agriculture

Article

Pre-Trained Deep Neural Network-Based Features Selection Supported Machine Learning for Rice Leaf Disease Classification

Meenakshi Aggarwal [1], Vikas Khullar [1,*], Nitin Goyal [2], Aman Singh [3,4,5], Amr Tolba [6], Ernesto Bautista Thompson [4,7] and Sushil Kumar [2]

1 Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, Punjab, India; meenakshi1017cse.phd21@chitkara.edu.in
2 Department of Computer Science and Engineering, School of Engineering and Technology, Central University of Haryana, Mahendragarh 123031, Haryana, India; dr.nitin@cuh.ac.in (N.G.); sushil.kumar@cuh.ac.in (S.K.)
3 Higher Polytechnic School, Universidad Europea del Atlántico, C/Isabel Torres 21, 39011 Santander, Spain; aman.singh@uneatlantico.es
4 Department of Engineering, Universidad Internacional Iberoamericana, Arecibo, PR 00613, USA; ernesto.bautista@unini.edu.mx
5 Uttaranchal Institute of Technology, Uttaranchal University, Dehradun 248007, Uttarakhand, India
6 Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia; atolba@ksu.edu.sa
7 Engineering Research & Innovation Group, Universidad Europea del Atlántico, C/Isabel Torres 21, 39011 Santander, Spain
* Correspondence: vikas.khullar@gmail.com or vikas.khullar@chitkara.edu.in

Citation: Aggarwal, M.; Khullar, V.; Goyal, N.; Singh, A.; Tolba, A.; Thompson, E.B.; Kumar, S. Pre-Trained Deep Neural Network-Based Features Selection Supported Machine Learning for Rice Leaf Disease Classification. *Agriculture* **2023**, *13*, 936. https://doi.org/10.3390/agriculture13050936

Academic Editor: Maciej Zaborowicz

Received: 8 April 2023
Revised: 16 April 2023
Accepted: 21 April 2023
Published: 24 April 2023

Abstract: Rice is a staple food for roughly half of the world's population. Some farmers prefer rice cultivation to other crops because rice can thrive in a wide range of environments. Several studies have found that about 70% of India's population relies on agriculture in some way and that agribusiness accounts for about 17% of India's GDP. In India, rice is one of the most important crops, but it is vulnerable to a number of diseases throughout the growing process. Farmers' manual identification of these diseases is highly inaccurate due to their lack of medical expertise. Recent advances in deep learning models show that automatic image recognition systems can be extremely useful in such situations. In this paper, we propose a suitable and effective system for predicting diseases in rice leaves using a number of different deep learning techniques. Images of rice leaf diseases were gathered and processed to fulfil the algorithmic requirements. Initially, features were extracted by using 32 pre-trained models, and then we classified the images of rice leaf diseases such as bacterial blight, blast, and brown spot with numerous machine learning and ensemble learning classifiers and compared the results. The proposed procedure works better than other methods that are currently used. It achieves 90–91% identification accuracy and other performance parameters such as precision, Recall Rate, F1-score, Matthews Coefficient, and Kappa Statistics on a normal data set. Even after the segmentation process, the value reaches 93–94% for model EfficientNetV2B3 with ET and HGB classifiers. The proposed model efficiently recognises rice leaf diseases with an accuracy of 94%. The experimental results show that the proposed procedure is valid and effective for identifying rice diseases.

Keywords: rice leaf disease; machine learning; deep learning; ensemble learning; segmentation; pre-trained models

1. Introduction

Globally, rice serves as a fundamental food source for over 3.5 billion individuals [1]. Rice, wheat, and maize are the three largest grains. Rice is a highly self-sufficient crop that is widely consumed as a primary food source in various regions worldwide [2]. It is the primary source of food all over the world in agriculture. Most people include it as a complete meal in their meals. Due to its low cost, starchy nature, and high caloric value, rice is an affordable and easily accessible food for everyone [3]. Rice crops are very important for employment on the Asian continent, and they also help to some extent reduce poverty. Rice crops require hot, humid weather to grow because it grows in water. Rice production depends on effective irrigation, which includes building dams and having good soil. India is the second-largest rice producer, producing approximately 116.42 million tonnes [4].

There are a number of factors, including soil quality, environmental factors, the choice of unfavourable crops, pest weeds, poor manure, and various plant diseases that can cause different diseases and infections in plants. Plant diseases have a major impact on agricultural production [5]. Plant diseases that are contagious are brought about by viruses, fungi, and bacteria, and their impact can vary from minor harm to fruits or leaves to the death of the plant [6]. Infected leaves can cause significant damage to rice crops and lower productivity. Once infected, they spread quickly, and the rice crop is susceptible to a number of different diseases, including blast, brown spot, bacterial blight, tungro, sheath rot, false smut, and hispa [7]. On rice leaves, these diseases' symptoms are typically visible. They can be recognised by a circular or oval spot that is coloured orange and greenish-grey. Blast is identified by a greyish-green border with a dark green outline; a brown to purple oval spot on leaves is an indication of brown spot; and bacterial blight is identified by a greenish-white lesion on the leaves. It effectively lowers the quality and quantity of the harvest. Table 1 provides a brief explanation of the diseases' key characteristics [8,9]. Different rice plant diseases can occur, which has an adverse effect on crop growth and, if they are not identified in time, could have disastrous effects on food security [10].

Table 1. Classification of various rice diseases with symptoms.

Disease	Stage	Symptoms	Important Season	Factors for Infection
Blast	In growing stage	Green-grey spot with dark green outline and more difficult to detect with grey centre and green outline	Rain shower and cooled temperature	High humidity and nitrogen level
Sheath Blight	At tillering	Greenish grey irregular spot between water and leaf blade	Rainy season	High temperature and humidity with high level of nitrogen
False Smut	At flowering to maturity	Follicles are in orange and at maturity turn greenish yellow or black	In periodic rain fall	Extreme nitrogen and high humidity
Brown Spot	Flowering to maturity	Brown to purple-brown oval spot on leaves	Periodic rain	High humidity, soil deficiency and high temperature
Bacterial Blight	Tillering to heading	Tan-greyish to white	In wet	High temperature and humidity

Leaf diseases have a direct impact on the rice crop production of a country because the plants are not consistently monitored. Farmers may not always be aware of these diseases and their occurrence periods, which can result in diseases appearing unexpectedly on any plant, ultimately affecting the overall production of rice [11]. In the conventional method, a knowledgeable expert who is capable of spotting slight variations in leaf colour visually detects disease. The method is labour-intensive, takes more time, and makes it impossible to assess the harshness and stained areas in large-scale farming accurately. Predicting and forecasting diseases affecting rice leaves is crucial for maintaining the quantity and quality of rice production. Detecting plant diseases at an early stage is crucial in agriculture as it enables prompt intervention to prevent their spread, promote healthy plant growth, and increase rice production and supply [12]. Therefore, the identification of plant diseases is currently a significant requirement in agriculture.

A large portion of India's population works in fields, and the sector accounts for about 17 percent of the country's GDP. The country of India holds the position of being the second-largest producer of rice globally, with a yield of 116.42 million tonnes [4]. Automated non-destructive methods for spotting leaf diseases have emerged as a result of recent advancements in farming technology. Farmers can benefit greatly from a rapid leaf disease detection tool [13]. In order to diagnose diseases of the rice leaf, advanced automated techniques such as image processing and machine learning must be used. A new branch of data mining called machine learning (ML) enables a programme to predict outcomes more accurately without having to be explicitly programmed. ML algorithms are frequently divided into supervised and unsupervised categories [14]. Classification refers to the process of transforming a given set of instances into a designated set of attributes or labels, commonly referred to as target attributes. DT classifiers, NN, K-NN classifiers, RF, and SVM are all used in a number of applications. DL is an enhancement of ML that effectively trains huge data, automatically picks up the input features, and produces results based on predetermined rules.

A CNN that has already been trained can be transferred to a different problem. As a result, the proposed model performs better than the model created from scratch, and the training time for the model can be reduced [15]. Transfer learning can be utilised to create a model that acts as a fixed feature extractor for a particular dataset by either fine-tuning the last few layers of the model or removing the fully connected layers. This allows the model to perform efficiently with the given dataset. Recently, DL techniques have been expanded in the agricultural sector as well. Many researchers conduct tremendous research for the early detection of paddy leaf diseases at early stages, such as in Ref. [16] author using the Minimum Distance (MDC) and the K-NN classifier to accurately classify SR, blast, and BS rice leaf diseases. However, the same idea is also presented in [17], which compares two classifiers, Minimum Distance and Naïve-based classifier, for the identification of the rice crop disease with the R2016 tool. The authors obtained a dataset of 200 digital images featuring diseased rice leaves, achieving an accuracy rate of 69% with Bayes classifier and 81.06% with MDC.

According to [18], a technique for detecting rice diseases using DCNN is proposed. The authors trained CNNs to recognize ten distinct rice leaf diseases, achieving an accuracy of 95.48% with 10-fold cross-validation. The authors of Ref. [19] propose an INC-VGGN module that combines the inception and VGGNet modules to identify plant diseases. The module involves the addition of a pooling layer and modification of the activation task. The VGGNet image net is pooled with the inception module to create the module. The proposed module achieves an average accuracy of 91.83% on public datasets and 92% in complex conditions. The authors of Ref. [20] introduced a two-layer detection method based on the RCNN algorithm for detecting Brown Rice Planthoppers (BRPH) in images. The method showed good performance in identifying BRPH, with accuracy and recall rates of 94.5% and 88.0%, respectively. The study also compared the results of this method with the YOLO v3 algorithm.

The study found that the performance of the BRPH detection algorithm was consistent, and it outperformed the YOLO v3 algorithm. The authors also introduced a client-server architecture-based technique in their discussion. There are three aspects to the scheme: a mobile phone client that allows users to upload photographs to the server; a programme on the server-side that analyses the images and displays the results to the user; and also, the server must keep all the relevant results in the database. The authors of Ref. [14] created a dataset of 5932 field images of rice leaf diseases such as tungro, BB, and BS and assessed the performance of 11 CNN models in deep learning approaches based on various parameters, including accuracy, F1-score, FPR, and training time. The results indicated that SVM outperformed transfer learning methods. The authors of Ref. [21] proposed a model for detecting rice leaf diseases such as BS, LS, and BB using hue threshold segmentation. The model also integrated a classification algorithm called gradient boosting decision tree to improve performance, achieving an accuracy of 86.58% and [5] proposed two

CNN architectures, namely Simple CNN and Inception ResNetV2, along with their hyperparameters. In Inception ResNetV2, transfer learning was used for feature extraction, and the model aggregated the data for experimentation. The model's parameters were optimised for the categorisation task, and it achieved an accuracy of 95.67%. Table 2 shows the different ML/DL algorithms that can be used to find diseases on rice leaves. The accuracy of these algorithms ranges from 86% to 95%.

Table 2. Comparative analysis of ML/DL techniques.

