Journal of Advanced Zoology
ISSN: 0253-7214
Volume 45 Issue 3 Year 2024 Page 316-335

A Deep Learning Approach For Automated Rice Disease Detection And Classification

Deepika Mandwariya[1*], Dr. Varsha Jotwani[2]

[1*]*Ph.D Scholar, Department Of Computer Science, RNTU Bhopal*
[2]*Professor And HOD, Department Of Computer Science & IT, RNTU Bhopal*

***Corresponding Author:** Deepika Mandwariya*
**Ph.D Scholar, Department Of Computer Science, RNTU Bhopal*

Abstract:

Rice is considered one the most important plants globally because it is a source of food for over half the world's population. Like other plants, rice is susceptible to diseases that may affect the quantity and quality of produce. Because of this, crops sometimes lose 20% to 40% of their value. A good harvest can depend on early discovery of these diseases, so farmers would need to be able to read and understand images of the diseases. Also, farmers still can't reach their goal of doing daily studies of their huge farmlands. It would be very expensive to do this, so the price of rice for buyers would go up even if it were possible. This paper proposed a pre-trained Deep Convolutional Neural Network (DCNN) method based on optimization for accurately finding and classifying rice leaf disease. It uses both transfer learning and baseline learning. A precise diagnosis method can find and classify eleven different types of rice diseased healthy, leaf blast, brown spot, bacterial blight and bacterial leaf blight, false stump, neck blast, stemborer, tumgro, hispa, and BPH. The most advanced large-scale architecture, such as XceptionNet, ResNet50, DenseNet VGG19, SequeezeNet and CNN used for recognition of the Rice disease, with SGDM, ADAM, RMS propagation optimization methods for predictions for a dataset. The -proposed models were trained and tested using datasets gathered from websites. In the simulation results consistently demonstrate that the XceptionNet model outperforms other architectures in terms of higher accuracy 93.3 %.

Keywords: Machine learning, deep learning, transfer learning, plant leaf disease detection, rice leaf disease detection, and VGG19 are some of the terms that are utilized over the course of this research.

I INTRODUCTION

One of the mainly important food crops in the world is rice, which gives people energy and is a mainstay of the diets of more than half of the world's population. This means that a large part of the world's population depends on it to survive, and it is an main part of feeding a great part of the population [1]. The need for food keeps going up because the world's population keeps growing. As a result, it is becoming more and more important to make sure that the rice production is regular and produces enough rice. A huge part of the world's population, especially in Asia, where it is a basic food, depends on rice as their main source of food. Countries all over the world grow and eat rice. Also, more than half of the world's people, who live in Asia, get most of their calories from rice. There are a lot of carbs, proteins, vitamins, and minerals in rice. So, to make sure there is enough food for everyone, it is important to make sure there is a steady harvest of rice. In other words, the rice field needs to be kept safe from things that could hurt it, like diseases, pests, and other things that could cause it to fail. Increased crop output can be achieved by using the right farming methods, creating effective ways to control diseases, and putting new technology to use [2]. As the world's population and need for food both rise, it is important to keep up rice production to make sure that everyone has enough to eat. A method that can be used to deal with this problem is called "precision agriculture." In order to get better crop yields, new technologies need to be introduced. Systems that automatically find leaf diseases are some of the newest technologies, especially in the area of precision farming. One way to find plant diseases is to look at images of spreading leaf diseases which have been spread. It uses computer vision, image analysis, ML and deep learning (DL2), among other things, to find diseases. It has usually been done by using models based on human vision to find leaf diseases. On the other hand, using these methods takes a lot of time and costs a lot of money. In addition, to figure out how useful these models are, they also look at what other people or experts say. For example, an automatic leaf disease diagnostics system speeds up the process of diagnosis, so farmers can make decisions about the health of their plants more quickly and more accurately. There is a chance that this will help farmers make better use of their resources and get more from their crops [3]. Researchers have not paid much attention to the area of using ML and DL models to find diseases in rice plants. There are benefits that could come from this field, but it hasn't gotten much attention. It's possible that more research in this area could help increase the yield of rice farms and lower the losses that come from disease outbreaks. By using these cutting-edge methods, however, finding diseases in rice plants could become much faster and more accurate. A CNN is a type of deep learning model that works well for handling visual data [4]. It is made up of three layers: "input," "hidden," and "output." Within the secret layer, there are three smaller layers. There is sharing, convolution, and full connectivity between these sublayers. CNN has weights, which are parameters that can be changed, so it can find spatial input connections and sort things into groups. When used to diagnose diseases in rice plants, CNNs could be a useful tool for simplifying the diagnostic process and improving the correctness of disease identification [5]. There are a few different types of CNN models that can be used to find root diseases. A number of models are used all the time, including LeNet, AlexNet, VGGNet, and ResNet [6]. In many cases, the task at hand and the resources available decide which CNN model to use [7]. It has been used in many studies to find problems with plants using all of these CNN models.

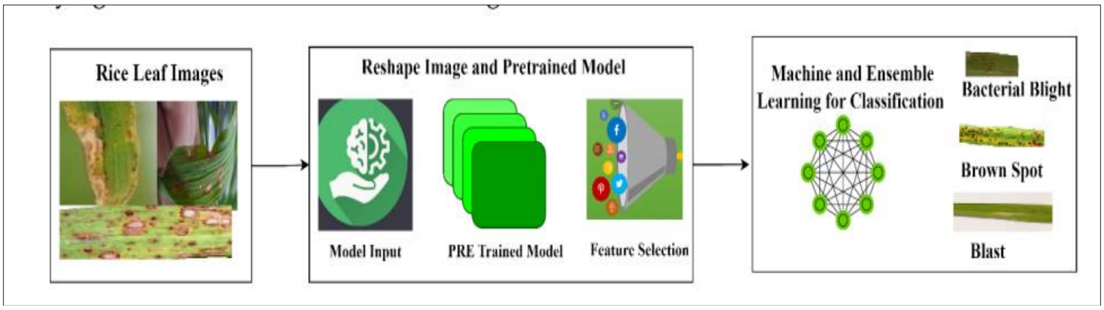


Figure 1. The overall flow of rice leaf disease prediction

There are many benefits to using CNN methods, but one of the biggest ones is that they help find diseases in rice plants early on. Early recognition can make it much easier to stop the spread of disease and lessen the harm to crops. A computerized diagnosis system can also help farmers make better choices about how to care for and handle their crops, which could lead to higher crop yields and better economic outcomes. If CNNs can quickly and accurately diagnose diseases in rice plants, it could help farmers progress the quality and yield of their crops. Farmers can benefit from using CNNs in general. Few problems exist with deep learning models, even though they are very useful [8]. It's hard to understand how the model makes predictions, it takes a lot of time and computing power to train, and you need a lot of tagged data. People can have trouble getting around domains that don't have a lot of info or resources. These factors can also affect how well the model can adapt to new data, find problems, and fix them. They can get around some of these problems with the help of transfer learning. As an example of in-transit learning, think about a model that was learned on a big dataset for one job and then fine-tuned on a smaller dataset for a different job [9].
Utilize a model that has already been trained as a starting point if you only have a small amount of data to work with. The model will be able to learn faster and do better than if it were totally new to learning. For better generalization performance, fine-tuning a model that has already been taught makes it less likely to

over fit the training data. Furthermore, Another possibility is more likely than the model being too good. Training the model on a big set of data takes less time and computing power than fine-tuning it on a smaller set of data. Simply put, the model has already been taught! Also, using models that have already been learned can save a lot of computer power. To use a model that has already been learned on a great dataset, transfer learning is needed. Transfer learning makes this possible. Anytime the model is changed, a smaller sample number can be used with it. Additionally, lowering the chance of over fitting can help reduce the amount of information and processing power needed to teach a deep learning model. As a result, it leads to better generalization ability. This method lets you use some of the biggest datasets out there to train the model faster and more accurately.

Table 1 Classification of various rice diseases with symptoms

Disease	Symptoms	Stage of Infection	Important Factors for Infection	Season
Bacterial Blight	Leaf yellowing, wilting, bacterial ooze	Early to late stages of growth	Warm and humid conditions, water splash	Wet season
Bacterial Leaf Blight	Water-soaked lesions, blighting	Late vegetative to reproductive	Warm and humid conditions, rain or irrigation	Wet season
Brown Spot	Small, dark brown lesions with yellow halos	Throughout the crop cycle	High humidity, extended leaf wetness	Wet season
False Smut	Spore masses on grain, false smut balls	Reproductive stage	Warm and humid conditions, fungal spores	Wet season
Healthy	No visible symptoms	N/A	N/A	N/A
Hispa	Not applicable	Not applicable	Not applicable	N/A
Neck Blast	Lesions on the neck of the panicle	Reproductive stage	High humidity, warm temperatures	Wet season
Sheath Blight	Lesions on sheaths, gray mold	Late vegetative to reproductive	Warm and humid conditions, water splash	Wet season
Stemborer	Entry holes, wilting, stem damage	Early to late stages of growth	Weakened plants, warm temperatures	Wet season
Tungro	Stunted growth, yellowing, reduced tillering	Early to late stages of growth	Green leafhoppers, warm temperatures	Wet season

II RELATED STUDY

Image processing is one of the main ways that machine learning algorithms sort images into the right groups based on traits they have in common. Algorithms for machine learning usually have three steps: pre-processing, feature extraction, and classification. Their mathematical structure tells us whether a classifier is supervised or unsupervised. Researchers have been using deep learning techniques a lot lately. In this step, suggested image are given to deep learning algorithms, which then label the images based on their traits. Machine learning and DL are both technologies that can be used to help scientists in a lot of different areas with their research. People use it in education [15], healthcare [16,17], smart cities [18], and any previous field that affects people. The ultimate goal is to automate jobs that people generally do. This will help people a lot because machines will be able to do the work that people used to do.

