Transform (SIFT) for texture extraction. Ensemble classification of ANN, SVM, KNN and Logistic regression and Naive Bayes classifiers is done. Accuracy of 95.66% is achieved with proposed features Law's mask, Gabor and ensemble approaches for potato leaf with three classes.

Azim *et al.* [9] utilized 120 images, RGB to HSV color space conversion and binary thresholding is done for background removal. Image masking is done to get background removed image. Hue based segmentation is done for finding infected spots. Totally 26 features, of which 5 shape, 12 color and 5 texture features are extracted. GLCM and LBP as texture feature descriptors. Classification is done by XGBoost, SVM (RBF kernel) with 86.58% and 81.67% accuracy. XGBoost classifier used learning rate, maximum depth, and minimal child weight as parameters. Yasiran *et al.* [10] deal with breast cancer classification with 80 image data from standard MIAS database. Image is resized, median filter and histogram stretching is applied. 22 features are extracted including Haralick textures and Hu invariant moments features with four features: Number of spots, area, perimeter and compactness. Classification using SVM (RBF kernel) produces 90.5% accuracy for two classes (Benign and malignant) and 77.5% accuracy for multi class categories (Fatty, glandular, dense). Quadratic programming, least squares, and sequential minimal optimization were evaluated and contrasted for multiclass SVM. SVM minimizes generalization error. The ROC's AUC (Area under the Curve) and 10 fold cross-validation is done to assess performance.

Ansari *et al.* [11] used totally 400 images with 250 diseased and 150 healthy images. Denoising done using Adaptive Mean Filtering and Contrast Limited AHE (CLAHE) performs image enhancement. Fuzzy C means algorithm performs segmentation. PCA does feature extraction. PSOSVM, BPNN, and RF algorithms perform classification. Highest accuracy achieved using PSO SVM. Mahapatra *et al.* [12] uses 40,000 images from Kaggle dataset, considering 9 plant leaves with total of 33 classes. Grayscale image conversion, Gaussian filter, OTSU Thresholding, morphological operation for closing gaps and edge detection using Sobel and contours are done. Classification done using SVM and achieved 91% accuracy.

Joshi *et al.* [13] discusses benchmark datasets, feature identification and description techniques, and performance assessment. Singh *et al.* [14] make use of hybrid features of CNN and Bayesian Optimized SVM and Random Forest. 37315 images utilized from Plant Village Dataset. Data augmentation techniques are employed in first phase. Deep features extracted using CNN. This work proposes an algorithm that customizes SVM parameters. Image resized to 300 × 450 pixels, RGB to Grey for texture and RGB to LAB conversion are done for color feature extraction in second phase. Color moments: mean and standard deviation are calculated. GLCM extracts texture characteristics and energy, contrast, homogeneity, and correlation are calculated. HOG extracts color feature. Color moments, GLCM features and HOG are combined. A binary PSO is used to choose these hybrid characteristics and categorized using RF classifier and both the results are compared.

Thakur *et al.* [15] reviews classification and localization methods in diseases along with data collection methods and data preprocessing techniques. Image processing, machine learning and deep learning methods are focused. Public datasets are reviewed for different plant cultures. Research articles are selected from 2005 to 2022 and highly scrutinized. Ibrahim *et al.* [16] proposed a system to detect rice leaf disease using deep learning and machine learning with 1000 images including brown spot, bacterial blight and leaf smut diseases in 90:10 ratios for training and testing. After cropping, noise and shadow removal performed using bilateral filter and canny edge detection algorithm. Images are converted into features using color layout filter method. For attribute selection among 35 attributes, correlation-based attributes selection technique is applied. RF algorithm outperforms with more than 95% accuracy. 10 fold cross validation test is done for performance assessment.

III. Materials and Methods

This study intends to develop an accurate and effective plant disease detection method for paddy plants using proposed machine learning and image processing techniques. As a result, we provide a thorough explanation of the proposed machine learning model in this part. Fig. 1 depicts proposed system workflow.