References	ML/DL Technique	Disease Type	Data Set Size (Images)	Improved Technique	Performance Measure/Score	Limitation
[22]	K-NN classifier with global threshold	Blast, BS	330	Segmentation	Accuracy = 0.76	Lower accuracy
[18]	Deep CNN	Blast, FS, BS, SB	500	None	Accuracy mean = 0.95	Time consuming because deep learning architectures contained several layers
[23]	LVQ with CNN	BS	500	None	Accuracy = 0.86	Only one class is used
[24]	DCNN	Rice blast	5808	None	Mean Accuracy = 0.89 AUC = 0.95	Only one rice disease discussed
[25]	Image Processing	BB, BS, blast	-	Segmentation	Accuracy = 0.91	Back propagation method was not discussed
[26]	DCNN VGG-16	BB, FS	6000	None	Accuracy 0.95	Feature extraction technique was not accurate
[21]	Extreme Gradient Boosting	BB, LS, BS	120	Segmentation	Accuracy = 0.86 F1-Score = 0.87	Less data set size
[27]	CNN with Transfer Learning (VGG 16)	Blight BS, LB	1649	Augmentation	Accuracy = 0.92	Augmentation approach is not appropriate
[20]	RCNN	Brown rice plant hooper	4600	None	Accuracy = 0.94 Recall rate = 0.88	Feature extraction technique was not appropriate
[28]	VGG16, ResNet50, ResNet101, and YOLOv3	SB, BS	5320	None	Mean F1-score = 0.74, Recall rate = 0.77 Precision = 0.74	Performance parameters are low
[29]	AlexNet Neural Network	BS, BB, LS	900	Augmentation	Accuracy = 0.9	Augmentation technique was appropriate
[7]	ANN	BS, LS	96	Segmentation	Accuracy = 0.79	Less data set
[9]	Probabilistic Neural Network (PNN)	Rice blast	1800	None	Accuracy = 0.91 F1-Score = 0.92	Only one rice leaf disease was discussed
[5]	CNN and InceptionResNeV2	Blast, BB, BS	5200	Augmentation	Accuracy = 0.95	Feature extraction technique was not appropriate
[30]	Neural Network with YOLOv3	Blast, BS Streak	6538	None	TPR = 0.78	Performance parameters are not enough

The objective is to present a model for the identification of rice leaf diseases that helps farmers identify rice leaf diseases timely and also helps to improve production. The proposed method in this paper employs pre-trained models with knowledge stored in the weights (ImageNet) that are converted into an experiment for the feature extraction process using a transfer learning technique. For classification, the approaches of machine learning and ensemble learning are used, and the outcomes are compared using different performance metrics.

The major contributions of this study are as follows:

1. Implementation of the pre-trained deep learning-based feature selection techniques on segmented images.
2. Implementation analysis of machine and ensemble learning classification techniques using pre-trained deep learning models based on selected features.
3. The experimental results show the effectiveness of the proposed procedure in comparison to existing techniques with high parameters for the classification of rice leaf diseases.

Further, this paper is organized as follows. In Section 2, the overall procedure of the proposed model is discussed. Section 3 gives the experimental results and comparative analysis between normal images and segmented images of rice leaves, and finally, a conclusion and future scope are discussed in Section 4.

2. Materials and Methods

The overall procedure of our proposed methodology for identifying the rice leaf diseases is discussed: first, a collection of rice disease images is gathered and properly labelled based on expert knowledge; then, various image processing techniques, such as image resizing, reshaping, grey colour conversion, and so on, are performed on the acquired dataset, and segmentation techniques are used to enhance the data set; and finally, the proposed method involved feeding both segmented and normal images into the model for feature extraction, which is then used to train the model. The trained model is subsequently utilised in the analysis. Thus, the final results are obtained. The proposed model was trained on the basis of the Algorithm 1.

Algorithm 1: Proposed Algorithm for Pre-trained Deep Neural Network-Based Features Selection Supported Machine Learning for Rice Leaf Disease Classification

Input: Infected rice leaf images $((X_i, Y_i) \ldots \ldots (X_m, Y_m))$
Output: Class of rice leaf disease

1. For each K:=1 -> P, where P is the total number of input leaf image do
2. Convert K_{th} image into RGB leaf image.
3. Read K_{th} RGB leaf image.
4. Resize K_{th} image to $(h \times w)$ size.
5. Apply segmentation technique to each image. For each T:=1 -> t, where t is the number of pre-trained model do Load each model by Initializing imagenet weights and extract feature from the second last layer. Update weights $w_k = w_{k-1} - a^* \hat{m}/(\sqrt{vk} + \epsilon$ where k is the class index, w is the weights, a* learning rate, $\hat{m}$ and vk is the first and second bias. Store extracted feature in $F_{pt} = i \times FV$, where i is the number of sample images and FV is the feature vector. End for
6. Input extracted feature (F_{pt}) for classification to classify function y = f(x).

Further, this section divides the proposed work's process into several steps for identifying rice leaf diseases, as shown in Figure 1.

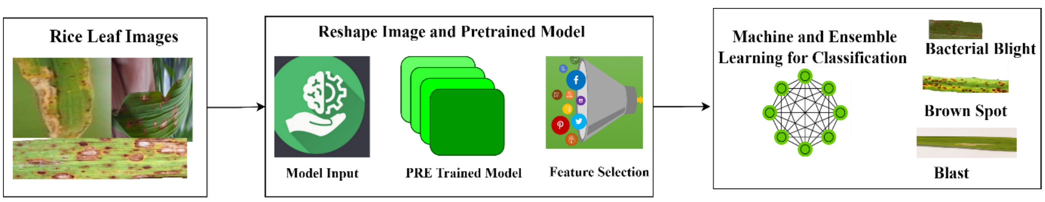

Figure 1. The overall flow of rice leaf disease prediction.

2.1. Data Acquisition and Pre-Processing

In the experiments, we collected 551 images of rice leaf diseases from the internet [31]. It includes the three different types of diseases that affect rice leaves: BB, BS, and blast. Figure 2 displays a few sample images of leaf diseases. All the images are properly labelled and saved in JPG format. There are 551 images in total, of which 192 depict bacterial leaf blight, 159 depict blast, and 200 depict the brown spot discussed in Table 3. There is only one disease in each image. The data set is divided into training and test sets in an 80:20 ratio. Initially, the model was trained on 80 percent of the training dataset and 20 percent of the testing dataset to validate a trained model.

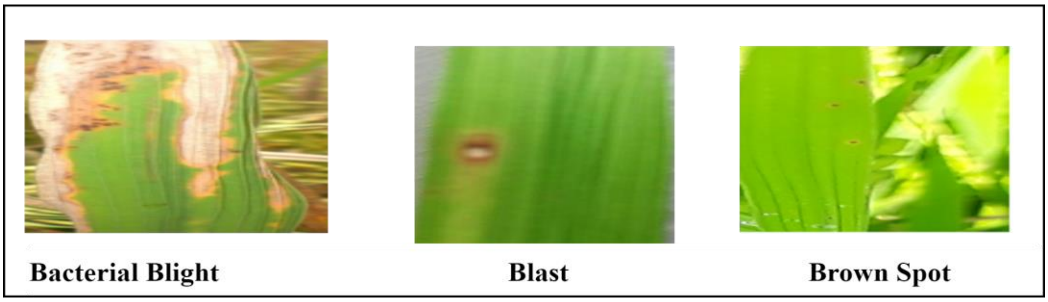

Figure 2. Images of rice leaf diseases.

Table 3. Details of images present in dataset.

Disease Name	No. of Images	Images for Training (80%)	Images for Validation (20%)
Bacterial leaf Blight	192	154	38
Brown Spot	200	160	40
Blast	159	127	32
Total	551	441	110

In order to transform the raw data into useful data and improve the effectiveness and accuracy of a model, image pre-processing is required. Pre-processing is important because it makes it possible for input images to be processed more smoothly [8]. Pre-processing is thought to be a crucial step in processing the images so they are suitable for the detection process [32]. In order to enhance the quality of the images in the collected dataset, various pre-processing techniques such as image resizing, reshaping, and converting to greyscale are applied, resulting in sharper images. Each image is uniformly processed and resized to 224 × 224 pixels. Formal methods can be used to verify the correctness and safety of AI-based solutions, including data collection and processing, by providing rigorous mathematical models and techniques for verification [33]. This can involve using techniques such as formal proof or program analysis to check that the algorithms are correct and do not have any unintended behaviours [34].

2.2. Segmentation

The pre-processed images of rice leaves are fed into the segmentation module to provide high-dimensional data segmentation. Segmentation is used to divide an image into areas that are homogeneous in terms of one or more characteristics or features (also known as classes or subsets). Segmentation is a crucial tool for image processing and has numerous applications such as feature extraction, image measurement, and display [7]. As a crucial stage in the image processing pipeline, segmentation enables us to locate and extract desired features from a given image. However, for all imaging applications, there is not a single standard segmentation method that can deliver satisfactory results [25]. Depending on the classification scheme, there are numerous ways to categorise segmentation techniques such as manual, automatic, and semi-automatic; region- and global-based approaches; low-level thresholding; model-based thresholding, etc. Each method has its own pros and cons. Segments from the images are represented as a

$$S = \{S1, S2, \ldots, Sd, \ldots, Sn\}$$

where n is the total number of segments in the image and sd is the dth segment of the image.

In this study, we separated disease spots from images of rice leaf disease, as shown in Figure 3. In order to extract features from the images, we used watershed and graph cut techniques for segmentation as mentioned in Ref. [35]. Compared to the conventional threshold segmentation method, this method produces better segmentation results. The two main goals of the image segmentation algorithm are as follows:

(1) It can increase the quality of the image and reduce background noise in the lesion image, which will increase recognition accuracy.
(2) It can decrease the volume of data, which will shorten the program's execution time. To shorten the program's runtime and increase the program's recognition effectiveness.

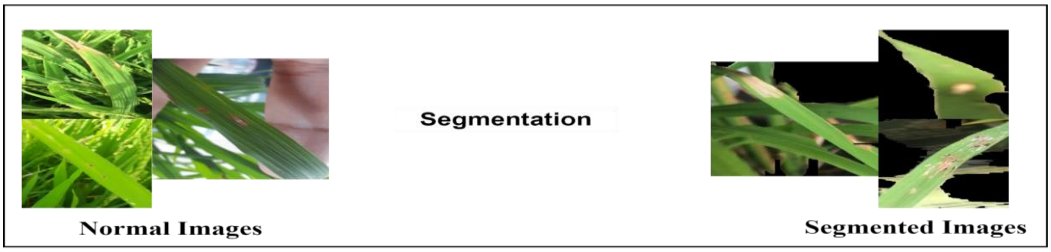

Figure 3. Segmented image samples.

When applied to complex images, the watershed and graph cut algorithms perform better than thresholding and contour detection. Remove the background and foreground elements first, and then use the markers to find the precise edges. Generally speaking, this algorithm aids in the detection of touching and overlapping objects in images. For markers, it is possible for the user to define them by manually clicking to obtain the coordinates for the markers or by using predefined algorithms to reduce noise, such as thresholding or any morphological operations [36].

2.3. Feature Extraction Using Pre-Trained Models

A quick and effective way to utilise the features that a neural network has already been trained through feature extraction. The crucial component of the deep learning network is feature extraction. There are numerous pooling and convolutional layers in it. It aids in the extraction of image features useful for target positioning and identification [37]. To improve our model(s), we could experiment with various configurations, such as adding more layers, changing the learning rate, adjusting the number of neurons per layer, and so on. Fortunately, by using pre-trained models, we can speed up the process [27]. These models save computational resources and time. Pre-trained models, or neural networks that have been trained on large-scale datasets, can be reused for subsequent tasks. These could be used for extracting the features. By choosing the right feature extractor, system performance is improved. There are numerous pre-trained models that perform different tasks, such as Xception, VGG16, VGG19, ResNet101V2, InceptionNetV2, DenseNet, EfficientNetV2, NasNet, MobileNet, ResNet50, etc.

In our research area, we used 32 pre-trained models for feature extraction. Using large datasets such as ImageNet, pre-trained models VGG, ResNet, and Inception have learned to extract meaningful visual features from images. Inputs to subsequent processes such as object recognition, segmentation, and classification can be derived from these characteristics. Each segment is modified for the feature extraction process in order to increase the accuracy of the features in identifying rice leaf diseases. This frequently results in excellent outcomes with fewer data.

The approach described speeds up the training process and improves the accuracy by using a specific model architecture consisting of various layers, including reshape, flatten, dense, dropout, and activation functions. A global pooling layer such as max global pooling or average global pooling can be utilised to summarise the activations for use in a classifier or as a feature vector representation of the input [38]. In this study, a new classifier model is constructed using the output of a layer in the model that precedes the output layer responsible for classifying rice leaf disease images. Table 4 presents the feature extractors used in this study, along with their input shape, number of parameters, size, and feature layer.

Table 4. Feature extractor with Imagenet Weights used in this work.