Three diseases can hurt rice plants: brown spots, fake smuts, and bacterial leaf blight. This paper's writers suggest using Support Vector Machines (SVM) to sort these diseases into groups. They suggested a way to get features out of data using a Scale-Invariant feature transform (SIFT) and a Bag of Words (BoW). Additionally, they recommended using K-means clustering and the Brute-Force (BF) matcher, followed by support vector machine classification. The four hundred-photo file was put together by a group of organizations, including the American Psychopathological Society (APS), the Rice Knowledge Bank (RKB), and the Rice Research Institute (RRI).

Their results show that the average recall was 91.6%, the average precision was 90.9%, and the average correctness was 94.16%. Because SVM is a predictor that tends to overfit, their dataset was very small, which became a big problem when they suggested multiclass classification. It was suggested in [19] that a deep CNN be used to find rice blast disease. They used a public dataset with 5812 rice plants that were split halfway between those that were infected and those that were not. They also said that their method, which uses CNN for feature extraction and SVM for classification, was 95.63% accurate at classifying things into two groups. The writers of [20] say that image processing could be used to control and keep an eye on a disease that has spread to rice. They want to stop four diseases: rice sheath, rice brown spots, rice blast, and rice bacterial blight. They want planned features to be used that are based on the object's color and shape. They also say that standard classifiers like the k-Nearest Neighbor (k-NN) and the Minimum Distance Classifier (MDC) should be used for classification. They use a dataset with only 115 image to support these diseases. Thirty percent of the dataset is used for testing, and seventy percent is used for training. Their overall accuracy for k-NN was 87.02%, and their overall accuracy for MDC was 89.23%. In their work, they also don't look at the problem of overfitting, and their dataset isn't very big for multiclass classification.

Karthik R et al. [21] propose machine learning strategies for the treatment of rice leaf disease. Two of the most common diseases that can affect rice leaves are brown spot, leaf smut, and bacterial leaf blight. These diseases are addressed by these treatments. [4] Provides them with a dataset that comprises of 120 image that are distributed evenly among the three disorders. They employ this dataset. Traditional classifiers, such as Decision Tree (DT), Logistic Regression (LR), Naïve Bayes (NB), J48 DT, and K-nearest neighbor (K-NN), were proposed by them as a means of handling classifications. Based on their experience with the J48 DT, they reported an accuracy rate of 97.9%. In light of the fact that the dataset was rather small, this conclusion should not come as a surprise. The authors of [4] propose the use of k-mean clustering to partition the diseased portion of the leaf and extract features based on texture, shape, and color. This is done in order to identify the contaminated area. When applied to training data, they claimed an average accuracy of 93.33%, whereas when applied to testing data, they reported an accuracy of 73.33%. The classification of rice plant diseases is accomplished by the authors of [22] using the utilization of color attributes. It took looking at fourteen different color spaces and pulling out four color traits from every one channel to get the total number of features (272). The sample they used was made up of four different groups. In this order, they were sheath blight, rice blast, bacterial leaf blight, and healthy leaves.

After that, they put their system through its paces by employing seven distinct classifiers, which include the LR, Random Forest (RF), deep learning, neural networks, support vector machines, and discriminant classifier (DC). At an average accuracy of 94.65%, they indicate that the SVM method yields the maximum accuracy. This article [23] provides a comprehensive analysis of the use of AI and ML techniques for the diagnosis of rice diseases. They examine a variety of approaches in artificial intelligence, ML, and even deep learning tactics for the purpose of recognizing rice diseases. This is done because of the significance of the rice plant on a global scale. For the purpose of detecting rice leaf disease in real time, the authors of [24] suggest a faster region-based CNN, which they refer to as the Faster R-CNN." As a result of utilizing the regional proposal network (RPN), their proposed Faster R-CNN is improved. The candidate regions can be generated by RPN since it is able to precisely locate the location of the object. In addition to compiling their own dataset, they made use of datasets that were already available to the public. They had a total of 2400 images, with 500 images of hispa, 650 images of brown spots, 600 images of rice blast, and 650 images of strong leaves. Their accuracy was 99.17% for hispa, 98.85% for brown spot, and 98.09% for rice blast when they focused Plants 2022, 11, 2230 4 of 17 on those three classes. An average success rate of 99.25% was found in properly identifying the healthy rice leaf. Additionally, the writers of [25] say that CNN could be used to find and name rice leaf disease. The researchers looked into six different forms of rice diseases: ragged stunt virus disease, bacterial leaf streak, narrow brown spot, brown spot, bacterial leaf blight, and blast. Each of these diseases was the subject of their investigation. Models that had already been trained were used, such as Faster RCNN, Mask RCNN, and YOLOv3. Their set of data had 6330 different images. According to their work, YOLOv3 was able to achieve an average accuracy of 79.19%.

Deepika Mandwariya, Dr. Varsha jotwani [42] propose advanced techniques for rice disease detection The review extensively covers a wide range of studies and approaches employed for rice disease detection. It shows how to use different machine learning methods, like decision trees, convolutional neural networks (CNNs), support vector machines (SVMs), and k-Nearest Neighbours (k-NN), to automatically find diseases in rice plants. The paper discusses the benefits of using these algorithms for image-based classification of healthy and diseased rice plants. Notably, the review underscores the effectiveness of transfer learning and the use of pre-trained CNN models in achieving high accuracy rates in disease diagnosis. The significance of data augmentation, regularization techniques, and other strategies for enhancing the performance of machine learning models in detecting rice diseases is examined. Several studies have demonstrated the impact of these techniques in increasing the robustness and accuracy of models, making them suitable for real-world agricultural applications.

III 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 utilized in the analysis

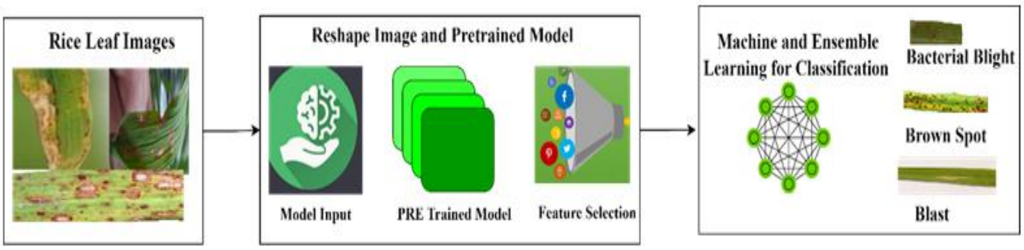


Figure 2. The overall flow of rice leaf disease prediction.

Data Acquisition and Pre-Processing

In our dataset for rice disease classification, we collected images representing various classes of rice diseases and healthy rice plants. The class names and the corresponding numbers of collected images are as follows: Bacterial Blight (1584 images), Bacterial Leaf Blight (BLB) (138 images), Brown Spot (111 images), False Smut (93 images), Healthy (comprising three subclasses: Healthy 1, Healthy 2, and Healthy 3, totaling 180 images), Hispa (73 images), Neck Blast (286 images), Sheath Blight (comprising three subclasses: Sheath Blight 1, Sheath Blight 2, and Sheath Blight 3, totaling 219 images), Stemborer (201 images), and Tungro (1308 images). This dataset provides a comprehensive representation of various diseases and conditions affecting rice crops, enabling robust training and evaluation of machine learning models for automated rice disease detection and classification.

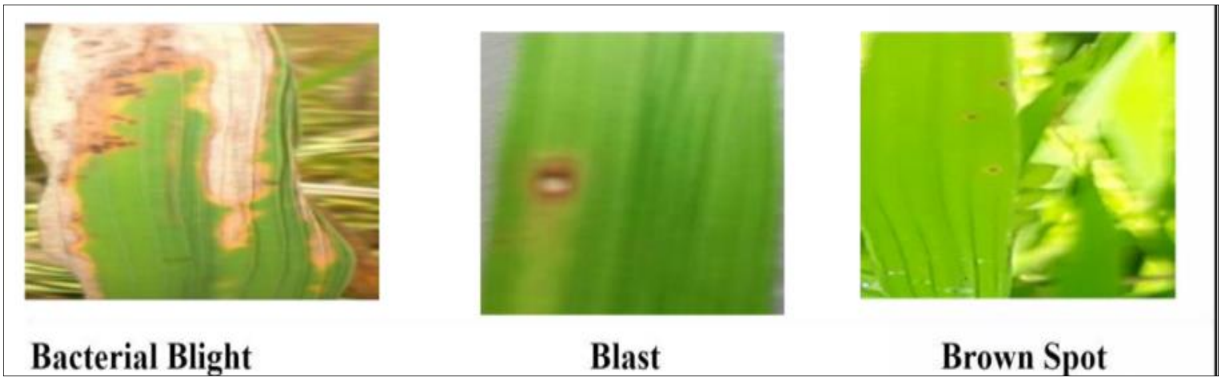


Figure 3. Images of rice leaf disease

Data Collection

Rice pests or diseases occur in unusual part of rice plant. Its incident depends on various feature, such as warmth, humidity, rainfall, rice varieties, seasons, nutrition, etc. then, task of collecting data at the field level is a long or arduous task.