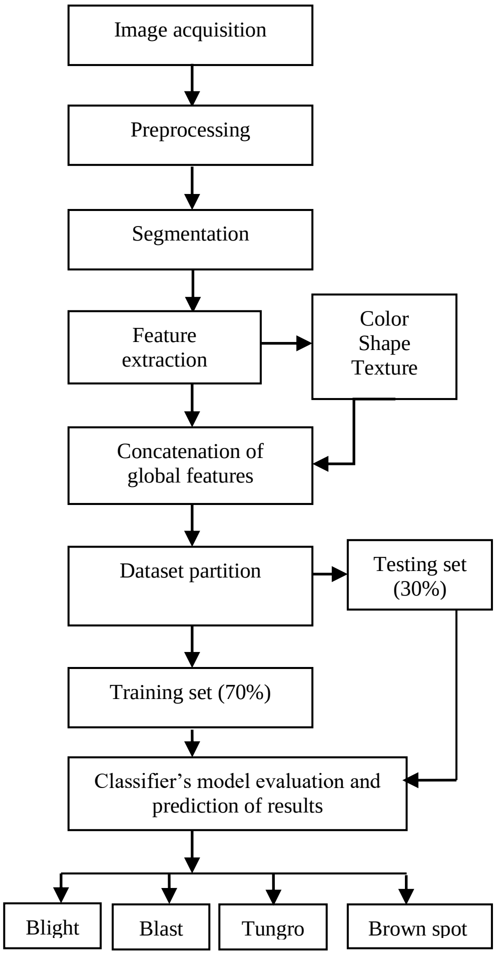
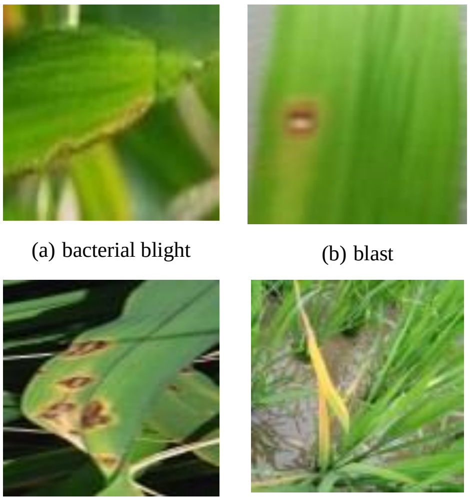


Figure 1. Proposed methodology.

A. Image Acquisition

Image acquisition is the first and foremost step. Images are acquired from Sethy [17]. This dataset has a total of 5932 images in jpg format. Fig. 2 displays the sample of images from four classes of paddy diseases. Fig. 2(a) depicts bacterial blight, Fig. 2(b) depicts blast, Fig. 2(c) brown spot and Fig. 2(d) depicts tungro. Our research includes 140 images from each class and a total of 560 images.



(a) bacterial blight

(b) blast

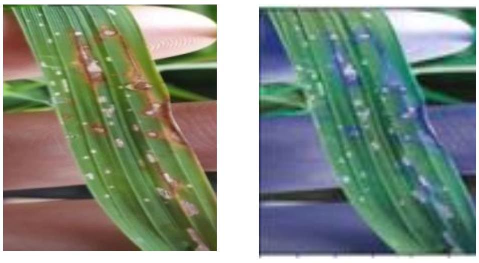
(c) brown spot

(d) tungro

Figure 2. Sample image of four classes of paddy leaf diseases.

B. Preprocessing

Preprocessing improves the quality of image. Every image is resized to 300 × 300 pixels on training phase to reduce memory consumption and to improve performance. Basically, image is read in BGR format using python's OpenCV library. So actual image is converted is carried out to BGR format. Fig. 3(a) shows Actual image from dataset and Fig. 3(b) shows BGR converted preprocessed image.



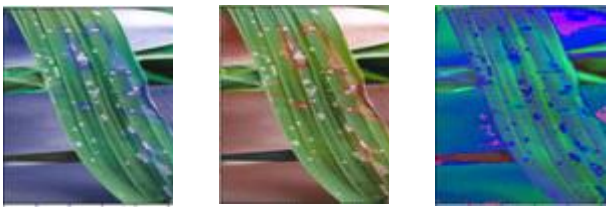
(a) Actual image

(b) BGR converted image

Figure 3. Preprocessed images.

C. Segmentation

Segmentation is done to identify healthy and diseased regions of leaf image. Green pixels are considered to be healthy and brown pixels are considered to be unhealthy. For extracting green color, image should be converted from BGR image to RGB format. For extracting diseased portion, we take brown ranges. So we convert image from RGB to HSV format. From HSV format, we are extracting unhealthy leaf portion. Masking is applied on leaf image to find the segmented portion of healthy and diseased regions. Fig. 4(a) depicts BGR converted image, Fig. 4(b) depicts RGB converted image and Fig. 4(c) depicts HSV converted image.



(a) BGR converted image

(b) RGB converted image

(c) HSV converted image

Figure 4. Various Color space converted images.

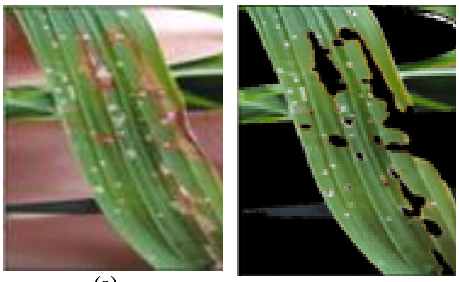
The algorithm for segmentation is explained below.

Algorithm1: Image segmentation

```
Input: RGB image and HSV image
Output: Segmented image

Step 1: start
Step 2: Read lower and upper green values from RGB
        image.
Step 3: Create a healthy mask with result obtained from
        step1.
Step 4: Read lower and upper brown values from HSV
        image.
Step 5: Create a disease mask with values obtained in
        step3.
Step 6: Finally a single mask is created using
        bitwise_and() using values obtained in Step 2 and
        Step 3.
Step 7: Return the segmented image
Step 8: End
```

Fig. 5(a) and Fig. 5(b) show the RGB image and segmented image, which clearly depicts the healthy and blast infected regions obtained as per the algorithm.