Model	Input Shape	Selected Features Size	Number of Parameters	Memory (in Bytes)	Feature Layer
Xception	(229, 229, 3)	2048	20,861,480	272,784,948	Global Average Pooling 2D
VGG16	(224, 224, 3)	4096	134,260,544	195,307,328	Dense
VGG19	(224, 224, 3)	4096	139,570,240	205,835,328	Dense
ResNet50	(224, 224, 3)	2048	23,587,712	172,064,560	Global Average Pooling 2D
ResNet50V2	(224, 224, 3)	2048	23,564,800	149,085,616	Global Average Pooling 2D
ResNet101	(224, 224, 3)	2048	42,658,176	266,198,320	Global Average Pooling 2D
ResNet101V2	(224, 224, 3)	2048	42,626,560	247,667,120	Global Average Pooling 2D
ResNet152	(224, 224, 3)	2048	58,370,944	374,636,336	Global Average Pooling 2D
ResNet152V2	(224, 224, 3)	2048	58,331,648	361,348,528	Global Average Pooling 2D
InceptionV3	(229, 229, 3)	2048	21,802,784	152,016,332	Global Average Pooling 2D
InceptionResNetV2	(229, 229, 3)	1536	54,336,736	379,140,364	Global Average Pooling 2D
MobileNet	(224, 224, 3)	1000	4,253,864	71,638,760	Reshape
DenseNet121	(224, 224, 3)	1024	7,037,504	206,739,952	Global Average Pooling 2D
DenseNet169	(224, 224, 3)	1664	12,642,880	253,015,536	Global Average Pooling 2D
DenseNet201	(224, 224, 3)	1920	18,321,984	327,486,960	Global Average Pooling 2D
NASNetMobile	(224, 224, 3)	1056	4,269,716	115,028,536	Global Average Pooling 2D
NASNetLarge	(331, 331, 3)	4032	84,916,818	1,247,153,502	Global Average Pooling 2D
EfficientNet B0	224, 224, 3	1280	4,049,571	105,116,063	Dropout
EfficientNet B1	240, 240, 3	1280	6,575,239	167,863,763	Dropout
EfficientNet B2	260, 260, 3	1408	7,768,569	212,211,693	Dropout
EfficientNet B3	300, 300, 3	1536	10,783,535	361,891,419	Dropout
EfficientNet B4	380, 380, 3	1792	17,673,823	739,756,747	Dropout
EfficientNet B5	456, 456, 3	2048	28,513,527	1,464,166,467	Dropout
EfficientNet B6	528, 528, 3	2304	40,960,143	2,466,985,915	Dropout
EfficientNet B7	600, 600, 3	2560	64,097,687	4,252,866,467	Dropout
EfficientNetV2B0	224, 224, 3	1280	5,919,312	70,588,560	Dropout
EfficientNetV2B1	240, 240, 3	1280	6,931,124	107,709,828	Dropout
EfficientNetV2B2	260, 260, 3	1408	8,769,374	143,981,142	Dropout
EfficientNetV2B3	300, 300, 3	1536	12,930,622	228,894,934	Dropout
EfficientNetV2S	384, 384, 3	1280	20,331,360	512,960,320	Dropout
EfficientNetV2M	480, 480, 3	1280	53,150,388	1,301,769,700	Dropout
EfficientNetV2L	480, 480, 3	1280	117,746,848	2,317,398,688	Dropout

2.4. Classification

Classification is a supervised learning technique in which input data is mapped to a specific class. It is critical to perform data mining and classify data obtained from a database. Sometimes combinations of more classifiers give reliable and accurate results as compared to a single classification model [39]. Various machine learning and ensemble learning algorithms were applied to detect rice leaf diseases. In this work, ten classification algorithms were applied to detect the diseases. On the normal data set and segmented dataset, we apply DT, QDA, K-NN, AB, GNB, LR, RF, ET, HGB, and MLP ML to the base classifier. We use

32 pre-trained models to extract features from different shapes and numerous classifiers to classify different disease classes. Further details are mentioned in Figure 4.

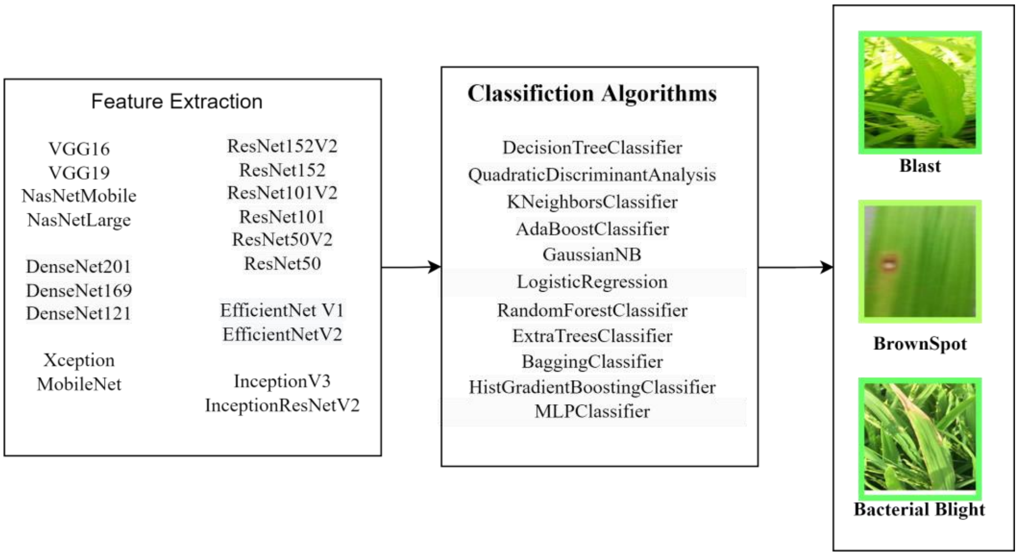

Figure 4. Classification process.

2.5. Experimental Setup and Evaluation Metrics

For this experiment, a Windows 10 PC, a Jupyter notebook, 8 GB of storage space on Google Drive, and a 64-bit operating system were utilised. The Keras 2.4.3 framework and Tensorflow backend were employed to facilitate the training and validation processes of the deep neural network. The crucial phase in the proposed model is the evaluation stage, which enables the calculation of the discrepancy between the predicted and actual value. This inference will help us achieve a consistently reliable model for identifying rice diseases. There are a number of parameters, such as accuracy, precision, recall rate, F1-Score, Matthews Coefficient, and Kappa Statistics, as discussed in Table 5 [40]. In the next section, we discuss the various results that were achieved during the implementation process and compare the results.

Table 5. Performance Parameters.

Metrics	Definition	Formula
Accuracy	Comparison between actual and predicted value	Accuracy = (TP + TN)/TP + FP + TN + FN)
Precision	Actual corrected positive prediction	Precision = TP/(TP + FP)
Recall rate	Actual positive incorrected prediction	Recall = TP/(TP + FN)
F1-score	Single value for both precision and recall rate	F1-Score = 2TP/(2TP + FP + FN)
MC (Matthews Coefficient)	Used to measure of the quality of binary and multiclass classification.	MC = (TP × TN)(FP × FN)/ $\sqrt{TP + FP)(TP + FN)(TN + FN)}$
KP (Kappa Statistics)	Used to measure the inter-rater reliability for categorical items.	K = (po − pe)/(1 − pe)

3. Results

This section demonstrates and discusses the outcome obtained using the suggested methods. By using a dataset on rice leaf diseases, this section describes in detail the accuracy, precision, recall rate, F1-Score, Matthews Coefficient, and Kappa Statistics of the evaluation of the proposed technique with respect to conventional strategies. Comparative analysis of different machine learning and deep learning approaches with normal and segmented datasets is discussed in this section. In case 1, results are discussed on the basis of a normal image set. In case 2, an analysis of the segmented image set is discussed, and in case 3, a comparative analysis of results from the normal and segmented image sets is discussed.

3.1. Analysis of Normal Data

An analysis of the normal data on the basis of accuracy is represented in Table 6, and it is noted that the maximum accuracy achieved was 91% for the pre-trained model EfficientNetB3 with ET and HGB classifiers and 90% for EfficientNetV2B3 with ET classifier. Table 7 shows an analysis based on precision for normal data. It is clear that the classifier HGB gave the highest precision value of 92% with model EfficientNetV2B3 and 91% with EfficientNetV2B3, EfficientNetB3 Models with ET, and HGB classifiers.

Table 6. Accuracy on Normal Data.

Classifiers/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.7	0.35	0.63	0.66	0.66	0.65	0.79	0.8	0.86	0.69
VGG19	0.67	0.39	0.67	0.72	0.59	0.64	0.82	0.83	0.83	0.79
VGG16	0.61	0.36	0.64	0.66	0.69	0.56	0.82	0.81	0.78	0.73
ResNet152V2	0.63	0.38	0.56	0.66	0.57	0.54	0.76	0.78	0.8	0.65
ResNet152	0.74	0.48	0.72	0.77	0.56	0.66	0.83	0.83	0.84	0.7
ResNet101V2	0.56	0.41	0.56	0.67	0.63	0.59	0.74	0.71	0.79	0.7
ResNet101	0.71	0.44	0.68	0.71	0.62	0.7	0.84	0.81	0.81	0.77
ResNet50V2	0.54	0.38	0.53	0.71	0.65	0.53	0.76	0.78	0.73	0.66
ResNet50	0.72	0.47	0.7	0.71	0.6	0.65	0.84	0.81	0.8	0.73
NASNetMobile	0.61	0.45	0.62	0.67	0.51	0.57	0.77	0.77	0.78	0.66
NASNetLarge	0.7	0.44	0.62	0.66	0.36	0.73	0.81	0.78	0.8	0.72
MobileNet	0.61	0.4	0.55	0.61	0.61	0.69	0.79	0.81	0.8	0.7
InceptionV3	0.65	0.39	0.64	0.66	0.77	0.61	0.79	0.83	0.77	0.76
InceptionResNetV2	0.64	0.48	0.68	0.65	0.79	0.67	0.79	0.82	0.8	0.72
EfficientNetV2S	0.72	0.4	0.76	0.69	0.71	0.69	0.91	0.89	0.89	0.73
EfficientNetV2M	0.66	0.38	0.47	0.71	0.67	0.57	0.77	0.79	0.79	0.66
EfficientNetV2L	0.66	0.38	0.66	0.56	0.78	0.64	0.82	0.87	0.87	0.74
EfficientNetV2B3	0.68	0.43	0.72	0.81	0.78	0.76	0.85	0.9	0.89	0.85
EfficientNetV2B2	0.69	0.39	0.61	0.69	0.61	0.56	0.8	0.78	0.81	0.66
EfficientNetV2B1	0.71	0.41	0.61	0.73	0.6	0.63	0.78	0.8	0.82	0.69
EfficientNetV2B0	0.63	0.33	0.63	0.67	0.6	0.57	0.76	0.76	0.74	0.77
EfficientNetB7	0.83	0.46	0.65	0.73	0.79	0.71	0.86	0.86	0.83	0.74
EfficientNetB6	0.71	0.43	0.72	0.63	0.67	0.6	0.9	0.86	0.86	0.74
EfficientNetB5	0.64	0.44	0.69	0.79	0.71	0.63	0.85	0.88	0.83	0.74
EfficientNetB4	0.69	0.46	0.76	0.78	0.71	0.65	0.81	0.83	0.85	0.74
EfficientNetB3	0.76	0.4	0.7	0.81	0.87	0.69	0.89	0.91	0.91	0.86
EfficientNetB2	0.74	0.5	0.61	0.65	0.67	0.66	0.77	0.82	0.78	0.71
EfficientNetB1	0.68	0.4	0.67	0.72	0.63	0.69	0.8	0.83	0.8	0.71
EfficientNetB0	0.65	0.36	0.7	0.73	0.63	0.57	0.8	0.83	0.82	0.71
DenseNet201	0.6	0.34	0.56	0.73	0.64	0.63	0.79	0.77	0.81	0.65
DenseNet169	0.59	0.36	0.64	0.7	0.53	0.66	0.81	0.78	0.77	0.67
DenseNet121	0.72	0.4	0.71	0.74	0.4	0.62	0.79	0.81	0.83	0.72

Table 7. Precision on Normal Data.