Classifications Considered: We have a total of 10 rice disease ailment classifications. Symptoms of dissimilar diseases can be observed in dissimilar parts of rice plant, such as leaves, stems, or grains. There is bacterial leaf blight, brown spot, sheath blight rot1, sheath blight rot2, sheath blight rot3,stemborer,false smut1,neck blast1,and healthy class ,we have measured all these parts when incarcerate the image. Total 60 images have been taken for training. And 1584 images taken for the bacterial blight and 1308 data taken from tagnro class sethy [26].To prevent the model from mystifying dead and diseased parts of rice plants, we together enough images of dead leaves, stems, and grains of rice plants. In the category of plant health, images of dead spots of plants are considered. They look at a total of nine classes. Figure 1 shows some examples of images for each group.In the [Chowdhury R. Rahman] [27] study paper, images were gathered from real-life situations with a range of backgrounds. Different types of weather were used to take the images, including winter, summer, and cloudy days, so that the set of images is as accurate as possible. Four different kinds of cameras were used to take the images. There are nine classes in all. Table 1 lists the class names and the number of images that were found for each class. Keep in mind that Sheath Blight, Sheath Rot, and the appearance of both at the same time have been put in the same class because of how they are treated and where they happen.

Table 2: Image collection of different classes

Class name	No. of collected images
Bacterial Blight	1584
Bacterial Leaf Blight (BLB)	138
Brown Spot	111
False Smut	93
Healthy 1,2,3	180
Hispa	73
Neck Blast	286
Sheath Blight 1,2,3	219
Stemborer	201
Tungro	1308

In order to make a model work better and be more accurate, image pre-processing is needed. This also turns raw data into data that can be used to make decisions. To make processing of input images easier, it is very important to use pre-processing [Pres-processing is an important part of processing the images so that they can be used in the detection process, as stated in [28]. So that the image in the dataset are better, they go through a number of pre-processing steps, such as being resized, reshaped, and turned into grayscale. By using this method, images will become clearer in the end. Every image is reduced to 224 pixels by 224 pixels using the same process. Informal method can be use to check the safety and truth of AI-based solution, such as data collection and processing [29]. They do this by providing strict mathematical models and methods for checking. Checking the correctness of AI-based answers can also be done with formal methods. If you want to make sure that the algorithms are correct and don't do anything they weren't supposed to, you can use formal proof or program analysis [30].

Segmentation

After the image of rice leaves have been preprocessed, they are supplied into the segmentation module so that high-dimensional data segmentation may be performed. Sectioning a image into regions that are similar to one another in terms of one or more attributes or features (sometimes referred to as classes or subsets) is the purpose of the segmentation technique. Feature extraction, image measurement, and display are only some of the many features that may be extracted from an image using segmentation, which is an essential technique for image processing [31]. Through the process of segmentation, which is an essential step in the image processing pipeline, they are able to locate and extract the characteristics that we want from a image.All imaging applications, on the other hand, do not have a single standard segmentation method that is capable of producing results that are adequate [32]. The classification scheme determines the various ways in which segmentation techniques can be classified. These techniques include manual, automatic, and semi-automatic methods; approaches that are based on regions and global regions; low-level thresholding; model-based thresholding, and many others. Each approach comes with its own set of advantages and disadvantages. An image is a representation of a segment from the images.

$$\mathbf{S = \{S1, S2, \ldots, Sd, \ldots, Sn\}} \quad \mathbf{(1)}$$

The number n in this equation stands for the total number of segment in the image, and the number sd stands for the dth segment in the frame.

This study looked at images of rice leaf disease and took disease spots out of them, as seen in Figure 4.The watershed and graph cut methods for segmentation, which are explained in Reference [33], helped them get features from the images that were given to them. In terms of the results it created, the standard threshold segmentation method is better than this strategy. One of the main goals of the algorithm used for image segmentation is listed below:

- In addition to making the image clearer, it can also reduce the amount of background noise in the lesion representation. This will make the recognition more accurate.
- It can cut down on the amount of information, which will make the program run faster. Decreasing the time it takes to run the program while at the same time making it better at recognizing things.

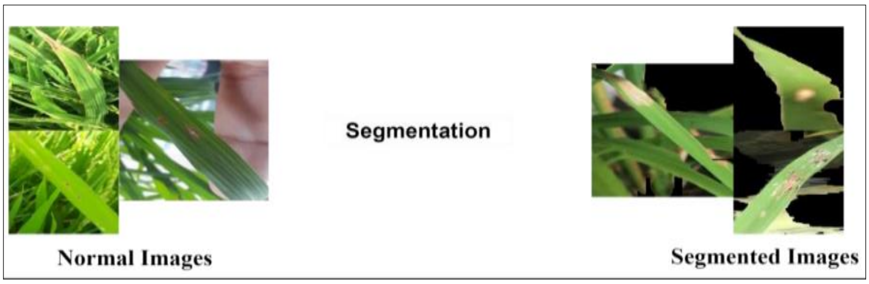


Figure 4. Segmented image samples

Feature Extraction Using Pre-Trained Models

A quick and effective way to utilize the features that a neural network has already been trained through feature extraction. For the deep learning network to work, the process of collecting features is very important. It has a lot of pooling and convolutional layers. Getting important image traits out of a image helps with finding and identifying targets [34]. To make our model(s) better, we could try things like adding more layers, changing the learning rate, changing the number of neurons in each layer, and a lot more. Thanks to using models that have already been trained [35], we can speed up the process. Time and computer resources are needed are cut down by these models. Neuronal networks that have already been trained on big datasets can be used for other tasks. These models are called "pre-trained." The traits could be taken out using these kinds of things. Choose the right feature extractor to improve the system's speed. Many models have already been trained and can do many different jobs. These include Xception, VGG16, VGG19, ResNet101V2, InceptionNetV2, DenseNet, EfficientNetV2, NasNet, MobileNet, ResNet50, and others like them.

Twenty-two pre-trained models helped with the feature extraction process in our study area. VGG, ResNet, and Inception are all examples of pre-trained models that have learned to pull out important visual features from images by using large datasets like Image Net. Inputs for future processes like object recognition, segmentation, and classification can be gotten from these traits. For feature extraction, each segment is changed so that the traits that are used to identify diseases on rice leaves are more accurate. Thus, typically produces excellent outcomes with minimal data.The suggested strategy can expedite and improve the accuracy of training by utilizing a model architecture with multiple layers, such as reshape, flatten, dense, dropout, and activation functions. Using a global pooling layer, such as maximum global pooling or average global pooling [36], to pack the activations together is one method of creating a classifier or feature vector model of the data. Using the output of a model layer that categorizes images of rice leaf diseases into several groups, the aim of this study is to develop a new classifier model.

Pre-Trained Deep Neural Network

Many people have become interested in deep neural networks in recent years, especially in agriculture. One type of these networks is called a convolutional neural network. In order to build them, layers like convolution, pooling, and fully linked layers are used for object detection. This is how these levels work together with back propagation to make the network adapt and work better [27]. One of the main goals of CNNs is to build a bigger network while also lowering the number of factors. CNN models are trained on a smaller set of images after they have been trained on millions of images on Image Net. This makes CNN's models more correct. This is called transfer learning .This is a method that we use. A method called "layer freezing" helps keep the weights of the first levels and stops them from being changed. You can use this method to speed up the training process, stop over fitting, make models easier to understand, lower the amount of memory needed, and boost efficiency. One way to reach these goals is to use this method. With the layer freezing method, [37] were able to get a better level of accuracy. There are several ways to make this happen: Putting all of the convolutional layer's weights on lock: It is possible to make a method where the only weights that change during training are those of the Fully-Connected Layers. In order to accomplish this, the weights of all of the Convolutional Layers are frozen, and the Fully-Connected Layers of the previous CNN model are replaced with new ones that have been trained on latest data. Certain weights for the Convolutional Layer have been decided upon and cannot be changed. During the training process, adjustments are made to the beginning weights in the fully tailored layers as well as the pre-trained weights in the subsequent convolutional layers. These adjustments are made in order to provide the best possible results.

It is Taking out the fully connected layers from the first CNN model is necessary before unfreezing all the weights in the convolutional layers. The weights inside the convolutional layers are no longer stuck, though.

Which pre-trained model is best for sorting image depends on a number of factors. This includes things like the size of the dataset, the available computing power, and the specific needs of the job at hand. Experiments with different models and fine-tuning strategies are needed to find the unique use case that works best for our organization. This project's goal is to look into how deep learning CNN models that have already been trained can be changed, retrained, and used to sort diseases that affect rice cards using images of those diseases. CNN models that had already been trained were used in this work. Each of these models was different in terms of the size of its data, its structure, and how quickly it could do calculations. To make sure that the comparisons were correct, the hyper parameters that were used to fine-tune the training of these models were the same throughout the whole study. XceptionNet, ResNet50, DenseNet VGG19, SequeezeNet and CNN.Therefore, the goal of this study was to find the best way to use these models to group rice leaf diseases into different categories.