(a)

(b)

Figure 5. (a) RGB image (b) segmented region of blast affected leaf.

D. Feature Extraction

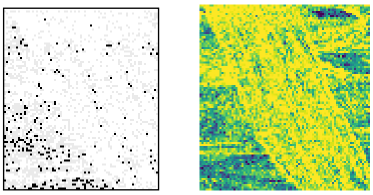
Feature detectors are used to find the essential features from the given image. Feature descriptors are algorithms which extract the features. In this research, color, texture and shape features are extracted. Three feature descriptors: hu moments, Haralick texture and color histogram extract shape, texture and color features respectively.

The shape of an object in an image is evaluated using **Hu Moments** image descriptor. It is implemented using cv2.HuMoments function in python's cv2 package. Color conversions performed to convert BGR image to gray scale using the flag, cv2.COLOR_BGR2GRAY. Table I shows seven Hu moments extracted for shape feature.

TABLE I. HU MOMENTS — SHAPE EXTRACTION

Hu moment values
1.396161×10^{-3}
6.54941×10^{-9}
1.04335×10^{-11}
8.62052×10^{-12}
4.44126×10^{-23}
3.22539×10^{-16}
-6.86398×10^{-23}

The texture of an object is evaluated using Haralick texture and it is computed using GLCM. Total of 13 Haralick features are extracted. These features are extracted in four directions. Altogether, 52 features are extracted. Mean of each direction is taken into consideration. Fig. 6(a) and Fig. 6(b) show entropy feature extraction for blast disease in gray scale and RGB scale. It shows the amount of information in the image. If the entropy value is higher, then the image will be more detailed [18].



(a)

(b)

Figure 6. Entropy feature extraction for blast diseases in (a) gray scale and (b) RGB scale.

The color feature of an image is extracted using **Histogram**. It is computed using calcHist function in cv2 package and normalization is performed to improve overall contrast of the image. Fig. 7 shows normalized histogram distribution of segmented leaf image.

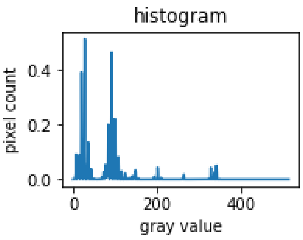


Figure 7. Normalized histogram of segmented leaf.

After extracting the global features: hu moments, Haralick and histogram, they are concatenated for every image. Feature vector is formed with these extracted features. The size of combined feature vectors size obtained is (560,532). Training labels are obtained, sorted and encoded for conversion to integers. Feature vector is imported and labels are trained. Normalization is done to scale values in the range of 0 to 1. Entire Feature vector is normalized by MinMaxScaler and saved in a file. The formula for min-max normalization is given in Eq. (1).

$$x_{\text{normalized}} = \frac{(x_{\min})}{(x_{\max} - x_{\min})} \qquad (1)$$

where, *x* is the original value.

The coding for minmax scaler is given below:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler (feature_range= (0, 1))
rescaled_features=scaler.fit_transform(global_features)
```

E. Classification

Classifier is used to classify the images into relevant classes. Our research uses classifiers like Logistic Regression, Decision Tree Classifier, Random Forest Classifier, KNN Classifier, Linear Discriminant Analysis, Gaussian NB and SVC (SVM). The rest of this section briefs these algorithms in no particular order.

1) SVM classifier

SVM is a supervised machine learning algorithm and is best suited for classification problems. Since it improves the model's performance on test or unseen data, the margin maximization method employed by SVM seeks to choose the hyperplane that splits the data points as much as possible. SVM utilizes support vectors not the entire training data set. Our research uses linear kernel since there are more features involved as well as classification problems are linear separable and is mathematically expressed in Eq. (2).

$$f(X) = w^T \times X + b \qquad (2)$$

2) Logistic regression classifier

Logistic Regression is a machine learning algorithm for classification problems and uses probability concepts. It uses the cost function, 'sigmoid function' which ranges between 0 and 1. Cost function minimizes the error and results in an accurate model and is mathematically expressed in Eqs. (3a), (3b) and (3c).

$$h_\theta\, x(i) = 1 / (1 + \exp(-z)) \qquad (3a)$$

$$\text{Cost}(i) = -[\, y(i) \times \log(h_\theta(x(\text{i})) + (1-y(i)) \times \log(1\text{-}h_\theta(x(i)))] \qquad (3b)$$

$$J(\theta) = -\frac{1}{m} \times \sum_{i=0}^{m} (\text{Cost}(i)\,) \qquad (3c)$$

3) Decision tree classifier

Decision Tree is a supervised classification algorithm for classification problems and regression problems as well. It is basically a tree-based structure. To train a decision tree, we need metrics such as Entropy and Information gain. Entropy determines how a decision tree makes choice of splitting data. Randomness in data is what Entropy and information gain makes use of entropy to make decisions. The mathematical representations for Entropy and Information gain are given in Eq. (4) and Eq. (5a), (5b).

$$E(S) = \sum_{i=0}^{c} \text{-}p_i \log_2 p_i \qquad (4)$$

where $p_i\log_2 p_i = p \times \log_2(p)$, p is the probability of outcome and $\log_2$ is logarithm function of base 2.

$$IG(Y, X) = E(Y) - E(Y|X) \qquad (5a)$$

$$E(Y|X) = \sum P(Y{=}y|X{=}x) \times y \qquad (5b)$$

where *IG* is information gain and *X*, *Y* are two independent random variables. *E*(*Y*) is entropy of target variable *Y* and E (*Y*|*X*) is the conditional entropy of *Y* given *X*.

4) KNN classifier

K-Nearest neighbor is a supervised classification algorithm. KNN classifier works using "distance" metric, it locates the closest neighbors around unknown data point to classify under a specific class. "K" is the hyper parameter for KNN and its value is to be fine-tuned for achieving best accuracy. KNN is accurate yet slow. The mathematical formula for finding out the distance between any two points is given in Eq. (6) and the process is repeated with this formula to all set of points to find out the distance.

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \qquad (6)$$

The code using python is given as follows.