Classifiers/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.67	0.23	0.52	0.69	0.64	0.44	0.72	0.77	0.83	0.64
VGG19	0.62	0.23	0.6	0.69	0.59	0.62	0.77	0.78	0.79	0.74
VGG16	0.55	0.21	0.62	0.67	0.67	0.39	0.77	0.75	0.73	0.67
ResNet152V2	0.59	0.23	0.49	0.59	0.55	0.36	0.66	0.7	0.76	0.61
ResNet152	0.69	0.3	0.67	0.72	0.61	0.61	0.78	0.78	0.79	0.67
ResNet101V2	0.51	0.25	0.5	0.63	0.62	0.5	0.67	0.62	0.74	0.63
ResNet101	0.66	0.3	0.65	0.66	0.71	0.47	0.79	0.75	0.76	0.7
ResNet50V2	0.48	0.24	0.45	0.66	0.69	0.54	0.69	0.73	0.67	0.63
ResNet50	0.67	0.28	0.57	0.67	0.62	0.58	0.79	0.75	0.74	0.7
NASNetMobile	0.57	0.29	0.57	0.63	0.44	0.53	0.71	0.71	0.72	0.61
NASNetLarge	0.66	0.23	0.57	0.64	0.6	0.5	0.77	0.74	0.78	0.6
MobileNet	0.59	0.15	0.52	0.56	0.69	0.64	0.75	0.74	0.76	0.65
InceptionV3	0.62	0.22	0.53	0.69	0.73	0.43	0.75	0.78	0.71	0.71
InceptionResNetV2	0.6	0.28	0.63	0.65	0.76	0.47	0.75	0.79	0.76	0.67
EfficientNetV2S	0.67	0.24	0.7	0.63	0.7	0.62	0.89	0.85	0.85	0.68
EfficientNetV2M	0.61	0.24	0.4	0.66	0.63	0.58	0.68	0.72	0.7	0.55
EfficientNetV2L	0.59	0.21	0.58	0.6	0.73	0.66	0.77	0.85	0.85	0.67
EfficientNetV2B3	0.62	0.26	0.66	0.75	0.72	0.84	0.81	0.91	0.92	0.81
EfficientNetV2B2	0.66	0.27	0.56	0.67	0.62	0.48	0.74	0.72	0.75	0.62
EfficientNetV2B1	0.67	0.21	0.57	0.71	0.53	0.46	0.73	0.76	0.78	0.66
EfficientNetV2B0	0.59	0.22	0.59	0.63	0.68	0.62	0.71	0.7	0.68	0.72
EfficientNetB7	0.79	0.28	0.56	0.69	0.75	0.65	0.88	0.9	0.79	0.7
EfficientNetB6	0.71	0.27	0.68	0.66	0.7	0.43	0.88	0.82	0.82	0.69
EfficientNetB5	0.61	0.28	0.66	0.74	0.69	0.42	0.82	0.86	0.81	0.71
EfficientNetB4	0.65	0.28	0.67	0.74	0.7	0.46	0.78	0.79	0.8	0.7
EfficientNetB3	0.72	0.24	0.66	0.78	0.84	0.72	0.89	0.91	0.9	0.84
EfficientNetB2	0.67	0.32	0.57	0.65	0.64	0.63	0.7	0.76	0.74	0.66
EfficientNetB1	0.64	0.25	0.61	0.72	0.66	0.62	0.75	0.78	0.77	0.66
EfficientNetB0	0.63	0.18	0.63	0.68	0.63	0.53	0.74	0.76	0.76	0.65
DenseNet201	0.55	0.19	0.53	0.71	0.63	0.43	0.72	0.7	0.75	0.61
DenseNet169	0.56	0.2	0.57	0.66	0.58	0.55	0.75	0.71	0.71	0.62
DenseNet121	0.69	0.26	0.69	0.71	0.55	0.61	0.73	0.76	0.78	0.68

Following that, recall rate analysis is represented in Table 8, and it is seen that EfficientNetB3 with the ET and HGB classifiers achieved the highest recall rate of 89. Model EfficientNetV2B3 with the HGB classifier achieved a recall rate of 86%. Moreover, pre-trained models EfficientNetB5, EfficientNetB6, and EfficientV2S with ET, HGB classifier gave an 84% recall rate value. Next, analysis was conducted on the basis of the metric F1-Score, and from Table 9, it was observed that model EfficientNetB3 with ET and HGB gave the maximum F1-Score value, i.e., 90%. EfficientNetV2B3 with the HGB classifier gave a value of 89%.

Table 8. Recall Rate on Normal Data.

Classifiers/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.67	0.39	0.53	0.67	0.63	0.52	0.69	0.76	0.84	0.63
VGG19	0.63	0.38	0.6	0.68	0.56	0.55	0.78	0.78	0.78	0.74
VGG16	0.55	0.36	0.62	0.68	0.68	0.45	0.78	0.76	0.72	0.66
ResNet152V2	0.47	0.41	0.45	0.66	0.66	0.48	0.68	0.72	0.66	0.64
ResNet152	0.67	0.47	0.58	0.67	0.56	0.58	0.79	0.75	0.74	0.7
ResNet101V2	0.58	0.39	0.49	0.59	0.49	0.43	0.66	0.7	0.74	0.61
ResNet101	0.68	0.45	0.68	0.72	0.52	0.61	0.79	0.78	0.8	0.68
ResNet50V2	0.51	0.41	0.5	0.61	0.57	0.49	0.66	0.62	0.71	0.62
ResNet50	0.66	0.52	0.67	0.67	0.58	0.56	0.8	0.76	0.77	0.69
NASNetMobile	0.57	0.51	0.56	0.62	0.43	0.52	0.7	0.71	0.7	0.61
NASNetLarge	0.66	0.38	0.57	0.63	0.46	0.59	0.76	0.73	0.74	0.6
MobileNet	0.61	0.32	0.51	0.55	0.54	0.6	0.72	0.72	0.75	0.62
InceptionV3	0.61	0.36	0.54	0.66	0.71	0.48	0.72	0.75	0.71	0.69
InceptionResNetV2	0.59	0.48	0.63	0.63	0.76	0.54	0.76	0.77	0.74	0.67
EfficientNetV2S	0.68	0.38	0.71	0.61	0.72	0.62	0.87	0.84	0.84	0.69
EfficientNetV2M	0.62	0.39	0.4	0.65	0.62	0.48	0.68	0.71	0.68	0.55
EfficientNetV2L	0.59	0.35	0.58	0.59	0.73	0.52	0.74	0.8	0.85	0.66
EfficientNetV2B3	0.62	0.43	0.65	0.76	0.72	0.62	0.78	0.85	0.86	0.79
EfficientNetV2B2	0.68	0.47	0.56	0.67	0.58	0.49	0.73	0.71	0.74	0.6
EfficientNetV2B1	0.68	0.35	0.57	0.73	0.5	0.5	0.72	0.76	0.77	0.61
EfficientNetV2B0	0.57	0.35	0.59	0.63	0.64	0.56	0.71	0.69	0.68	0.73
EfficientNetB7	0.79	0.45	0.56	0.69	0.72	0.58	0.83	0.8	0.78	0.67
EfficientNetB6	0.73	0.43	0.67	0.66	0.67	0.47	0.86	0.83	0.84	0.68
EfficientNetB5	0.6	0.48	0.68	0.74	0.69	0.5	0.82	0.84	0.78	0.71
EfficientNetB4	0.64	0.45	0.67	0.75	0.71	0.52	0.8	0.81	0.81	0.71
EfficientNetB3	0.73	0.39	0.65	0.8	0.85	0.59	0.87	0.89	0.89	0.84
EfficientNetB2	0.67	0.5	0.57	0.63	0.65	0.61	0.69	0.77	0.75	0.66
EfficientNetB1	0.63	0.42	0.6	0.73	0.64	0.62	0.75	0.78	0.79	0.66
EfficientNetB0	0.64	0.31	0.61	0.69	0.61	0.49	0.74	0.75	0.74	0.63
DenseNet201	0.55	0.32	0.52	0.71	0.58	0.5	0.72	0.7	0.76	0.61
DenseNet169	0.56	0.31	0.57	0.65	0.56	0.55	0.75	0.7	0.7	0.62
DenseNet121	0.7	0.45	0.68	0.71	0.46	0.58	0.73	0.76	0.79	0.69

Table 9. F1-Score on Normal Data.

Classifiers/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.67	0.28	0.52	0.64	0.62	0.47	0.7	0.77	0.83	0.63
VGG19	0.62	0.29	0.6	0.68	0.55	0.54	0.78	0.78	0.78	0.74
VGG16	0.55	0.27	0.62	0.64	0.66	0.4	0.78	0.76	0.73	0.66
ResNet152V2	0.47	0.3	0.44	0.66	0.62	0.47	0.68	0.72	0.66	0.63
ResNet152	0.66	0.35	0.55	0.66	0.53	0.58	0.79	0.75	0.74	0.69
ResNet101V2	0.58	0.29	0.49	0.59	0.49	0.39	0.65	0.7	0.74	0.6
ResNet101	0.68	0.35	0.67	0.72	0.47	0.61	0.79	0.78	0.79	0.66
ResNet50V2	0.51	0.31	0.5	0.61	0.57	0.48	0.66	0.62	0.71	0.62
ResNet50	0.66	0.36	0.65	0.66	0.57	0.51	0.79	0.75	0.76	0.69
NASNetMobile	0.57	0.36	0.55	0.62	0.42	0.52	0.71	0.71	0.7	0.61
NASNetLarge	0.66	0.28	0.57	0.62	0.37	0.54	0.76	0.73	0.75	0.59
MobileNet	0.59	0.21	0.51	0.54	0.54	0.61	0.73	0.73	0.75	0.63
InceptionV3	0.6	0.27	0.53	0.64	0.72	0.44	0.73	0.76	0.71	0.7
InceptionResNetV2	0.59	0.35	0.63	0.62	0.76	0.5	0.75	0.78	0.74	0.67
EfficientNetV2S	0.67	0.29	0.7	0.6	0.69	0.62	0.88	0.85	0.85	0.68
EfficientNetV2M	0.61	0.29	0.4	0.65	0.63	0.46	0.68	0.71	0.68	0.54
EfficientNetV2L	0.59	0.26	0.58	0.55	0.73	0.5	0.75	0.81	0.85	0.66
EfficientNetV2B3	0.62	0.32	0.66	0.76	0.72	0.58	0.79	0.87	0.89	0.8
EfficientNetV2B2	0.66	0.32	0.56	0.65	0.57	0.48	0.73	0.71	0.74	0.6
EfficientNetV2B1	0.68	0.25	0.57	0.7	0.48	0.45	0.72	0.76	0.77	0.62
EfficientNetV2B0	0.57	0.26	0.59	0.62	0.59	0.53	0.7	0.69	0.68	0.72
EfficientNetB7	0.79	0.34	0.55	0.68	0.72	0.56	0.85	0.83	0.78	0.68
EfficientNetB6	0.7	0.33	0.67	0.62	0.65	0.42	0.87	0.82	0.83	0.68
EfficientNetB5	0.6	0.34	0.67	0.74	0.68	0.46	0.82	0.85	0.79	0.7
EfficientNetB4	0.64	0.34	0.67	0.73	0.69	0.48	0.78	0.8	0.8	0.7
EfficientNetB3	0.72	0.29	0.65	0.78	0.84	0.59	0.87	0.9	0.9	0.84
EfficientNetB2	0.67	0.39	0.57	0.62	0.63	0.61	0.69	0.77	0.74	0.66
EfficientNetB1	0.63	0.31	0.6	0.7	0.59	0.61	0.75	0.78	0.77	0.66
EfficientNetB0	0.62	0.23	0.61	0.68	0.58	0.48	0.74	0.76	0.75	0.63
DenseNet201	0.55	0.24	0.52	0.69	0.58	0.46	0.72	0.7	0.75	0.61
DenseNet169	0.55	0.24	0.57	0.65	0.49	0.54	0.75	0.7	0.7	0.62
DenseNet121	0.69	0.32	0.69	0.7	0.4	0.56	0.73	0.76	0.79	0.68

Analysis on the basis of the Matthews Coefficient and the Kappa Statistics is represented in Tables 10 and 11, and it is noted that the maximum value is 86% for both the Matthews Coefficient and the Kappa Statistics with model EfficientNetB3 and ET, HGB classifiers. For model EfficientNetV2B3 with Classifiers ET, the Matthews Coefficient value is 85% and Kappa Coefficient value is 84%.