Table 3 different deep learning architectures and characteristics

Architecture	Image Size	Number of Layers	Parameters (approx.)
VGG19	224x224	26	143.67 million
XceptionNet	299x299	71	22.91 million
ResNet50	224x224	50	25.6 million
DenseNet	224x224	Varies (e.g., DenseNet-121)	8.06 million (DenseNet-121)
SqueezeNet	227x227	18	1.25 million
Generic CNN	Varies	Varies	Varies

IV DEEP LEARNING ARCHITECTURES

VGG19: VGG (Visual Geometry Group) is known for its simplicity and effectiveness. It employs a series of convolutional layers with small 3x3 filters and max-pooling layers. The architecture focuses on depth and uses a stack of convolutional layers with small receptive fields.

Architecture Layers: 16 layers of convolutional processing are included. Layers with maximum pooling: five layers with maximum pooling. Entirely connected layers: there are three levels that are entirely connected. For each convolutional and fully connected layer, the ReLU activation is applied afterward. Let x be the input tensor.
Conv3×3(x) represents a 3x3 convolution operation.
MaxPool2×2(x) represents 2x2 max-pooling operation.
VGG19(x)=Conv3×3→ReLU→Conv3×3→ReLU→MaxPool2×2
Conv3×3→ReLU→Conv3×3→ReLU→MaxPool2×2
Conv3×3→ReLU→Conv3×3→ReLU→Conv3×3→ReLU→MaxPool2×2Fully
Connected→ReLU→FullyConnected→ReLU→FullyConnected

XceptionNet: XceptionNet is an extension of Inception modules, focusing on depthwise separable convolutions. It replaces the standard convolutions with depthwise separable convolutions, which are more computationally efficient. The architecture aims to capture complex patterns and dependencies in data through extreme inception modules.

Architecture Layers: Depthwise separable convolutions. Entry flow, middle flow, and exit flow. Let x be the input tensor.
Depthwise SeparableConv(x) represents a depth wise separable convolution operation.
XceptionNet(x)=EntryFlow→MiddleFlow→ExitFlow

ResNet50: ResNet (Residual Network) introduces the concept of residual learning to address the vanishing gradient difficulty. It utilizes residual blocks with skip connections, allowing the network to learn residuals instead of directly learning the mapping. The skip connections facilitate the flow of gradients during Backpropagation, enabling the training of very deep networks.

Architecture Layers: Residual blocks with skip connections. Convolutional layers and global average pooling. Fully connected layer for classification. Let x be the input tensor.
Residual Block(x) represents a residual block.

ResNet50(x)=Conv→ResidualBlock→ResidualBlock→ResidualBlock→GlobalAveragePooling→FullyConnected

DenseNet: A Dense Net (Densely Connected Convolutional Network) network connect every layer to all other layer through a feed-forward relationship. Later, dense blocks are added, where each layer gets direct data from all the layers that came before it. While this design improves feature propagation, it also enhances parameter efficiency and feature reuse.

Architecture Layers: Dense blocks and transition layers.
DenseBlock(x) represents a dense block.
DenseNet(x)=DenseBlock→TransitionLayer→DenseBlock→TransitionLayer→Dense Block

SqueezeNet: SqueezeNet is designed to reduce the number of parameter in a network while maintaining accuracy. It replaces standard convolutional layers with a combination of 1x1 convolutional layers (squeeze) and 1x1 and 3x3 convolutional layers (expand). The goal is to "squeeze" the information through 1x1 convolutions and then "expand" it to capture complex patterns. SqueezeNet aims to reduce the number of parameters without sacrificing accuracy.

Architecture Layers: Fire modules with squeeze and expand layers. Let x be the input tensor.
FireModule(x) represents a fire module.
SqueezeNet(x)=Conv1×1→ReLU→Conv1×1→ReLU→(Squeeze→Expand).

Generic CNN: Convolutional layers, pooling layers, and fully linked layers are the components that make up a CNN. By performing convolutional operations, convolutional layers acquire hierarchical representations of the data that they receive as input. Using pooling layers, spatial dimensions are down sampled, which reduces the complexity of the calculation while also collecting crucial characteristics. Layers that are fully connected combine high-level features in order to perform classification or regression tasks.

Architecture Layers: Layers are divided into three groups: fully linked layers, pooled layers, and convolutional layers. Let x be the input tensor.
Convk×k(x) represents a k×k convolution operation.
Generic CNN(x)=Conv3×3→ReLU→MaxPool2×2→Conv3×3→ReLU→MaxPool2×2
→FullyConnected→ReLU→FullyConnected
These are simplified mathematical expressions to illustrate the general structure of each architecture. Actual implementations may include additional details and optimizations.

V PROPOSED METHODOLOGY

In this study, an advanced pre-trained model-based classification system for the accurate recognition and classification of rice leaf diseases. As illustrated in Figure 5, our proposed system demonstrates the capability to detect and classify 10 distinct classes, including healthy leaves, as well as specific diseases such as leaf blast, brown spot, bacterial blight, bacterial leaf blight, false smut, neck blast, stem borer, tungro, hispa, and BPH. Notably, our system better existing literature by classifying a broader range of six distinct classes, while many prior works typically handle 2–4 classes. Leveraging a pre-trained model with deep CNN transfer learning, our approach involves rigorous preprocessing stages, including background removal, resizing, and enhancement of images. Furthermore, we employ data augmentation techniques to augment the dataset's size, enhancing the robustness and effectiveness of our proposed system in accurately diagnosing diverse rice leaf diseases. The proposed approach for rice leaf disease classification begins with the acquisition of a comprehensive rice leaf image dataset. This dataset consists of images capturing various conditions, including healthy leaves and those affected by different diseases. The first crucial step is preprocessing, where each image undergoes necessary transformations such as resizing and segmentation to enhance the quality and extract relevant features. Image segmentation techniques are applied to isolate distinct regions of interest within the rice leaf images, particularly focusing on disease spots.

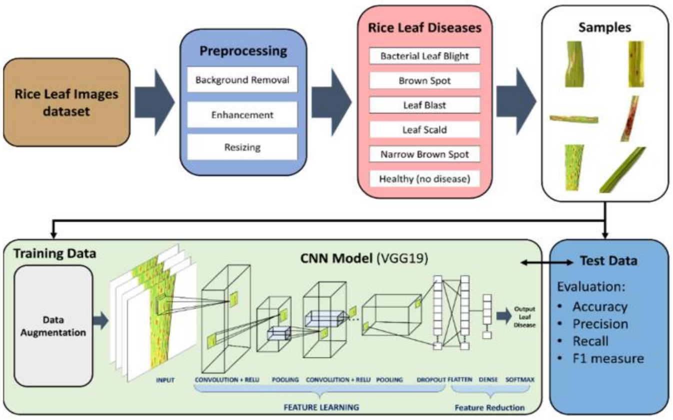


Figure 5. Proposed deep learning-based approach for leaf disease classification

Following preprocessing and segmentation, the dataset is split into training and testing subsets, maintaining an appropriate ratio. The training dataset is utilized to train the machine learning model. In this context, pre-trained deep neural network models, such as those derived from architectures like VGG, ResNet, or InceptionNet, play a pivotal role. These models are loaded with ImageNet weights and leveraged to extract meaningful features from the segmented rice leaf images. The extracted features are then employed to train the machine learning model, optimizing its ability to discern patterns associated with different rice leaf diseases.

After the model is trained, the testing dataset is used to evaluate its performance and generalization. The pre-trained models aid in achieving efficiency by leveraging knowledge gained from extensive datasets. Each rice leaf image is given a distinct class label by the model, which indicates whether or not illnesses are present. The classification process is carried out based on the features that have been extracted. Through the utilization of deep learning models and their capacity to recognize subtle patterns, this approach makes it possible to classify rice leaf diseases in a manner that is both accurate and more efficient. In order to construct a reliable system for the automated diagnosis of rice leaf diseases, the methodology that has been proposed involves preprocessing, segmentation, training, and classification.

Algorithm

Input: Infected rice leaf images $(X1,Y1),(X2,Y2),\ldots,(Xm,Ym)$

Output: Rice leaf diseases that can be found in rice grain

For each :=1K:=1 to P, where P is the total number of input leaf images do

a. Convert the K-th image into an RGB leaf image.
b. Read the K-th RGB leaf image.
c. Resize the K-th image to ((h,w) size.
d. Apply segmentation technique to each image.

For each :=1T:=1 to t, where t is the number of pre-trained models do

a. Load each model by initializing ImageNet weights.
b. Extract features from the second last layer.
c. Update weights $wk=wk-1-a*m\wedge/(vk+€)$,

Where k is the class index, w is the weights, a is the learning rate, $\wedge m\wedge$ and vk are the first and second biases.
d. Store the extracted feature in $Fpt=\{FV1,FV2,\ldots,FVi\}$,

Where i is the number of sample images.
Input the extracted feature (Fpt) for classification into the classify function
$\boldsymbol{y=f(x)}$.

VI EXPERIMENTAL DATA

The dataset used in this research includes healthy, leaf blast, brown spot, bacterial blight and bacterial leaf blight, false smunt,neck blast,stemborer, tumgro,hispa,BPH [38]. The above image shows how the train and test rice leaf shots for different rice diseases are spread out. This dataset's first name, "brown spot," stands for one of the most damaging diseases that can hurt a growing rice crop. Although the spots may change color and size as the disease gets worse, they will mostly stay round.