```
models.append((“KNN”,KNeighborsClassifier(n_neighb
ors=2)))
```

5) Linear Discriminant Analysis (LDA) classifier

LDA is a classification algorithm used for real world applications. It is a binary as well as multiclass classifier. LDA needs data standardization. LDA works as dimensionality reduction algorithm too, which reduces number of input variables for a dataset. LDA uses the Bayes Theorem to calculate the probabilities of each class and the probability that any given set of data belongs to that class. It is given in Eq. (7).

$$\text{P}(Y{=}x|X{=}x) = (Plk \times fk(x)) \,/\, \text{sum}\,(Pll \times fl(x)) \qquad (7)$$

where *x* isthe input, *k* is the output class, *plk* = Nk/n and *fk*(*x*) is the estimated probability of x belonging to class *k*.

6) Gaussian NB classifier

Gaussian Naive Bayes is a supervised classification algorithm. It is based on Bayes theorem, used to calculate conditional probability. When the predictor values are continuous and are anticipated to follow a Gaussian distribution, this classifier is used. The mathematical formula for Bayes theorem is given in Eqs. (8) and (9) shows formula to calculate probability of likelihood.

$$P\,(A|B) = P\,(A).\, P\,(B|A) \,/\, P\,(B) \qquad (8)$$

$$P(x_i\,|y) = \frac{1}{\sqrt{2\pi\sigma^2 y}} \exp\left(-\frac{(x_i - \mu y)^2}{2\sigma^2 y}\right) \qquad (9)$$

7) Random forest classifier

It is a supervised machine learning algorithm for classification problems. It uses ensembling learning by combining several classifiers to produce results.

Random forest is a group of decision trees. Bagging is a technique, where independent base classifiers like decision tree, random tree or extremely randomized tree, decide the final prediction. RF employs bagging techniques and addresses the overfitting issue [19]. The mathematical formula for calculating a nodes importance (with assumption of only two nodes) is given in Eq. (10).

$$ni_j = W_j C_j - W_{\mathrm{left}(j)} C_{\mathrm{left}(j)} W_{\mathrm{right}(j)} C_{\mathrm{right}(j)} \qquad (10)$$

Each feature can be calculated then using Eq. (11).

$$fi_i = \sum\nolimits_{j:\mathrm{node}\ j\ \mathrm{splits\ on\ feature}\ i}\, ni_j / \sum\nolimits_{k \in \mathrm{all\ nodes}} ni_k \qquad (11)$$

The algorithm for random forest algorithm is discussed.

Algorithm 2 Random Forest