Table 10. Matthews Coefficient Value.

Classifiers/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.53	0.1	0.4	0.53	0.49	0.43	0.66	0.68	0.78	0.51
VGG19	0.49	0.09	0.48	0.58	0.39	0.42	0.71	0.73	0.73	0.66
VGG16	0.38	0.03	0.43	0.52	0.53	0.28	0.71	0.7	0.64	0.57
ResNet152V2	0.27	0.11	0.25	0.54	0.49	0.26	0.61	0.64	0.58	0.46
ResNet152	0.57	0.23	0.52	0.56	0.39	0.44	0.74	0.69	0.68	0.59
ResNet101V2	0.43	0.06	0.3	0.46	0.33	0.23	0.6	0.64	0.67	0.44
ResNet101	0.6	0.23	0.57	0.64	0.36	0.46	0.73	0.73	0.75	0.55
ResNet50V2	0.31	0.11	0.3	0.48	0.41	0.33	0.59	0.54	0.66	0.52
ResNet50	0.54	0.26	0.51	0.55	0.43	0.51	0.75	0.7	0.7	0.62
NASNetMobile	0.38	0.24	0.4	0.5	0.21	0.32	0.62	0.63	0.64	0.46
NASNetLarge	0.53	0.12	0.39	0.49	0.24	0.57	0.7	0.65	0.68	0.55
MobileNet	0.4	0.02	0.28	0.39	0.4	0.5	0.66	0.69	0.68	0.52
InceptionV3	0.46	0.1	0.41	0.53	0.63	0.36	0.66	0.73	0.63	0.61
InceptionResNetV2	0.44	0.25	0.5	0.49	0.67	0.46	0.66	0.71	0.68	0.56
EfficientNetV2S	0.57	0.06	0.61	0.52	0.58	0.5	0.86	0.83	0.83	0.58
EfficientNetV2M	0.47	0.12	0.14	0.54	0.47	0.34	0.62	0.66	0.66	0.44
EfficientNetV2L	0.46	0.02	0.45	0.38	0.65	0.45	0.71	0.8	0.8	0.59
EfficientNetV2B3	0.49	0.16	0.55	0.7	0.65	0.61	0.76	0.85	0.83	0.76
EfficientNetV2B2	0.53	0.19	0.38	0.54	0.43	0.3	0.67	0.64	0.69	0.47
EfficientNetV2B1	0.55	0.05	0.38	0.6	0.36	0.42	0.64	0.68	0.71	0.52
EfficientNetV2B0	0.43	0.06	0.41	0.49	0.47	0.36	0.62	0.62	0.6	0.63
EfficientNetB7	0.73	0.19	0.43	0.6	0.66	0.53	0.78	0.78	0.73	0.59
EfficientNetB6	0.58	0.13	0.56	0.47	0.54	0.36	0.85	0.78	0.79	0.59
EfficientNetB5	0.43	0.23	0.52	0.67	0.56	0.38	0.76	0.81	0.73	0.6
EfficientNetB4	0.53	0.19	0.61	0.66	0.57	0.43	0.71	0.74	0.76	0.6
EfficientNetB3	0.63	0.13	0.52	0.71	0.8	0.54	0.83	0.86	0.86	0.78
EfficientNetB2	0.59	0.28	0.38	0.48	0.5	0.47	0.63	0.71	0.65	0.54
EfficientNetB1	0.51	0.09	0.47	0.6	0.48	0.5	0.68	0.73	0.7	0.54
EfficientNetB0	0.47	0	0.52	0.59	0.45	0.31	0.68	0.73	0.71	0.54
DenseNet201	0.36	-0.03	0.3	0.6	0.45	0.39	0.66	0.63	0.7	0.44
DenseNet169	0.37	0.06	0.43	0.54	0.35	0.44	0.69	0.64	0.63	0.49
DenseNet121	0.57	0.15	0.54	0.61	0.22	0.42	0.66	0.7	0.73	0.57

Table 11. Kappa Statistics values.

Classifiers/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	BG	MLP
Xception	0.53	0.08	0.39	0.5	0.48	0.4	0.65	0.68	0.78	0.51
VGG19	0.48	0.08	0.47	0.57	0.38	0.39	0.71	0.73	0.73	0.66
VGG16	0.38	0.03	0.42	0.49	0.52	0.25	0.71	0.7	0.64	0.57
ResNet152V2	0.27	0.09	0.24	0.54	0.46	0.23	0.6	0.64	0.58	0.46
ResNet152	0.57	0.18	0.5	0.55	0.35	0.43	0.74	0.69	0.68	0.58
ResNet101V2	0.43	0.05	0.3	0.46	0.3	0.22	0.6	0.64	0.67	0.44
ResNet101	0.6	0.15	0.56	0.63	0.3	0.46	0.73	0.73	0.75	0.54
ResNet50V2	0.31	0.09	0.3	0.47	0.39	0.32	0.58	0.54	0.65	0.52
ResNet50	0.54	0.21	0.5	0.55	0.37	0.49	0.75	0.7	0.7	0.62
NASNetMobile	0.38	0.19	0.39	0.49	0.2	0.32	0.62	0.63	0.64	0.46
NASNetLarge	0.53	0.08	0.39	0.48	0.17	0.55	0.69	0.64	0.67	0.54
MobileNet	0.39	0.01	0.28	0.38	0.34	0.49	0.65	0.69	0.68	0.51
InceptionV3	0.45	0.08	0.41	0.5	0.62	0.33	0.66	0.72	0.63	0.61
InceptionResNetV2	0.44	0.2	0.49	0.47	0.66	0.44	0.66	0.71	0.67	0.56
EfficientNetV2S	0.57	0.04	0.61	0.51	0.56	0.5	0.86	0.83	0.83	0.58
EfficientNetV2M	0.47	0.1	0.14	0.54	0.47	0.28	0.62	0.66	0.65	0.43
EfficientNetV2L	0.46	0.01	0.45	0.36	0.65	0.38	0.71	0.79	0.8	0.59
EfficientNetV2B3	0.49	0.13	0.55	0.7	0.64	0.58	0.76	0.84	0.83	0.76
EfficientNetV2B2	0.53	0.15	0.38	0.53	0.41	0.29	0.67	0.64	0.69	0.46
EfficientNetV2B1	0.55	0.03	0.38	0.59	0.33	0.36	0.64	0.68	0.71	0.49
EfficientNetV2B0	0.43	0.05	0.41	0.49	0.42	0.33	0.62	0.61	0.6	0.63
EfficientNetB7	0.73	0.14	0.42	0.59	0.65	0.51	0.78	0.77	0.73	0.58
EfficientNetB6	0.57	0.1	0.55	0.45	0.51	0.3	0.85	0.78	0.78	0.59
EfficientNetB5	0.42	0.19	0.52	0.67	0.56	0.36	0.76	0.81	0.72	0.59
EfficientNetB4	0.52	0.14	0.61	0.66	0.56	0.4	0.71	0.74	0.76	0.6
EfficientNetB3	0.62	0.1	0.52	0.7	0.8	0.48	0.83	0.86	0.86	0.78
EfficientNetB2	0.59	0.2	0.38	0.47	0.49	0.46	0.62	0.71	0.65	0.54
EfficientNetB1	0.51	0.07	0.47	0.58	0.45	0.5	0.68	0.73	0.69	0.54
EfficientNetB0	0.46	0	0.51	0.58	0.43	0.29	0.68	0.73	0.71	0.53
DenseNet201	0.36	-0.02	0.29	0.59	0.41	0.36	0.66	0.63	0.7	0.44
DenseNet169	0.36	0.05	0.43	0.54	0.32	0.44	0.69	0.64	0.63	0.48
DenseNet121	0.57	0.12	0.54	0.61	0.18	0.39	0.66	0.7	0.73	0.57

Discussion. From the case 1 analysis, it is observed that pre-trained models EfficientNetV2B3 and EfficientNetB3 gave better results with classifiers such as ET and HGB.

3.2. Analysis on Segmented Data

To enhance the performance of our model, we apply the segmentation technique to the same data set. Further, use the same approach to analyse the various parameters. We observed that after segmentation, our results were improved. Table 12 represents the analysis of proposed pre-trained models with machine learning and ensemble learning classifiers using accuracy. The most accurate model was found to be EfficientNetV2B3 with

HGB and ET, with 94% and 93% accuracy, respectively. Similarly, classifier RF and HFB gave an accuracy of 91% with mode EffiecientNetB3, respectively. The precision-based value analysis is shown in Table 13, which is the same as the accuracy model, EfficientNetV2B3, which achieved 93% accuracy with the ET and 92% precision with the HGB classifier. EfficientNetB3 with the HGB classifier achieved a 92% precision value.

Table 12. Accuracy on Segmented Data Set.

Classifiers/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.73	0.37	0.68	0.74	0.81	0.65	0.85	0.86	0.89	0.82
VGG19	0.66	0.48	0.76	0.76	0.68	0.69	0.76	0.77	0.81	0.72
VGG16	0.63	0.4	0.72	0.68	0.68	0.62	0.76	0.79	0.85	0.76
ResNet152V2	0.62	0.37	0.52	0.64	0.52	0.59	0.74	0.71	0.78	0.61
ResNet152	0.69	0.4	0.56	0.76	0.67	0.46	0.77	0.79	0.84	0.76
ResNet101V2	0.68	0.37	0.56	0.66	0.51	0.48	0.69	0.72	0.7	0.67
ResNet101	0.66	0.43	0.68	0.71	0.66	0.59	0.84	0.82	0.87	0.79
ResNet50V2	0.63	0.35	0.52	0.64	0.64	0.51	0.73	0.73	0.76	0.65
ResNet50	0.69	0.36	0.69	0.72	0.79	0.7	0.8	0.81	0.81	0.7
NASNetMobile	0.64	0.39	0.59	0.64	0.64	0.67	0.77	0.8	0.73	0.71
NASNetLarge	0.65	0.45	0.52	0.68	0.38	0.46	0.81	0.83	0.79	0.69
MobileNet	0.55	0.43	0.52	0.74	0.57	0.52	0.77	0.76	0.73	0.67
InceptionV3	0.78	0.45	0.7	0.74	0.82	0.69	0.86	0.87	0.85	0.82
InceptionResNetV2	0.77	0.38	0.66	0.73	0.79	0.68	0.83	0.82	0.87	0.8
EfficientNetV2S	0.72	0.38	0.77	0.74	0.72	0.71	0.84	0.85	0.89	0.82
EfficientNetV2M	0.66	0.48	0.6	0.76	0.78	0.56	0.8	0.85	0.85	0.78
EfficientNetV2L	0.74	0.38	0.64	0.77	0.67	0.62	0.8	0.84	0.87	0.71
EfficientNetV2B3	0.71	0.44	0.71	0.8	0.87	0.77	0.9	0.93	0.94	0.84
EfficientNetV2B2	0.71	0.4	0.64	0.77	0.63	0.59	0.8	0.83	0.85	0.77
EfficientNetV2B1	0.72	0.32	0.68	0.68	0.59	0.68	0.82	0.81	0.77	0.77
EfficientNetV2B0	0.67	0.46	0.66	0.74	0.65	0.53	0.79	0.79	0.79	0.7
EfficientNetB7	0.71	0.47	0.68	0.78	0.81	0.77	0.86	0.87	0.89	0.8
EfficientNetB6	0.82	0.47	0.69	0.76	0.74	0.71	0.86	0.87	0.9	0.81
EfficientNetB5	0.71	0.41	0.72	0.77	0.74	0.7	0.76	0.8	0.81	0.77
EfficientNetB4	0.79	0.45	0.74	0.79	0.72	0.67	0.87	0.9	0.83	0.74
EfficientNetB3	0.73	0.43	0.8	0.8	0.9	0.76	0.91	0.88	0.91	0.85
EfficientNetB2	0.69	0.49	0.68	0.72	0.73	0.71	0.84	0.82	0.85	0.76
EfficientNetB1	0.59	0.41	0.7	0.74	0.72	0.68	0.83	0.84	0.84	0.73
EfficientNetB0	0.79	0.37	0.67	0.71	0.74	0.61	0.82	0.84	0.77	0.74
DenseNet201	0.68	0.46	0.66	0.78	0.72	0.57	0.83	0.83	0.82	0.73
DenseNet169	0.71	0.43	0.54	0.74	0.68	0.67	0.81	0.84	0.82	0.7
DenseNet121	0.71	0.45	0.67	0.74	0.71	0.67	0.76	0.78	0.85	0.77

Table 13. Precision value on Segmented Data Set.