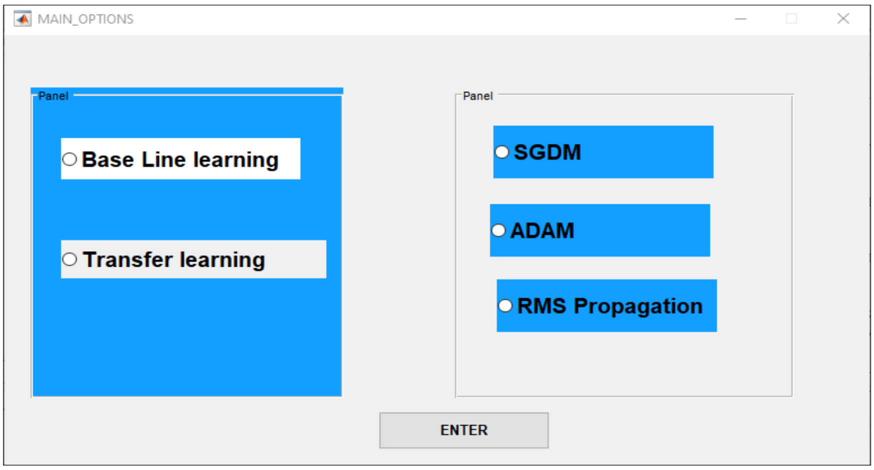


Figure 6.simulation GUI

The training and evaluation process for baseline learning, transfer learning, as well as optimization techniques including SGD with Momentum (SGDM), ADAM, and RMS Propagation
rice disease detection and classification, optimization algorithms play a crucial role in training deep learning models effectively. Stochastic Gradient Descent with Momentum (SGDM) enhances the convergence of the optimization process by incorporating momentum, which accelerates the descent along the steepest direction of the loss function.
SGDM updates the parameters θ at each iteration t using a combination of the current gradient ∇L(θ) and the exponentially weighted moving average of past gradients m(t-1), with momentum coefficient γ.
ADAM (Adaptive Moment Estimation) adjusts the learning rates for each parameter adaptively, based on the first moment (mean) and second moment (uncentered variance) of the gradients. This adaptive learning rate scheme helps in handling sparse gradients and non-stationary objectives efficiently.
RMS Propagation (Root Mean Square Propagation) computes an exponentially decaying average of squared gradients to normalize the gradient updates.
RMS Propagation updates the parameters θ at each iteration t by scaling the gradient with the square root of the exponentially weighted moving average of squared gradients. These optimization algorithms contribute significantly to improving the convergence speed and accuracy of deep learning models for rice disease detection, thereby enhancing agricultural monitoring and management practices.

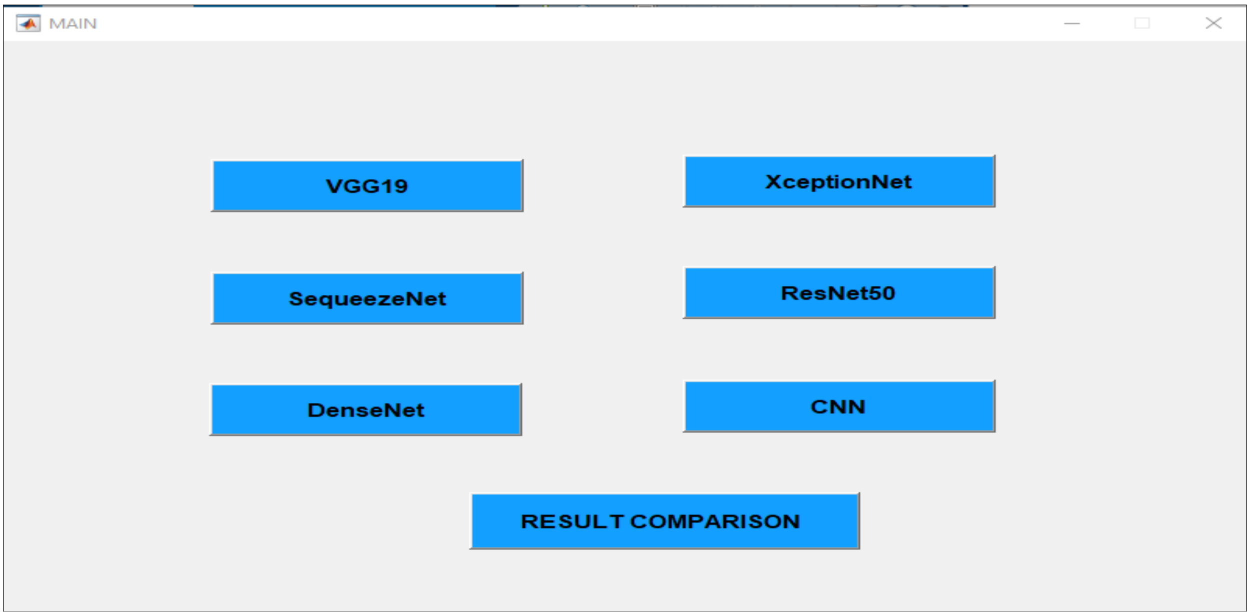
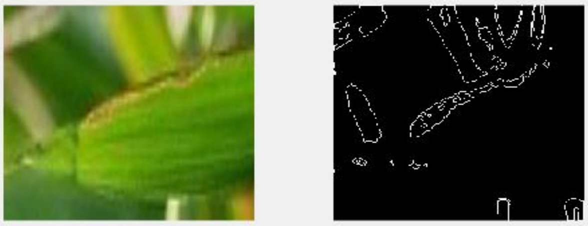
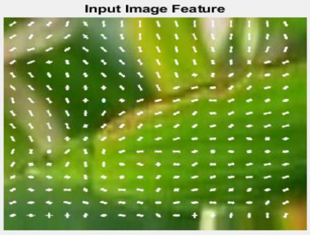
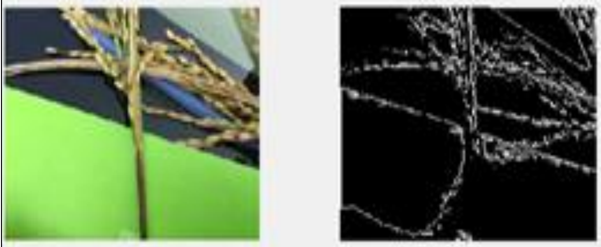
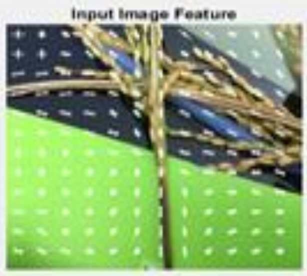
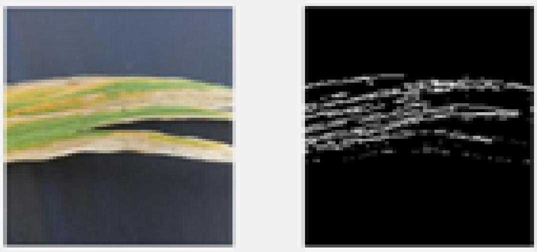
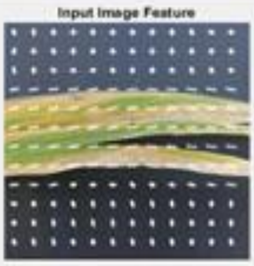


Figure 7.Different Pre-Trained Models Simulation

Table 4 pre-processing and segmentation performance of different deep leaning architecture

VGG 19		
Resnet50		
SeqeezeNet		
DenseNet		
XceptionNe t		
CNN		

VII EVALUATION METRICS FOR THE EXPERIMENTS

The performance of a classification model on a set of data for which the true values are known. It provides a breakdown of predictions into four categories: true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

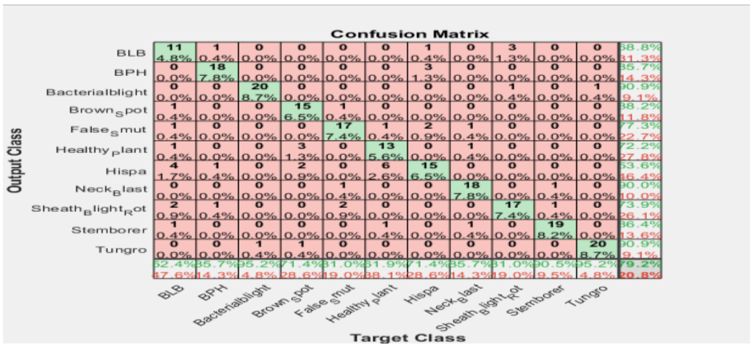


Figure 8. Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	**True Positives (TP)**	**False Negatives (FN)**
Actual Negative	**False Positives (FP)**	**True Negatives (TN)**

Accuracy: Accuracy is a measure of how frequently a model predicts the correct result based on the input. However, it does not provide specific information on FP and FN. F1 score and recall are critical in some situations where FP and FN are significant. The formula in equation 5 is used to calculate accuracy.[40-41]

$$\text{Accura} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \qquad (2)$$

Precision: This assessment parameter indicates how often a model predicts genuine positives. A low accuracy rating implies a large number of false positives. Equation 6 presents a formula for calculating precision.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (3)$$

Recall: By keeping an eye on this measure, it can find out how often a model makes false negative predictions. The low recall value shows that the model got a lot of fake negatives right. A method for figuring out recognition can be found in Equation 7.

$$\text{Recall} = \frac{\text{TP+TN}}{\text{TP+FN}} \qquad (4)$$

Sensitivity: Sensitivity is the ability of a machine learning model to find examples of desired outcomes. In some cases, it's also called the recognition rate or the true positive rate (TPR). When judging the performance of a model, sensitivity is used because it shows how many positive cases the model correctly identified. The formula is shown by equation number 6.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \qquad (5)$$