```
Input: Dataset to be used
Output: Ensemble of trees

Step 1: Start
Step 2: Create a number of trees in forest B with
        following steps.
Step 3: Create a bootstrap sample dataset of size n from
        actual data set.
Step 4: Build decision tree from sample dataset in step1
        by executing step 3 to step 5.
Step 5: Select m features randomly out of all the
        features.
Step 6: Compute information gain using bootstrapped
        dataset and randomly selected m features (in
        step 3)
Step 7: Split the node into children nodes using the best
        split.
Step 8: Execute step 1 for required number of trees in
        forest B.
Step 9: Return the ensemble of trees B.
Step 10: End.
```

Table II displays parameters for classification algorithms from Pedregosa *et al.* [20].

TABLE II. VARIOUS CLASSIFICATION ALGORITHMS AND ITS PARAMETERS

Classification algorithm	Parameters
Logistic regression	LogisticRegression(penalty=“l2”,*,dual=False,tol=0.0001,C=1.0,fit_intercept=True,intercept_scaling=1,class_weight=None,random_state=None,solver=‘lbfgs’,max_iter=100,multi_class=‘auto’,verbose=0,warm_start=False,n_jobs=None,l1_ratio=None)
Decision Tree classifier	DecisionTreeClassifier(*,criterion=‘gini’,splitter=‘best’,max_depth=None,min_samples_split=2,min_samples_leaf=1,min_weight_fraction_leaf=0.0,max_features=None,random_state=None,max_leaf_nodes=None,min_impurity_decrease=0.0,class_weight=None,ccp_alpha=0.0)
Random Forest Classifier	RandomForestClassifier(n_estimators=100,*,criterion=‘gini’,max_depth=None,min_samples_split=2,min_samples_leaf=1,min_weight_fraction_leaf=0.0,max_features=‘sqrt’,max_leaf_nodes=None,min_impurity_decrease=0.0,bootstrap=True,oob_score=False,n_jobs=None,random_state=None,verbose=0,warm_start=False,class_weight=None,ccp_alpha=0.0,max_samples=None)
KNN classifier	KNeighborsClassifier(n_neighbors=5,*,weights=‘uniform’,algorithm=‘auto’,leaf_size=30,p=2,metric=‘minkowski’,metric_params=None,n_jobs=None)
Linear Discriminant Analysis	LinearDiscriminantAnalysis(solver=‘svd’,shrinkage=None,priors=None,n_components=None,store_covariance=False,tol=0.0001,covariance_estimator=None)
Gaussian NB classifier	GaussianNB(*,priors=None, var_smoothing=1e-09)
SVC (SVM)	SVC(*,C=1.0,kernel=‘rbf’,degree=3,gamma=‘scale’,coef0=0.0,shrinking=True,probability=False,tol=0.001,cache_size=200,class_weight=None,verbose=False,max_iter=1,decision_function_shape=‘ovr’,break_ties=False,random_state=None)

F. *Training and Testing*

Data is split into 70:30 ratios for training and testing. 10 fold cross validation is done using parameters. In learning phase: best parameter is chosen, data is trained, feature vector is sent to algorithm, and model gets trained. Phase 2 is getting inference from model: Data is trained, feature vector is sent to model. Model is fit and prediction is done. The best classifier is chosen and is used to make predictions out of the dataset. The model predicts labels for data to be tested. Confusion matrix analyzes the performance of classification algorithm with performance evaluation parameters such as precision, recall, support and f1-score. Table III explains confusion matrix outcome. Confusion matrix is a matrix used to evaluate the performance of classification model and compares actual target values with the values predicted machine learning model.

TABLE III. CONFUSION MATRIX OUTCOME

Outcome	Explanation
TP	True Positive, actual positives are correctly predicted.
TN	True Negative, actual negatives are correctly predicted.
FP	False Positive, actual negatives are incorrectly predicted as positive class.
FN	False Negative, actual positives are incorrectly predicted as negative class.

Some performance evaluation metrics in confusion matrix are:

- Precision refers to the number of true positives to sum of true and false predictions.