Classifier/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.66	0.23	0.58	0.72	0.75	0.45	0.84	0.85	0.87	0.78
VGG19	0.63	0.29	0.71	0.72	0.65	0.63	0.72	0.73	0.77	0.7
VGG16	0.61	0.23	0.68	0.65	0.63	0.46	0.72	0.75	0.82	0.7
ResNet152V2	0.55	0.2	0.4	0.61	0.59	0.56	0.68	0.61	0.76	0.54
ResNet152	0.68	0.26	0.5	0.67	0.54	0.42	0.68	0.69	0.78	0.69
ResNet101V2	0.66	0.23	0.53	0.67	0.62	0.45	0.6	0.63	0.64	0.63
ResNet101	0.59	0.24	0.62	0.67	0.65	0.52	0.82	0.78	0.84	0.77
ResNet50V2	0.6	0.22	0.46	0.61	0.59	0.35	0.66	0.65	0.71	0.59
ResNet50	0.65	0.27	0.63	0.67	0.78	0.7	0.76	0.78	0.76	0.64
NASNetMobile	0.57	0.25	0.54	0.67	0.59	0.6	0.7	0.74	0.7	0.66
NASNetLarge	0.6	0.27	0.56	0.63	0.58	0.37	0.77	0.81	0.73	0.73
MobileNet	0.51	0.15	0.39	0.71	0.54	0.48	0.7	0.7	0.65	0.56
InceptionV3	0.72	0.27	0.65	0.73	0.76	0.47	0.81	0.82	0.81	0.76
InceptionResNetV2	0.7	0.24	0.62	0.71	0.76	0.63	0.78	0.79	0.85	0.74
EfficientNetV2S	0.7	0.23	0.73	0.69	0.69	0.68	0.81	0.81	0.86	0.78
EfficientNetV2M	0.62	0.29	0.54	0.71	0.74	0.51	0.76	0.8	0.8	0.74
EfficientNetV2L	0.69	0.25	0.59	0.72	0.63	0.66	0.74	0.78	0.81	0.66
EfficientNetV2B3	0.68	0.29	0.65	0.8	0.87	0.84	0.88	0.93	0.92	0.81
EfficientNetV2B2	0.68	0.24	0.61	0.73	0.65	0.54	0.76	0.81	0.81	0.73
EfficientNetV2B1	0.67	0.27	0.65	0.68	0.65	0.67	0.76	0.75	0.69	0.73
EfficientNetV2B0	0.63	0.26	0.6	0.71	0.62	0.66	0.71	0.71	0.72	0.64
EfficientNetB7	0.66	0.3	0.63	0.75	0.74	0.67	0.84	0.84	0.89	0.72
EfficientNetB6	0.78	0.27	0.64	0.69	0.67	0.48	0.83	0.85	0.88	0.77
EfficientNetB5	0.66	0.26	0.66	0.72	0.71	0.73	0.69	0.77	0.76	0.7
EfficientNetB4	0.75	0.28	0.7	0.76	0.65	0.45	0.84	0.88	0.79	0.69
EfficientNetB3	0.7	0.26	0.76	0.77	0.93	0.85	0.9	0.88	0.92	0.82
EfficientNetB2	0.66	0.29	0.65	0.68	0.7	0.7	0.79	0.78	0.83	0.72
EfficientNetB1	0.55	0.24	0.66	0.69	0.7	0.65	0.77	0.8	0.8	0.68
EfficientNetB0	0.73	0.22	0.62	0.64	0.69	0.49	0.77	0.79	0.7	0.68
DenseNet201	0.62	0.27	0.63	0.73	0.68	0.5	0.79	0.78	0.76	0.7
DenseNet169	0.66	0.27	0.48	0.7	0.62	0.64	0.77	0.8	0.81	0.63
DenseNet121	0.7	0.24	0.59	0.69	0.51	0.63	0.69	0.71	0.81	0.71

Next, an analysis on the basis of recall rate and F1-Score is represented in Tables 14 and 15. It is noted that, as measured by precision, EfficientNetV2B3 with the HGB classifier achieved a 92% recall rate and an F1-Score.

Table 14. Recall Rate on Segmented Data Set.

Classifier/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.67	0.38	0.58	0.73	0.73	0.51	0.79	0.79	0.87	0.78
VGG19	0.65	0.49	0.71	0.75	0.65	0.6	0.74	0.75	0.81	0.73
VGG16	0.61	0.38	0.63	0.65	0.64	0.48	0.74	0.76	0.84	0.68
ResNet152V2	0.55	0.33	0.42	0.62	0.59	0.48	0.66	0.61	0.71	0.54
ResNet152	0.7	0.44	0.5	0.67	0.55	0.4	0.67	0.68	0.78	0.65
ResNet101V2	0.67	0.38	0.52	0.68	0.57	0.43	0.6	0.63	0.64	0.63
ResNet101	0.59	0.41	0.61	0.67	0.61	0.5	0.8	0.78	0.84	0.76
ResNet50V2	0.58	0.38	0.44	0.6	0.59	0.4	0.66	0.65	0.71	0.58
ResNet50	0.64	0.46	0.63	0.69	0.71	0.58	0.76	0.78	0.76	0.64
NASNetMobile	0.57	0.43	0.53	0.64	0.58	0.59	0.7	0.74	0.71	0.67
NASNetLarge	0.61	0.4	0.5	0.64	0.47	0.35	0.78	0.78	0.73	0.6
MobileNet	0.5	0.33	0.42	0.73	0.49	0.44	0.68	0.68	0.65	0.56
InceptionV3	0.73	0.45	0.64	0.74	0.76	0.55	0.79	0.82	0.84	0.76
InceptionResNetV2	0.7	0.41	0.62	0.73	0.72	0.55	0.78	0.75	0.83	0.73
EfficientNetV2S	0.73	0.39	0.73	0.7	0.7	0.62	0.82	0.81	0.86	0.8
EfficientNetV2M	0.62	0.51	0.54	0.7	0.75	0.48	0.77	0.81	0.81	0.74
EfficientNetV2L	0.69	0.43	0.59	0.73	0.63	0.54	0.74	0.78	0.79	0.66
EfficientNetV2B3	0.7	0.49	0.66	0.83	0.79	0.62	0.87	0.89	0.92	0.77
EfficientNetV2B2	0.7	0.41	0.59	0.74	0.65	0.53	0.76	0.78	0.81	0.7
EfficientNetV2B1	0.68	0.44	0.65	0.66	0.6	0.67	0.75	0.75	0.69	0.74
EfficientNetV2B0	0.63	0.43	0.59	0.73	0.61	0.51	0.71	0.69	0.72	0.65
EfficientNetB7	0.66	0.51	0.63	0.78	0.72	0.65	0.78	0.82	0.81	0.73
EfficientNetB6	0.79	0.46	0.64	0.7	0.66	0.56	0.83	0.86	0.9	0.75
EfficientNetB5	0.66	0.43	0.66	0.73	0.71	0.58	0.69	0.74	0.77	0.7
EfficientNetB4	0.76	0.48	0.69	0.79	0.64	0.53	0.84	0.88	0.8	0.69
EfficientNetB3	0.72	0.44	0.76	0.78	0.83	0.61	0.89	0.87	0.89	0.79
EfficientNetB2	0.67	0.49	0.66	0.68	0.72	0.66	0.8	0.79	0.8	0.71
EfficientNetB1	0.53	0.4	0.64	0.69	0.72	0.61	0.77	0.78	0.82	0.68
EfficientNetB0	0.74	0.37	0.61	0.63	0.7	0.49	0.77	0.8	0.7	0.67
DenseNet201	0.62	0.46	0.61	0.74	0.7	0.48	0.8	0.79	0.76	0.72
DenseNet169	0.66	0.44	0.48	0.71	0.61	0.62	0.74	0.76	0.75	0.63
DenseNet121	0.66	0.4	0.59	0.7	0.57	0.6	0.69	0.71	0.81	0.72

Table 15. F1-Score on Segmented Data Set.

Classifier/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.66	0.28	0.58	0.7	0.73	0.47	0.8	0.81	0.87	0.77
VGG19	0.63	0.37	0.71	0.72	0.64	0.6	0.72	0.73	0.78	0.7
VGG16	0.59	0.28	0.63	0.64	0.63	0.44	0.72	0.74	0.82	0.69
ResNet152V2	0.54	0.24	0.4	0.6	0.52	0.46	0.67	0.6	0.73	0.54
ResNet152	0.67	0.31	0.5	0.67	0.53	0.38	0.68	0.68	0.78	0.66
ResNet101V2	0.65	0.28	0.52	0.64	0.5	0.42	0.6	0.63	0.64	0.62
ResNet101	0.59	0.3	0.61	0.66	0.6	0.5	0.81	0.78	0.84	0.76
ResNet50V2	0.58	0.27	0.44	0.6	0.59	0.36	0.66	0.65	0.71	0.58
ResNet50	0.64	0.3	0.63	0.67	0.72	0.57	0.76	0.78	0.76	0.64
NASNetMobile	0.57	0.31	0.53	0.62	0.57	0.59	0.7	0.74	0.69	0.66
NASNetLarge	0.6	0.3	0.52	0.63	0.39	0.26	0.77	0.79	0.73	0.61
MobileNet	0.49	0.2	0.4	0.71	0.47	0.42	0.68	0.69	0.65	0.55
InceptionV3	0.72	0.34	0.64	0.71	0.76	0.5	0.8	0.82	0.82	0.76
InceptionResNetV2	0.7	0.29	0.61	0.7	0.73	0.52	0.78	0.76	0.84	0.73
EfficientNetV2S	0.7	0.29	0.73	0.69	0.69	0.62	0.81	0.81	0.86	0.79
EfficientNetV2M	0.61	0.37	0.54	0.7	0.75	0.48	0.76	0.81	0.81	0.74
EfficientNetV2L	0.68	0.3	0.59	0.72	0.62	0.54	0.74	0.78	0.8	0.66
EfficientNetV2B3	0.68	0.35	0.66	0.78	0.81	0.59	0.87	0.9	0.92	0.79
EfficientNetV2B2	0.68	0.3	0.6	0.73	0.6	0.53	0.76	0.79	0.81	0.71
EfficientNetV2B1	0.67	0.27	0.65	0.65	0.57	0.64	0.75	0.75	0.69	0.73
EfficientNetV2B0	0.63	0.33	0.59	0.71	0.6	0.42	0.71	0.7	0.72	0.64
EfficientNetB7	0.66	0.37	0.63	0.75	0.73	0.65	0.8	0.83	0.82	0.73
EfficientNetB6	0.78	0.34	0.64	0.69	0.66	0.52	0.83	0.85	0.89	0.76
EfficientNetB5	0.66	0.32	0.66	0.72	0.7	0.58	0.69	0.75	0.76	0.7
EfficientNetB4	0.75	0.35	0.69	0.76	0.64	0.48	0.84	0.88	0.79	0.69
EfficientNetB3	0.7	0.32	0.76	0.76	0.85	0.58	0.9	0.87	0.9	0.8
EfficientNetB2	0.65	0.36	0.65	0.67	0.7	0.67	0.8	0.78	0.81	0.71
EfficientNetB1	0.53	0.3	0.65	0.68	0.69	0.61	0.77	0.79	0.81	0.68
EfficientNetB0	0.73	0.27	0.61	0.63	0.69	0.46	0.77	0.8	0.7	0.67
DenseNet201	0.62	0.34	0.6	0.73	0.69	0.48	0.79	0.79	0.76	0.7
DenseNet169	0.66	0.32	0.48	0.7	0.61	0.62	0.75	0.78	0.76	0.62
DenseNet121	0.67	0.3	0.59	0.69	0.53	0.6	0.69	0.71	0.81	0.71

Analysis on the basis of Matthews Coefficient and Kappa Statistics is represented in Tables 16 and 17, and it is noted that the maximum values were 90% for both Matthews Coefficient and Kappa Statistics with EfficientNetV2B3 for HGB classifiers.