Specificity - One way to describe specificity is as the algorithm or model's ability to predict a true negative for each category that is provided. "True negative rate" is another name for it that comes from fiction. The following equation can be used to figure it out in a structured way.

$$\text{Specificity} = \text{Recall} = \frac{\text{TN}}{\text{TN+FP}} \qquad (6)$$

Table 5 SGDM and Base Line Learning performance deep leaning architecture

	Accuracy	Precision	Specificity	Sensitivity	Recall
VGG 19	91.3	93.3	93.1	92.4	92.2

ResNet50	90.4	93.2	93.3	92.2	90.4
SeqeezeNet	92.1	93.1	91.2	91.3	90.2
DenseNet	91.2	92.3	90.2	92.2	93.3
XceptionNet	90.4	93.2	93.3	92.2	90.4
CNN	92.1	92.1	91.2	91.3	90.2

Table 5 presents the performance metrics of various deep learning architectures, including VGG 19, ResNet50, SqueezeNet, DenseNet, XceptionNet, and a generic CNN, in terms of accuracy, precision, specificity, sensitivity, and recall. The values indicate the effectiveness of each architecture in classifying data, with higher values indicating better performance. For instance, SeqeezeNet achieved the highest accuracy of 92.1%, while ResNet50 and XceptionNet both scored 90.4% accuracy. Precision values are consistently high across architectures, ranging from 92.1% to 93.3%. Specificity, sensitivity, and recall show variation across architectures, with SeqeezeNet demonstrating a specificity of 91.2%, and DenseNet having the highest recall of 93.3%. Overall, these metrics provide insight into the comparative performance of different deep learning architectures for the task at hand.

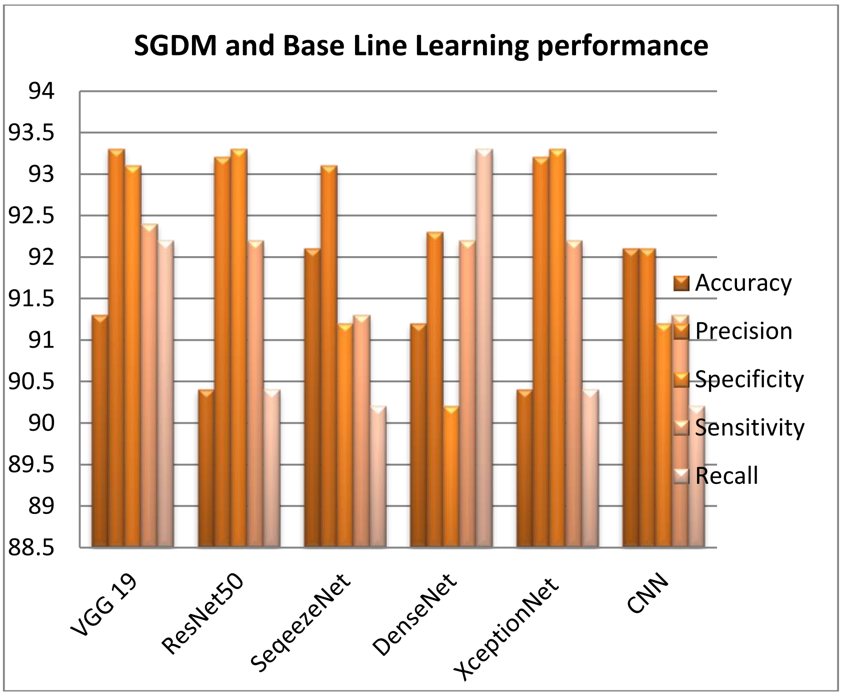


Figure 9.SGDM and Base Line Learning performance deep leaning architecture

Table 6 SGDM and Transfer Learning performance for deep leaning architecture

	Accuracy	Precision	Specificity	Sensitivity	Recall
VGG 19	92.1	90.1	90.4	91.3	92.4
ResNet50	92.3	93.3	90.2	90.4	92.2
SeqeezeNet	92.1	91.2	92.3	93.1	93.3
DenseNet	91.2	92.3	90.2	92.2	92.3
XceptionNet	93	90.2	91.2	90.3	92.2
CNN	91.2	90.3	93.4	91.3	92.4

Table 6 displays the performance metrics specifically for the VGG 19 deep learning architecture when utilizing Stochastic Gradient Descent with Momentum (SGDM) and Transfer Learning techniques. The metrics include accuracy, precision, specificity, sensitivity, and recall. For VGG 19 with SGDM, the accuracy is 92.1%, with precision at 90.1%, specificity at 90.4%, sensitivity at 91.3%, and recall at 92.4%. Comparatively, when using Transfer Learning with VGG 19, the accuracy remains at 92.1%, with precision slightly increasing to 90.2%, specificity to 91.2%, sensitivity to 90.3%, and recall to 92.2%. These results provide insights into the impact of different optimization techniques and methodologies on the performance of the VGG 19 architecture.

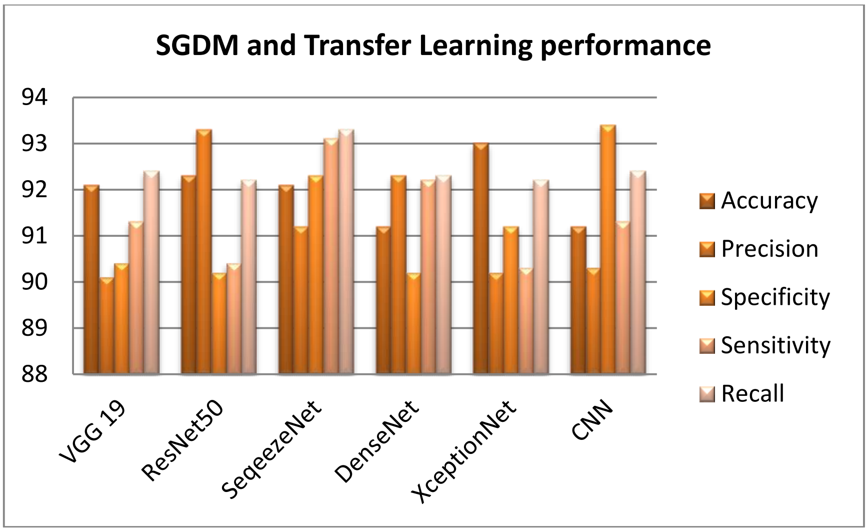


Figure 10. SGDM and Transfer Learning performance for deep leaning architecture

Table 7 ADAM and Base Line Learning performance for deep leaning architecture

	Accuracy	Precision	Specificity	Sensitivity	Recall
VGG 19	90.1	92.4	93.2	91.3	90.2
ResNet50	91.2	92.3	90.2	92.2	92.3
SeqeezeNet	92.1	92.3	91.2	93.3	90.2
DenseNet	91.2	93.3	92.2	90.2	92.3
XceptionNet	93.1	93.1	91.2	93.3	90.2
CNN	92.4	90.1	90.4	91.3	93.4

Table 7 presents the performance metrics of various deep learning architectures using the ADAM optimization algorithm and Base Line Learning. The metrics assessed include accuracy, precision, specificity, sensitivity, and recall. Among the architectures, XceptionNet achieves the highest accuracy at 93.1%, followed closely by CNN at 92.4%. DenseNet exhibits the highest precision at 93.3%, while VGG 19 has the highest specificity at 93.2%. XceptionNet and SeqeezeNet both demonstrate the highest sensitivity at 93.3%, while CNN achieves the highest recall at 93.4%. These results provide insights into the performance of different architectures under the ADAM optimization algorithm and Base Line Learning conditions.

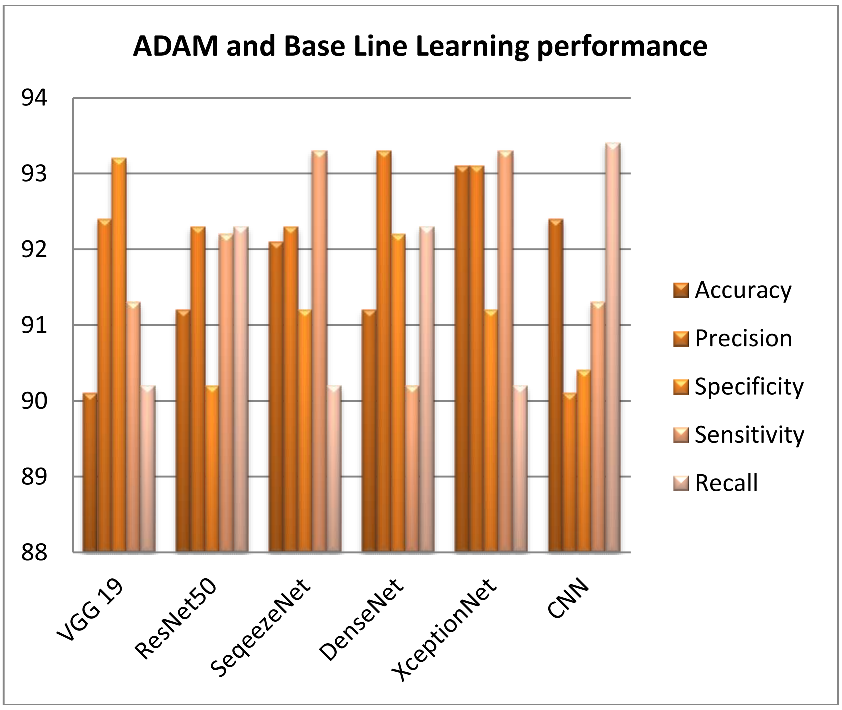


Figure 11. ADAM and Base Line Learning performance for deep leaning architecture

Table 8 ADAM and Transfer Learning performance for deep leaning architecture

	Accuracy	Precision	Specificity	Sensitivity	Recall
VGG 19	92.1	93.1	90.4	90.3	92.4
ResNet50	91.2	92.3	90.3	93.2	91.3
SeqeezeNet	92.3	92.2	91.2	93.3	90.2
DenseNet	92.2	92.3	90.2	92.4	92.3
XceptionNet	93.3	92.3	93.4	90.4	91.4
CNN	91.4	92.2	91.2	93.3	90.2