- Recall/Sensitivity refers to ability of a model to predict true positive of each available division.
- F1-score is the weighted average of sensitivity and precision.
- Support is the total number of instances of the class that actually occurred in the dataset.

The formula for precision, recall, f1-score and accuracy are given as follows.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{12}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{13}$$

$$\text{F1-score} = 2 \times \frac{\text{precision}\times\text{recall}}{\text{precision+recall}} \tag{14}$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{15}$$

IV. Results and Discussion

The proposed system uses Intel(R) Core (TM) i3-5005U CPU with 4GB Ram capacity and 64-bit processor. Python 3.9.7 version and anaconda 1.9.0 version is used. A total of 560 images (140 images for each class) are used for the experiment. We use multiple classifiers for our experiment and based on the results, Table IV shows overall validation and testing accuracy of various classifiers and Fig. 8 represent visual chart representation of validation-test accuracy of various classifiers.

TABLE IV. Overall Validation-Test Accuracy of Various Classifiers

Classification Algorithm	Validation accuracy	Testing accuracy
LR	88.26%	94.05%
LDA	67.81%	76.79%
KNN	84.20%	81.55%
CART	88.51%	94.05%
RF	**92.84%**	**97.62%**
NB	63.26%	66.07%
SVM	90.56%	96.43%

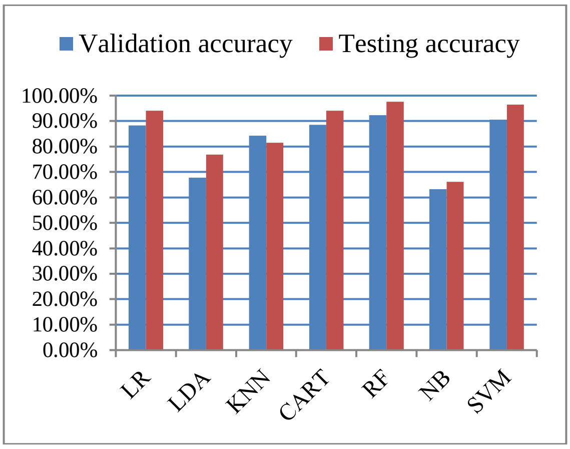


Figure 8. Validation -test accuracy of various classifiers.

Accuracy refers to the proximity of a measured value to a standard or true value. The results indicate that validation accuracy is highest for Random Forest classifier with 92.84%, SVM produces 90.56% and CART produces 88.51% and Logistic regression produces 88.26% accuracy. Testing accuracy is highest for Random Forest classifier with 97.62%, SVM produces 96.43% and CART and Logistic Regression produces 94.05% accuracy. Random forest reports less variation between validation and testing accuracy and it is chosen to be the best performing model with 97.62% testing accuracy.

Table V displays the performance evaluation parameters for Random forest classifier where weighted average of 0.97 is recorded for precision, recall shows 0.97, f1-score reports 0.97 and support reports 140. It is observed that the support parameter does not vary between models.

TABLE V. RF Classifier - Performance Evaluation Parameters

	Precision	Recall	f1-score	Support
Bacterial blight	0.97	0.96	0.96	140
Blast	0.96	0.96	0.96	140
Brown spot	0.97	0.97	0.97	140
Tungro	0.98	0.98	0.98	140
Accuracy			0.97	560
macro avg	0.97	0.97	0.97	560
weighted avg	0.97	0.97	0.97	560

Figs. 9 and 10 display confusion matrix of Random forest and logistic regression classifiers. In case of Random forest classifier, 137 images are correctly classified as bacterial blight, 134 images were correctly identified as blast, 137 images were found to be correctly identified as brown spot and 138 images were identified to be tungro correctly. RF classifier achieves 97.62% overall classification accuracy. In case of Logistic regression classifier, 131 images are classified correctly as bacterial blight, 132 images were correctly categorized as blast, 133 images were found to be exactly identified as brown spot and 131 images were classified to be tungro correctly. Logistic Regression classifier achieves 94.05% overall classification accuracy.

Figs. 11 and 12 display confusion matrix of CART and SVM classifiers. In case of CART classifier, 131 images were correctly classified as bacterial blight, 132 images were correctly identified as blast, 133 images were found to be correctly identified as brown spot and 132 images were identified to be tungro correctly. In case of SVM classifier, 131 images were classified correctly as bacterial blight, 135 images were correctly categorized as blast, 135 images were found to be exactly identified as brown spot and 138 images were classified to be tungro correctly. CART classifier achieves 94.05% overall testing accuracy and SVM achieves 96.43% testing accuracy.