Table 16. Matthews Coefficient on Segmented Data Set.

Classifier/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.57	0.08	0.48	0.62	0.69	0.43	0.76	0.78	0.83	0.72
VGG19	0.48	0.24	0.62	0.63	0.52	0.5	0.63	0.64	0.71	0.58
VGG16	0.44	0.1	0.54	0.51	0.49	0.4	0.61	0.67	0.77	0.6
ResNet152V2	0.39	-0.02	0.23	0.45	0.34	0.3	0.58	0.53	0.64	0.37
ResNet152	0.54	0.17	0.29	0.6	0.47	0.1	0.62	0.65	0.74	0.6
ResNet101V2	0.52	0.08	0.3	0.52	0.36	0.15	0.5	0.55	0.52	0.49
ResNet101	0.45	0.15	0.48	0.56	0.49	0.32	0.74	0.71	0.79	0.66
ResNet50V2	0.44	0.06	0.24	0.45	0.42	0.17	0.57	0.57	0.61	0.44
ResNet50	0.53	0.18	0.5	0.57	0.66	0.51	0.67	0.69	0.69	0.52
NASNetMobile	0.43	0.15	0.34	0.5	0.44	0.46	0.63	0.68	0.59	0.56
NASNetLarge	0.45	0.12	0.22	0.5	0.25	0.09	0.7	0.72	0.66	0.5
MobileNet	0.31	-0.01	0.22	0.61	0.33	0.21	0.62	0.6	0.58	0.46
InceptionV3	0.65	0.17	0.54	0.63	0.71	0.49	0.78	0.79	0.77	0.71
InceptionResNetV2	0.63	0.11	0.47	0.6	0.66	0.48	0.73	0.71	0.8	0.67
EfficientNetV2S	0.58	0.1	0.62	0.6	0.57	0.53	0.74	0.76	0.83	0.71
EfficientNetV2M	0.48	0.25	0.35	0.6	0.65	0.29	0.68	0.76	0.76	0.64
EfficientNetV2L	0.6	0.15	0.42	0.64	0.48	0.4	0.68	0.74	0.79	0.54
EfficientNetV2B3	0.56	0.25	0.55	0.72	0.8	0.62	0.85	0.88	0.9	0.74
EfficientNetV2B2	0.56	0.11	0.41	0.64	0.48	0.33	0.67	0.72	0.76	0.62
EfficientNetV2B1	0.57	0.16	0.49	0.54	0.44	0.52	0.71	0.69	0.62	0.63
EfficientNetV2B0	0.47	0.18	0.45	0.62	0.47	0.32	0.65	0.65	0.66	0.53
EfficientNetB7	0.55	0.29	0.5	0.67	0.69	0.62	0.78	0.79	0.83	0.68
EfficientNetB6	0.71	0.21	0.51	0.61	0.58	0.53	0.78	0.8	0.85	0.69
EfficientNetB5	0.55	0.18	0.55	0.63	0.6	0.53	0.61	0.67	0.69	0.62
EfficientNetB4	0.67	0.19	0.6	0.69	0.55	0.45	0.8	0.85	0.73	0.59
EfficientNetB3	0.58	0.17	0.67	0.7	0.85	0.61	0.86	0.81	0.86	0.76
EfficientNetB2	0.52	0.26	0.5	0.57	0.59	0.53	0.75	0.71	0.76	0.61
EfficientNetB1	0.37	0.12	0.51	0.6	0.59	0.48	0.73	0.74	0.75	0.57
EfficientNetB0	0.67	0.06	0.47	0.55	0.59	0.35	0.71	0.75	0.62	0.59
DenseNet201	0.49	0.2	0.48	0.65	0.56	0.32	0.73	0.73	0.71	0.59
DenseNet169	0.54	0.2	0.28	0.6	0.5	0.47	0.69	0.74	0.71	0.53
DenseNet121	0.54	0.16	0.47	0.6	0.53	0.46	0.61	0.64	0.76	0.63

Table 17. Kappa Statistics on Segmented Data Set.

Classifier/ Pre-Trained Model	DT	QDA	KNN	AB	GNB	LR	RF	ET	HGB	MLP
Xception	0.57	0.06	0.47	0.61	0.69	0.39	0.75	0.77	0.83	0.71
VGG19	0.47	0.19	0.61	0.62	0.51	0.49	0.62	0.64	0.7	0.57
VGG16	0.43	0.08	0.54	0.51	0.49	0.34	0.61	0.67	0.76	0.6
ResNet152V2	0.39	-0.02	0.22	0.45	0.31	0.29	0.58	0.53	0.63	0.37
ResNet152	0.53	0.13	0.29	0.6	0.46	0.09	0.62	0.65	0.74	0.59
ResNet101V2	0.51	0.06	0.3	0.5	0.31	0.14	0.5	0.55	0.52	0.48
ResNet101	0.45	0.12	0.48	0.55	0.46	0.31	0.74	0.71	0.79	0.65
ResNet50V2	0.43	0.05	0.24	0.44	0.42	0.16	0.57	0.57	0.61	0.44
ResNet50	0.52	0.13	0.5	0.57	0.65	0.49	0.67	0.69	0.69	0.52
NASNetMobile	0.43	0.12	0.33	0.47	0.42	0.46	0.63	0.68	0.59	0.55
NASNetLarge	0.45	0.08	0.21	0.5	0.18	0.04	0.69	0.72	0.66	0.48
MobileNet	0.3	0	0.22	0.6	0.29	0.19	0.61	0.6	0.58	0.45
InceptionV3	0.65	0.13	0.53	0.61	0.71	0.47	0.78	0.79	0.77	0.71
InceptionResNetV2	0.63	0.09	0.46	0.59	0.65	0.45	0.73	0.7	0.79	0.67
EfficientNetV2S	0.57	0.08	0.62	0.6	0.56	0.52	0.74	0.76	0.83	0.71
EfficientNetV2M	0.48	0.2	0.35	0.6	0.64	0.28	0.68	0.76	0.76	0.64
EfficientNetV2L	0.59	0.12	0.42	0.63	0.48	0.35	0.68	0.74	0.79	0.54
EfficientNetV2B3	0.56	0.2	0.55	0.7	0.79	0.6	0.85	0.88	0.9	0.74
EfficientNetV2B2	0.55	0.09	0.41	0.64	0.45	0.32	0.67	0.72	0.76	0.62
EfficientNetV2B1	0.57	0.11	0.49	0.52	0.4	0.5	0.71	0.69	0.62	0.63
EfficientNetV2B0	0.47	0.14	0.45	0.61	0.46	0.23	0.65	0.65	0.66	0.52
EfficientNetB7	0.55	0.23	0.5	0.66	0.69	0.61	0.77	0.79	0.82	0.68
EfficientNetB6	0.71	0.17	0.5	0.61	0.58	0.5	0.78	0.8	0.85	0.69
EfficientNetB5	0.55	0.15	0.55	0.63	0.59	0.49	0.61	0.67	0.69	0.62
EfficientNetB4	0.67	0.16	0.59	0.67	0.55	0.43	0.8	0.85	0.73	0.59
EfficientNetB3	0.58	0.13	0.67	0.69	0.84	0.58	0.86	0.81	0.86	0.76
EfficientNetB2	0.52	0.2	0.5	0.57	0.59	0.53	0.74	0.71	0.76	0.6
EfficientNetB1	0.36	0.09	0.51	0.6	0.58	0.47	0.73	0.74	0.75	0.57
EfficientNetB0	0.66	0.05	0.47	0.54	0.59	0.33	0.71	0.74	0.62	0.58
DenseNet201	0.49	0.16	0.47	0.65	0.56	0.31	0.73	0.73	0.71	0.59
DenseNet169	0.54	0.16	0.27	0.6	0.5	0.46	0.69	0.74	0.7	0.52
DenseNet121	0.53	0.12	0.47	0.6	0.52	0.45	0.61	0.64	0.76	0.63

From the case 2 analysis, it is observed that pre-trained model EfficientNetV2B3, EfficientNetB3, and EfficientNetB4 gave better results with classifiers such as ET and HGB.

3.3. Comparative Discussion

In comparison to all mentioned algorithms, maximum accuracy resulted from the approaches EfficientNetV2B3, EfficientNetB3, EfficientNetV2S, and EfficientNetB6 with classifiers such as RF, ET and HGB classifier, and it was near 91 percent. However, by implementing comparative analysis with the segmented dataset, the highest accuracy was 94 and 93 percent with the EfficientNetV2B3 model with HGB, the ET classifier. Similarly, the other efficiency parameters such as precision, recall rate, F1-Score, and Matthews and Kappa Coefficients achieved their highest values with the same models and classifiers discussed in Table 18.

Table 18. Comparative analysis on normal and segmented data.

Data	Pre-Trained Model	Classifier	Accuracy	Precision	Recall	F1-Score	Matthew Coefficient	Kappa Statistics
Normal Data	EfficientNetB3	HGB	0.91	0.90	0.89	0.90	0.86	0.86
	EfficientNetV2B3	HGB	0.89	0.92	0.86	0.89	0.83	0.83
Segmented Data	EfficientNetV2B3	HGB	0.94	0.92	0.92	0.92	0.90	0.90
	EfficientNetV2B3	ET	0.93	0.93	0.89	0.90	0.88	0.88

4. Conclusions

Rice leaf diseases have a devastating effect on global food security and are the primary threats to agricultural progress around the world. There may be no harvest at all if the leaf disease is severe [41]. To ensure the productivity of rice products, the prompt and precise identification of rice leaf diseases is essential. For this reason, it is very important to look for quick, less expensive, and accurate ways to identify rice leaf disease cases. In solving the majority of the technological issues related to the classification of leaf diseases, pre-trained transfer learning algorithms have demonstrated excellent performance. In this study, we proposed an analysis of various pre-trained models with different classifiers for the detection of rice leaf diseases. The three major rice leaf diseases, BB, BS, and blast, are considered for this research. Image-based rice leaf disease data set was collected and pre-processed according to algorithmic requirements. Initially, 32 pre-trained models were used to extract features, and then the images were classified using various machine and ensemble learning classifiers. Images are enhanced by the segmentation process, and the results are compared on various performance parameters such as accuracy, precision, recall rate, F1-Score, Matthews Coefficient, and Kappa Statistics. Experiments were performed on both the normal image data set and the segmented image data set. With the pre-trained models EfficientNetB3, EfficientNetB6, EfficientNetV2S, and EfficientNetV2B3 with an Extra Tree and HGB classifier, the proposed model achieves 91% accuracy on a normal data set and 94% accuracy on a segmented data set. In the future, we will deploy these results with mobile devices to recognise the rice leaf disease automatically, and also this model could be used to classify other related crops in agriculture.