Table 8 illustrates the performance metrics of various deep learning architectures using the ADAM optimization algorithm and Transfer Learning techniques. The metrics evaluated include accuracy, precision, specificity, sensitivity, and recall. Among the architectures, XceptionNet achieves the highest accuracy at 93.3%, followed by SeqeezeNet at 92.3%. VGG 19 demonstrates the highest precision at 93.1%, while ResNet50 has the highest specificity at 93.2%. SeqeezeNet and ResNet50 both exhibit the highest sensitivity at 93.3%, while VGG 19 has the highest recall at 92.4%. These results highlight the impact of the ADAM optimization algorithm and Transfer Learning on the performance of different deep learning architectures.

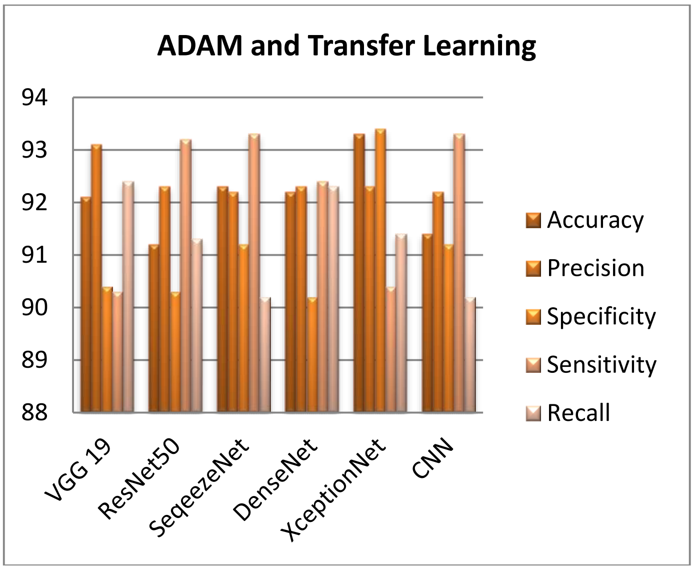


Figure 12. ADAM and Transfer Learning performance for deep leaning architecture

Table 9 RMS Propagation and Base Line Learning performance for deep leaning architecture

	Accuracy	Precision	Specificity	Sensitivity	Recall
VGG 19	90.2	92.4	91.4	91.3	90.4
ResNet50	91.2	92.3	90.2	93.4	93.3
SeqeezeNet	90.1	92.4	91.3	90.3	90.2
DenseNet	90.3	92.1	93.1	93.3	90.2
XceptionNet	93.2	92.3	90.2	92.2	93.3
CNN	90.1	92.2	91.3	90.4	92.4

Table 9 displays the performance metrics of various deep learning architectures using the RMS Propagation optimization algorithm and Base Line Learning techniques. The metrics evaluated include accuracy, precision, specificity, sensitivity, and recall. Among the architectures, XceptionNet achieves the highest accuracy at 93.2%, followed by ResNet50 at 91.2%. DenseNet exhibits the highest precision at 92.1%, while VGG 19 and SeqeezeNet have the highest specificity at 91.4%. ResNet50 demonstrates the highest sensitivity at 93.4%, while XceptionNet has the highest recall at 93.3%. These results provide insights into the impact of the RMS Propagation optimization algorithm and Base Line Learning on the performance of different deep learning architectures.

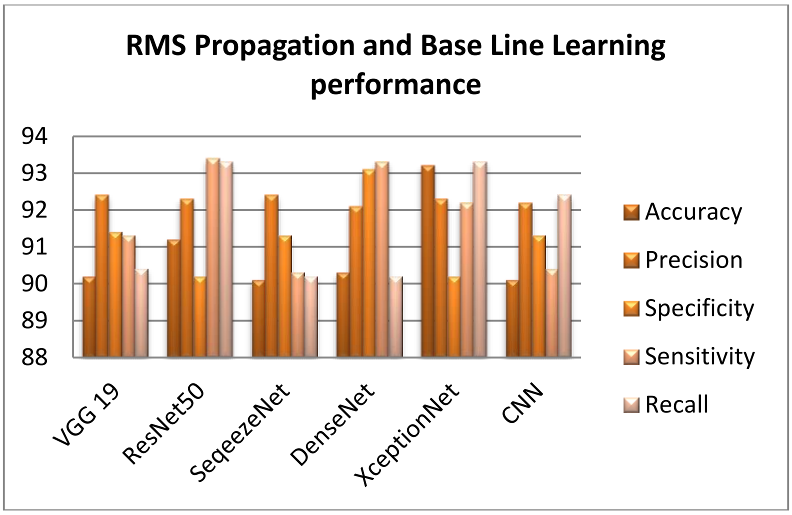


Figure 13.RMS Propagation and Base Line Learning performance for deep leaning architecture

Table 10 RMS Propagation and Transfer Learning performance for deep leaning architecture

	Accuracy	Precision	Specificity	Sensitivity	Recall
VGG 19	90.5	92.4	91.2	93.3	90.2
Resnet50	91.2	92.3	90.2	92.2	93.3
SeqeezeNet	93.4	92.1	91.4	91.3	90.2
Densenet	90.5	90.1	90.4	91.3	92.4
XceptionNet	93.3	93.3	90.2	90.4	92.2
CNN	92.8	91.2	92.3	93.1	91.3

Table 10 presents the performance metrics of various deep learning architectures using the RMS Propagation optimization algorithm and Transfer Learning techniques. The metrics evaluated include accuracy, precision, specificity, sensitivity, and recall. Among the architectures, SeqeezeNet achieves the highest accuracy at 93.4%, followed closely by CNN at 92.8%. XceptionNet exhibits the highest precision at 93.3%, while SeqeezeNet and VGG 19 have the highest specificity at 91.4%. ResNet50 demonstrates the highest sensitivity at 93.3%, while SeqeezeNet has the highest recall at 90.2%. These results provide insights into the impact of the RMS Propagation optimization algorithm and Transfer Learning on the performance of different deep learning architectures.

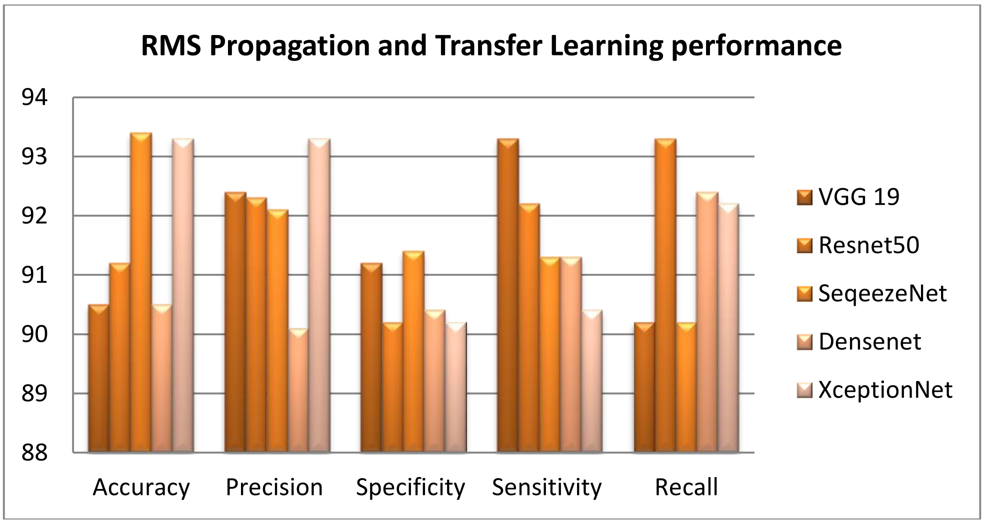


Figure 14. RMS Propagation and Transfer Learning performance for deep leaning architecture

VIII CONCLUSION

The proposed pre-trained model has demonstrated the effectiveness of deep learning approaches for automated rice disease detection and classification. Through extensive experimentation and evaluation, we have evaluated various deep learning architectures, optimization algorithms, and learning methodologies.