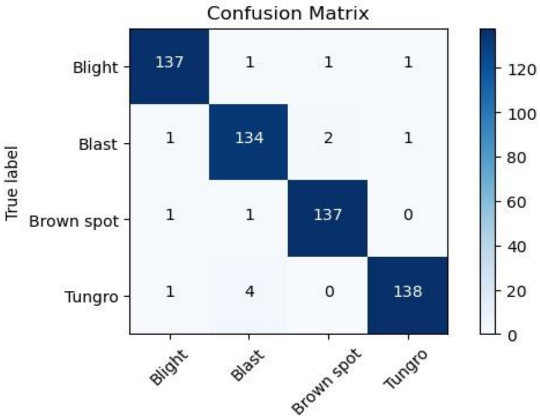


Figure 9. Random forest classifier’s confusion matrix.

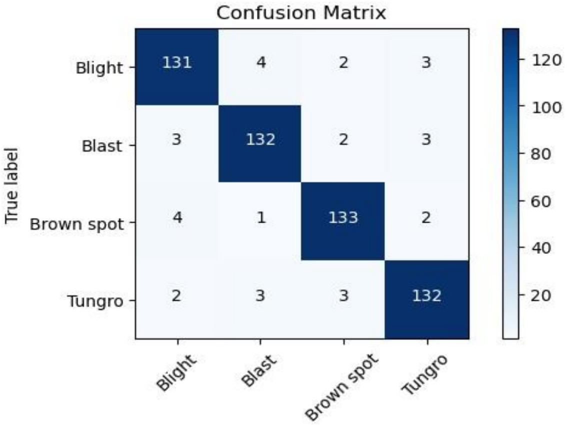


Figure 10. Logistic regression classifier’s confusion matrix.

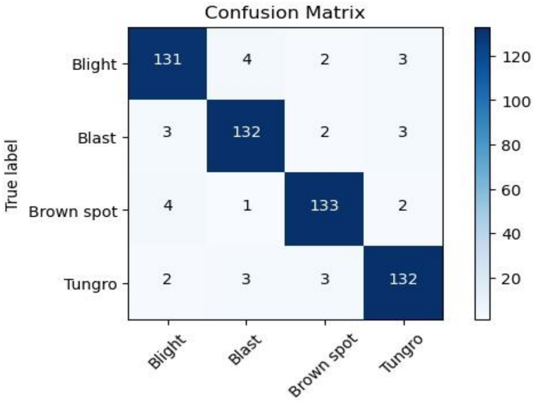


Figure 11. CART classifier’s confusion matrix.

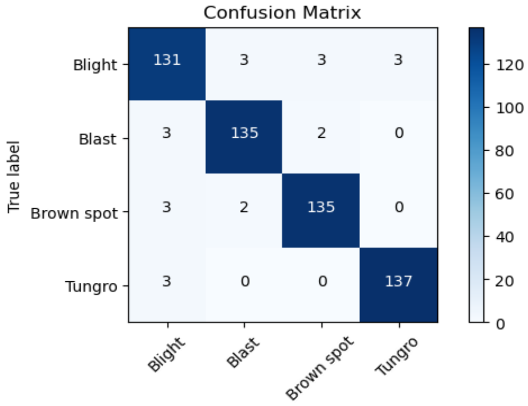


Figure 12. SVM classifier’s Confusion matrix.

Random forest algorithm produces probability of class belongings. SVM algorithm is based on statistical approaches where maximum distance between the classes is determined. It is observed in SVM that choice of ‘C’ value, the regularization parameter has an impact on model’s accuracy. Incorrect values of hyperparameter will lead to misclassification, which reduces model’s accuracy. This research sets “C” value to 2 as hyperparameter producing 90.56% validation and 96.43% testing accuracy, whereas setting “C” value to 1 will yield validation and testing accuracies 89.55% and 94.05% respectively, which is comparatively less accurate.

Our result is compared with existing methods and proposed methods shows higher accuracy. Fig. 13 shows comparison of classification accuracy with existing methods.

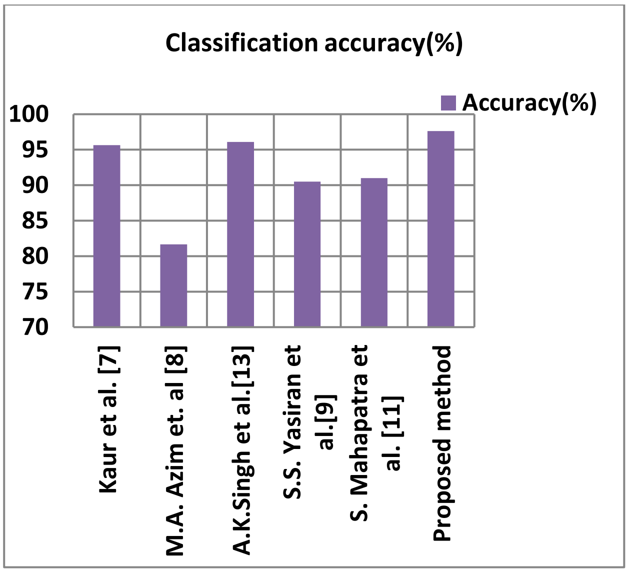


Figure 13. Comparison of classification accuracy with existing methods.

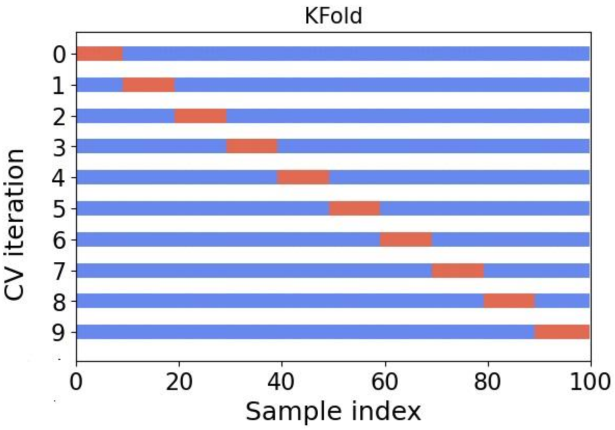


Figure 14. 10 fold cross validation.

Our results summarize that Random forest produces highest testing accuracy of 97.62% followed SVM producing 96.43% accuracy. There is a closest match in accuracy between Random forest and SVM. Cross validation is done to improve the accuracy of the model. 10 fold cross validation is done to train and test classifiers.

It helps to avoid overfitting problem. Fig. 14 shows the visualization of 10 fold cross validation.

The experimental results indicate that Random forest algorithm outperforms other machine learning algorithms and it is the best suited algorithm for this research.

V. Conclusion

Plant disease identification is crucial to mankind. This study attempts to find machine learning based approaches to diagnose and classify paddy leaf diseases. Bacterial blight, blast, tungro and brown spot are the classified diseases of paddy. The proposed work shows 97.62% testing accuracy with Random Forest classifier. 10 fold cross verification is done. Performance is evaluated with the help of confusion matrix with precision, recall, f1-score, accuracy and support as the performance evaluation metrics. Still there is a room for improvement as validation accuracy of Random forest classifier is 92.84%. The futuristic work is to improve the accuracy by including more number of training data and to implement the same problem in GUI based mobile application.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, S.K., S.B. and C.D.M.; Data curation, Formal Analysis, Methodology, Resources, Software, Validation, Visualization, Writing - original draft and Writing - review & editing, S.B. Investigation and supervision, S.K. and C.D.M.; Project administration, S.K., S.B. and C.D.M.; All authors have read and agreed to the published version of the manuscript.

References

[1] FAO in India. (2022). [Online]. Available: http://www.fao.org/india/fao-in-india/india-at-a-glance/en/

[2] S. Aggarwal, M. Suchithra, N. Chandramouli, M. Sarada, A. Verma, D. Vetrithangam, and B. A. Adugna, "Rice disease detection using artificial intelligence and machine learning techniques to improvise agro-business," *Scientific Programming*, vol. 2022, 1757888, 2022, doi: 10.1155/2022/1757888