Author Contributions: Study conception and design: M.A., V.K., N.G.; data collection: M.A., V.K., N.G.; methodology: M.A., V.K., N.G., A.T., A.S., E.B.T., S.K.; analysis and interpretation of results: M.A., V.K., N.G.; draft manuscript preparation: M.A., V.K., N.G, A.T., A.S., E.B.T., S.K.; Supervision, V.K., N.G, A.T., A.S.; Project Administration, V.K., N.G, A.T.; Funding Acquisition, E.B.T.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Researchers Supporting Project Number (RSPD2023R681), King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was funded by the Researchers Supporting Project Number (RSPD2023R681), King Saud University, Riyadh, Saudi Arabia. Authors are grateful to King Saud University, Riyadh, Saudi Arabia for carry out research work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviation

Abbreviation	Definition	Abbreviation	Definition
ML	Machine Learning	GNB	Gaussian Naïve Bayes
DL	Deep Learning	K-NN	K-Nearest Neighbour
DCNN	Deep CNN	LR	Logistic Regression
FS	False Smut	SVM	Support Vector Machine
BS	Brown Spot	DT	Decision Tree
SB	Sheath Blight	RF	Random factor
SB	Stem Borer	QDA	Quadratic Discriminant Analysis
LS	Leaf Smut	AB	Ada-boost
SR	Sheath Rot	ET	Extra Tree
FS	False Smut	HGB	Histogram Gradient boosting
BB	Bacterial Blight	GB	Gradient Boosting
MLP	Multi-LayerPreceptron	FN	False Negative
TP	True Positive	MC	Matthews Coefficient
TN	True Negative	KP	Kappa Statistics
FP	False Positive	YOLO	You only look once

References

1. Stanley, M. Food Staple. Available online: https://education.nationalgeographic.org/resource/food-staple (accessed on 26 December 2022).
2. Kodama, T.; Hata, Y. Development of Classification System of Rice Disease Using Artificial Intelligence. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; pp. 3699–3702. [CrossRef]
3. Chen, J.; Zhang, D.; Zeb, A.; Nanehkaran, Y.A. Identification of Rice Plant Diseases Using Lightweight Attention Networks. *Expert Syst. Appl.* **2021**, *169*, 114514. [CrossRef]
4. Wani, J.A.; Sharma, S.; Muzamil, M.; Ahmed, S.; Sharma, S.; Singh, S. Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges. In *Archives of Computational Methods in Engineering*; Springer: Amsterdam, The Netherlands, 2022; Volume 29, pp. 641–677. ISBN 0123456789.
5. Krishnamoorthy, N.; Prasad, L.N.; Kumar, C.P.; Subedi, B.; Abraha, H.B.; Sathishkumar, V.E. Rice Leaf Diseases Prediction Using Deep Neural Networks with Transfer Learning. *Environ. Res.* **2021**, *198*, 111275. [CrossRef]
6. Prajapati, H.B.; Shah, J.P.; Dabhi, V.K. Detection and Classification of Rice Plant Diseases. *Intell. Decis. Technol.* **2017**, *11*, 357–373. [CrossRef]
7. Gunawan, P.A.; Kencana, E.N.; Sari, K. Classification of Rice Leaf Diseases Using Artificial Neural Network. *J. Phys. Conf. Ser.* **2021**, *1722*, 012013. [CrossRef]
8. Narmadha, R.P.; Arulvadivu, G. Detection and Measurement of Paddy Leaf Disease Symptoms Using Image Processing. In Proceedings of the 2017 International Conference on Computer Communication and Informatics (ICCCI), Tamilnadu, India, 5–7 January 2017; pp. 5–8. [CrossRef]
9. Liu, L.W.; Hsieh, S.H.; Lin, S.J.; Wang, Y.M.; Lin, W.S. Rice Blast (Magnaporthe Oryzae) Occurrence Prediction and the Key Factor Sensitivity Analysis by Machine Learning. *Agronomy* **2021**, *11*, 771. [CrossRef]
10. Sharma, T.; Kaur, P.; Chahal, J.; Sharma, H. Classification of Rice Leaf Diseases Based on the Deep Convolutional Neural Network Architectures: Review. In Proceedings of the AIP Conference Proceedings, Himachal Pradesh, India, 2–3 July 2022; Volume 2451.
11. Maniyath, S.R.; Vinod, P.V.; Niveditha, M.; Pooja, R.; Prasad Bhat, N.; Shashank, N.; Hebbar, R. Plant Disease Detection Using Machine Learning. In Proceedings of the 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India, 25–28 April 2018; pp. 41–45. [CrossRef]
12. Bari, B.S.; Islam, M.N.; Rashid, M.; Hasan, M.J.; Razman, M.A.M.; Musa, R.M.; Nasir, A.F.A.; Majeed, A.P.P.A. A Real-Time Approach of Diagnosing Rice Leaf Disease Using Deep Learning-Based Faster R-CNN Framework. *PeerJ Comput. Sci.* **2021**, *7*, e432. [CrossRef] [PubMed]
13. Daniya, T.; Vigneshwari, S. Deep Neural Network for Disease Detection in Rice Plant Using the Texture and Deep Features. *Comput. J.* **2022**, *65*, 1812–1825. [CrossRef]
14. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Deep Feature Based Rice Leaf Disease Identification Using Support Vector Machine. *Comput. Electron. Agric.* **2020**, *175*, 105527. [CrossRef]
15. Rautaray, S.S.; Pandey, M.; Gourisaria, M.K.; Sharma, R.; Das, S. Paddy Crop Disease Prediction—A Transfer Learning Technique. *Int. J. Recent Technol. Eng.* **2020**, *8*, 1490–1495. [CrossRef]
16. Jadhav, A.A.J.B.D. Monitoring and Controlling Rice Diseases Using Image Processing Techniques. In Proceedings of the International Conference on Computing, Analytics and Security Trends (CAST), Pune, India, 19–21 December 2016; pp. 471–476.

17. Sharma, V.; Mir, A.A.; Sarwr, D.A. Detection of Rice Disease Using Bayes' Classifier and Minimum Distance Classifier. *J. Multimed. Inf. Syst.* **2020**, *7*, 17–24. [CrossRef]
18. Lu, Y.; Yi, S.; Zeng, N.; Liu, Y.; Zhang, Y. Identification of Rice Diseases Using Deep Convolutional Neural Networks. *Neurocomputing* **2017**, *267*, 378–384. [CrossRef]
19. Jiang, F.; Lu, Y.; Chen, Y.; Cai, D.; Li, G. Image Recognition of Four Rice Leaf Diseases Based on Deep Learning and Support Vector Machine. *Comput. Electron. Agric.* **2020**, *179*, 105824. [CrossRef]
20. He, Y.; Zhou, Z.; Tian, L.; Liu, Y.; Luo, X. Brown Rice Planthopper (Nilaparvata Lugens Stal) Detection Based on Deep Learning. *Precis. Agric.* **2020**, *21*, 1385–1402. [CrossRef]
21. Azim, M.A.; Islam, M.K.; Rahman, M.M.; Jahan, F. An Effective Feature Extraction Method for Rice Leaf Disease Classification. *Telkomnika (Telecommun. Comput. Electron. Control)* **2021**, *19*, 463–470. [CrossRef]
22. Suresha, M.; Shreekanth, K.N.; Thirumalesh, B.V. Recognition of Diseases in Paddy Leaves Using Knn Classifier. In Proceedings of the 2nd International Conference for Convergence in Technology, I2CT, Mumbai, India, 7–9 April 2017; pp. 663–666.
23. Sardogan, M.; Tuncer, A.; Ozen, Y. Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm. In Proceedings of the UBMK—3rd International Conference on Computer Science and Engineering, Sarajevo, Bosnia and Herzegovina, 20–23 September 2018; pp. 382–385.
24. Liang, W.-J.; Zhang, H.; Zhang, G.F.; Cao, H.-X. Rice Blast Disease Recognition Using a Deep Convolutional Neural Network. *Sci. Rep.* **2019**, *9*, 2869. [CrossRef] [PubMed]
25. Nidhis, A.D.; Pardhu, C.N.V.; Reddy, K.C.; Deepa, K. Cluster Based Paddy Leaf Disease Detection, Classification and Diagnosis in Crop Health Monitoring Unit. In *Lecture Notes in Computational Vision and Biomechanics*; Springer International Publishing: New York, NY, USA, 2019; Volume 31, pp. 281–291. ISBN 9783030040611.
26. Anami, B.S.; Malvade, N.N.; Palaiah, S. Deep Learning Approach for Recognition and Classification of Yield Affecting Paddy Crop Stresses Using Field Images. *Artif. Intell. Agric.* **2020**, *4*, 12–20. [CrossRef]
27. Ghosal, S.; Sarkar, K. Rice Leaf Diseases Classification Using CNN with Transfer Learning. In Proceedings of the IEEE Calcutta Conference, CALCON–Proceedings, Kolkata, India, 28–29 February 2020; pp. 230–236.
28. Li, D.; Wang, R.; Xie, C.; Liu, L.; Zhang, J.; Li, R.; Wang, F.; Zhou, M.; Liu, W. A Recognition Method for Rice Plant Diseases and Pests Video Detection Based on Deep Convolutional Neural Network. *Sensors* **2020**, *20*, 578. [CrossRef] [PubMed]
29. Matin, M.M.H.; Khatun, A.; Moazzam, M.G.; Uddin, M.S. An Efficient Disease Detection Technique of Rice Leaf Using AlexNet. *J. Comput. Commun.* **2020**, *8*, 49–57. [CrossRef]
30. Temniranrat, P.; Kiratiratanapruk, K.; Kitvimonrat, A.; Sinthupinyo, W.; Patarapuwadol, S. A System for Automatic Rice Disease Detection from Rice Paddy Images Serviced via a Chatbot. *Comput. Electron. Agric.* **2021**, *185*, 106156. [CrossRef]
31. Francisco, A.K.G. Rice Disease Data Set. Available online: https://github.com/aldrin233/RiceDiseases-DataSet (accessed on 12 October 2022).
32. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Image Processing Techniques for Diagnosing Rice Plant Disease: A Survey. *Procedia Comput. Sci.* **2020**, *167*, 516–530. [CrossRef]
33. Krichen, M.; Mihoub, A.; Alzahrani, M.Y.; Adoni, W.Y.H.; Nahhal, T. Are Formal Methods Applicable To Machine Learning And Artificial Intelligence? In Proceedings of the International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, 9–11 May 2022; pp. 48–53.
34. Seshia, S.A.; Sadigh, D.; Sastry, S.S. Toward Verified Artificial Intelligence. *Commun. ACM* **2022**, *65*, 46–55. [CrossRef]
35. Anantrasirichai, N.; Hannuna, S.; Canagarajah, N. Automatic Leaf Extraction from Outdoor Images. *arXiv* **2017**, arXiv:1709.06437. [CrossRef]
36. Watershed Alogithm and Application. Available online: https://www.aegissofttech.com/articles/watershed-algorithm-and-limitations.html (accessed on 25 December 2022).
37. Dara, S.; Tumma, P. Feature Extraction by Using Deep Learning: A Survey. In Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018, Coimbatore, India, 29–31 March 2018; pp. 1795–1801.
38. Rallapalli, S.M.; Saleem Durai, M.A. A Contemporary Approach for Disease Identification in Rice Leaf. *Int. J. Syst. Assur. Eng. Manag.* **2021**, 1–11. [CrossRef]
39. Sengupta, S.; Dutta, A.; Abdelmohsen, S.A.M.; Alyousef, H.A.; Rahimi-Gorji, M. Development of a Rice Plant Disease Classification Model in Big Data Environment. *Bioengineering* **2022**, *9*, 758. [CrossRef] [PubMed]
40. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using Deep Transfer Learning for Image-Based Plant Disease Identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [CrossRef]
41. Kaur, A.; Guleria, K.; Trivedi, N. A Deep Learning Based Model for Rice Leaf Disease Detection. In Proceedings of the 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 13–14 October 2022; pp. 1–5.