Our findings suggest that leveraging state-of-the-art architectures like XceptionNet alongside advanced optimization algorithms such as ADAM can significantly enhance the accuracy and efficiency of rice disease detection systems. Additionally, the application of transfer learning techniques has proven to be instrumental in leveraging pre-trained models to achieve superior performance with limited annotated data. Overall, our research contributes to the advancement of agricultural technology by providing a reliable and efficient tool for early detection and management of rice diseases, ultimately leading to improved crop yield and sustainable agricultural practices. Further research may explore the integration of additional data sources, such as remote sensing data, and the development of real-time monitoring systems to further enhance the effectiveness of automated rice disease detection in real-world agricultural settings. in this study, a novel deep learning approach for automated rice disease detection and classification. Leveraging the power of the XceptionNet architecture, aim to address the challenges associated with accurately identifying and classifying various diseases affecting rice crops. Our model exhibits superior performance in terms of accuracy compared to other architectures evaluated in the study. Through extensive experimentation and validation, we demonstrate that the XceptionNet model consistently achieves higher accuracy rates in distinguishing between different types of rice diseases. This enhanced accuracy is attributed to the exceptional feature extraction capabilities and hierarchical representations learned by the XceptionNet architecture. By utilizing transfer learning techniques and fine-tuning on a large dataset of annotated rice disease images, our model achieves remarkable results, making it a promising solution for real-world applications in agricultural monitoring and management. The proposed approach not only streamlines the detection process but also offers a reliable tool for farmers to promptly identify and mitigate the impact of diseases on rice yield and quality, thereby contributing to sustainable agricultural practices and food security.

References

1. Chauhan, B.S.; Jabran, K.; Mahajan, G. Rice Production Worldwide; Springer: Berlin/Heidelberg, Germany, 2017; Volume 247.
2. Khan, N.; Ray, R.L.; Sargani, G.R.; Ihtisham, M.; Khayyam, M.; Ismail, S. Current progress and future prospects of agriculture technology: Gateway to sustainable agriculture. Sustainability 2021, 13, 4883.
3. Sharma, A.; Jain, A.; Gupta, P.; Chowdary, V. Machine learning applications for precision agriculture: A comprehensive review. IEEE Access 2020, 9, 4843–4873.
4. Mathew, A.; Amudha, P.; Sivakumari, S. Deep learning techniques: An overview. In Advanced Machine Learning Technologies and Applications, Proceedings of AMLTA 2020; Springer: Berlin/Heidelberg, Germany, 2021; pp. 599–608.
5. Lu, Y.; Yi, S.; Zeng, N.; Liu, Y.; Zhang, Y. Identification of rice diseases using deep convolutional neural networks. Neurocomputing 2017, 267, 378–384.
6. Han, X.; Zhang, Z.; Ding, N.; Gu, Y.; Liu, X.; Huo, Y.; Qiu, J.; Yao, Y.; Zhang, A.; Zhang, L.; et al. Pre-trained models: Past, present and future. AI Open 2021, 2, 225–250.
7. Eunice, J.; Popescu, D.E.; Chowdary, M.K.; Hemanth, J. Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications. Agronomy 2022, 12, 2395.
8. Thompson, N.C.; Greenewald, K.; Lee, K.; Manso, G.F. The computational limits of deep learning. arXiv 2020, arXiv:2007.05558.
9. Hossain, S.M.; Tanjil, M.; Morhsed, M.; Ali, M.A.B.; Islam, M.Z.; Islam, M.; Mobassirin, S.; Sarker, I.H.; Islam, S. Rice leaf diseases recognition using convolutional neural networks. In Advanced Data Mining and Applications, Proceedings of the 16th International Conference, ADMA 2020, Foshan, China, 12–14 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 299–314.
10. Chen, S.; Zhang, K.; Zhao, Y.; Sun, Y.; Ban, W.; Chen, Y.; Zhuang, H.; Zhang, X.; Liu, J.; Yang, T. An approach for rice bacterial leaf streak disease segmentation and disease severity estimation. Agriculture 2021, 11, 420.
11. Stephen, A.; Punitha, A.; Chandrasekar, A. Designing self attention-based ResNet architecture for rice leaf disease classification. Neural Comput. Appl. 2022, 35, 6737–6751.
12. 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.
13. Rice Leaf Diseases Dataset. Available online: https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases (accessed on 17 January 2023).
14. Rice Leaf Disease Image Samples. Available online: https://data.mendeley.com/datasets/fwcj7stb8r/1 (accessed on 17 January 2023).

15. Nanni, L.; Paci, M.; Brahnam, S.; Lumini, A. Comparison of different image data augmentation approaches. J. Imaging 2021, 7, 254.
16. Thangaraj, R.; Anandamurugan, S.; Kaliappan, V.K. Automated tomato leaf disease classification using transfer learning-based deep convolution neural network. J. Plant Dis. Prot. 2021, 128, 73–86. Agronomy 2023, 13, 961 24 of 24
17. Noor, A.; Zhao, Y.; Koubâa, A.; Wu, L.; Khan, R.; Abdalla, F.Y. Automated sheep facial expression classification using deep transfer learning. Comput. Electron. Agric. 2020, 175, 105528.
18. Kaya, A.; Keceli, A.S.; Catal, C.; Yalic, H.Y.; Temucin, H.; Tekinerdogan, B. Analysis of transfer learning for deep neural network based plant classification models. Comput. Electron. Agric. 2019, 158, 20–29.
19. Artzai Picona,, Maximiliam Seitzc,d, Aitor Alvarez-Gilaa, Patrick Mohnked, Amaia Ortiz-Barredob,Jone Echazarraa Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification, TECNALIA, Parque Tecnolgico de Bizkaia, C/ Geldo. Edificio 700, E-48160 Derio, Bizkaia, Spain Elsevier 201
20. Parul Sharmaa, Yash Paul Singh Berwal , Wiqas Ghai Performance analysis of deep learning CNN modelsfor disease detection in plants using image segmentations RIMT University, Mandi Gobindgarh 147301, , Haryana, India Elsevier 2018
21. Karthik R. Hariharan M. Sundar Anand , Priyanka Mathikshara , Annie Johnson , Menaka R. Attention embedded residual CNN for disease detection in tomato leaves hennai, India 1568-4946/ 2019 Elsevier
22. S. Ramesh , D. Vydeki Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm 25 March Elsevier 201
23. Barbedo, J., 2016. A review on the main challenges in automatic plant disease identification based on visible range images. Biosyst. Eng. 144, 52–60.
24. Barbedo, J., 2019. Plant disease identification from individual lesions and spots using deep learning. Biosyst. Eng. 180, 96–107.
25. Singh, J., Kaur, H., 2018. Plant disease detection based on region-based segmentation and KNN classifier. In: International Conference on ISMAC in Computational Vision and Bio-Engineering. Springer, Cham, pp. 1667–1675.
26. Naik, M., Sivappagari, C., 2016. Plant leaf and disease detection by using HSV features and SVM classifier. Int. J. Eng. Sci. 3794.
27. Chaudhary, A., Kolhe, S., Kamal, R., 2016. An improved random forest classifier for multi-class classification. Inform. Process. Agric. 3 (4), 215–222.
28. Kamilaris, A., Prenafeta-Boldú, F., 2018. Deep learning in agriculture: A survey. Comput. Electron. Agric. 147, 70–90.
29. Li, Y., Chao, X., 2020. ANN-Based Continual Classification in Agriculture. Agriculture 10 (5), 178.
30. Chen, J., Chen, J., Zhang, D., et al., 2020. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agric. 173, 105393.
31. Li, Y., Yang, J., 2020. Few-shot cotton pest recognition and terminal realization. Comput. Electron. Agric. 169, 105240.
32. Thenmozhi, K., Reddy, U.S., 2019. Crop pest classification based on deep convolutional neural network and transfer learning. Comput. Electron. Agric. 164, 104906.
33. Too, E., Yujian, L., Njuki, S., et al., 2019. A comparative study of fine-tuning deep learning models for plant disease identification. Comput. Electron. Agric. 161, 272–279.
34. de Castro, A., Madalozzo, G.A., dos Santos, T.N., et al., 2020. BerryIP embedded: An embedded vision system for strawberry crop. Comput. Electron. Agric. 173, 105354.
35. Tao, M., Ma, X., Huang, X., et al., 2020. Smartphone-based detection of leaf color levels in rice plants. Comput. Electron. Agric. 173, 105431.
36. Liakos, K.G., Busato, P., Moshou, D., et al., 2018. Machine learning in agriculture: A review. Sensors 18 (8), 2674.
37. Liang, X.W., Jiang, A.P., Li, T., et al., 2020. LR-SMOTE–An improved unbalanced data set oversampling based on K-means and SVM. Knowl.-Based Syst. 105845.
38. Qi Yanga , Liangsheng Shia,⁎ , Jingye Hana , Jin Yua , Kai Huangb A near real-time deep learning approach for detecting rice phenology based on UAV images China https://doi.org/10.1016/j.agrformet.2020.107938 Received 29 July 2019; Received in revised form 30 January 2020; Accepted 9 February 2020

39. Manoj Agrawal1 , Dr. Shweta Agrawal2 Rice Plant Diseases Detection & Classification Using Deep Learning Models: A Systematic Review Journal Of Critical Reviews Issn- 2394-5125 Vol 7, Issue 11, 2020
40. Chowdhury R. Rahman, et al. “Identification and Recognition of Rice Diseases and Pests Using Deep Convolutional Neural Networks.” , Biosystems Engineering, published by Elsevier Ltd., vol. 194, 112-120, 2020.
41. sethy, prabira Kumar (2020), “Rice Leaf Disease Image Samples”, Mendeley Data, V1, doi: 10.17632/fwcj7stb8r.1
42. Deepika Mandwariya, Dr. Varsha Jotwani (2024), Advanced Techniques for Rice Disease Detection: A Comprehensive Review, IOSR Journal of Computer Engineering (IOSR-JCE); e-ISSN: 2278-0661, Volume 26, Issue 1, Ser. 1 (Jan. – Feb. 2024), pp 31-38. doi: 10.9790/0661-2601013138.