[3] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Global Transitions Proceedings*, vol. 3, issue 1, pp. 305–310, 2022, doi: 10.1016/j.gltp.2022.03.016

[4] P. Kartikeyanand and G. Shrivastava, "Review on emerging trends in detection of plant diseases using image processing with machine learning," *International Journal of Computer Application*, vol. 174, issue 11, 2021.

[5] A. S. Zamani, L. Anand, K. P. Rane, P. Prabhu, A. M. Buttar, H. Pallathadka, A. Raghuvanshi, and B. N. Dugbakie, "Performance of machine learning and image processing in plant leaf disease detection," *Journal of Food Quality*, vol. 2022, 1598796, 2022, doi: 10.1155/2022/1598796

[6] P. Dhar, M. S. Rahman, and Z. Abedin, "Classification of leaf disease using global and local features," *IJITCS*, vol. 14, pp. 43–57, 2022, doi: 10.5815/ijitcs.2022.01.05

[7] M. Astani, M. Hasheminejad, and M. Vaghefi, "A diverse ensemble classifier for tomato disease recognition," *Computers and Electronics in Agriculture*, vol. 198, 2022, doi: 10.1016/j.compag.2022.107054

[8] N. Kaur and V. Devendran, "Plant leaf disease detection using ensemble classification and feature extraction," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, issue 11, pp. 2339–2352, 2021.

[9] M. A. Azim, M. K. Islam, M. M. Rahman, and F. Jahan, "An effective feature extraction method for rice leaf disease classification," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, issue 2, pp. 463–470, 2021, doi: 10.12928/TELKOMNIKA.v19i2.16488

[10] S. S. Yasiran, S. Salleh, and R. Mahmud, "Haralick texture and invariant moments features for breast cancer classification," in *Proc. AIP Conference*, vol. 1750, issue 1, 020022, 2016, doi: 10.1063/1.4954535

[11] A. S. Ansari, M. Jawarneh, M. Ritonga, P. Jamwal, M. S. Mohammadi, R. K. Veluri, V. Kumar, and M. A. Shah, "Improved support vector machine and image processing enabled methodology for detection and classification of grape leaf disease," *Journal of Food Quality*, vol. 2022, 9502475, 2022, doi: 10.1155/2022/9502475

[12] S. Mahapatra, S. Kannoth, R. Chiliveri, and R. Dhannawat, "Plant leaf classification and disease recognition using SVM, a machine learning approach," *Sustainable Humanosphere*, vol. 16, issue 1, pp. 1817–1825, 2020.

[13] K. Joshi and M.I. Patel, "Recent advances in local feature detector and descriptor: a literature survey," *International Journal of Multimedia Information Retrieval*, vol. 9, issue 4, pp. 231–247, 2020, doi: 10.1007/s13735-020-00200-3

[14] A. K. Singh, S. V. N. Sreenivasu, U. S. B. K. Mahalaxmi, H. Sharma, D. D. Patil, and E. Asenso, "Hybrid feature-based disease detection in plant leaf using convolutional neural network, Bayesian optimized SVM, and random forest classifier," *Journal of Food Quality*, vol. 2022, 2845320, 2022, doi: 10.1155/2022/2845320

[15] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Trends in vision-based machine learning techniques for plant disease identification: A systematic review," *Expe*rt *Systems with Applications*, 118117, doi: 10.1016/j.eswa.2022.118117

[16] A. W. S. Ibrahim and B. A. K. Atya, "Detection of diseases in rice leaf using deep learning and machine learning techniques," *Webology*, vol. 19, issue 1, 2022.

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

[18] J. C. M. Roman, J. L. V. Noguera, H. Legal-Ayala, D. P. Pinto-Roa, S. Gomez-Guerrero, and M. García Torres, "Entropy and contrast enhancement of infrared thermal images using the multiscale top-hat transform," *Entropy*, vol. 21, issue 3, p. 244, 2019.

[19] P. R. Patil and S. A. Kinariwala, "Automated diagnosis of heart disease using random forest algorithm," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 3 issue 2, pp. 579–589, 2017.

[20] Pedregosa, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.