

Article

Rice Leaf Disease Classification—A Comparative Approach Using Convolutional Neural Network (CNN), Cascading Autoencoder with Attention Residual U-Net (CAAR-U-Net), and MobileNet-V2 Architectures

Monoronjon Dutta ¹, Md Rashedul Islam Sujan ¹, Mayen Uddin Mojumdar ¹, Narayan Ranjan Chakraborty ¹, Ahmed Al Marouf ^{2,*}, Jon G. Rokne ² and Reda Alhajj ^{2,3,4}

¹ Multidisciplinary Action Research Laboratory, Department of Computer Science and Engineering, Daffodil International University, Birulia, Dhaka 1216, Bangladesh; monoronjon15-3660@diu.edu.bd (M.D.); rashedul15-3698@diu.edu.bd (M.R.I.S.); mayen.cse@diu.edu.bd (M.U.M.); narayan@daffodilvarsity.edu.bd (N.R.C.)

² Department of Computer Science, University of Calgary, Calgary, AB T2N 1N4, Canada; rokne@ucalgary.ca (J.G.R.); alhajj@ucalgary.ca (R.A.)

³ Department of Computer Engineering, Istanbul Medipol University, Istanbul 34810, Turkey

⁴ Department of Health Informatics, University of Southern Denmark, 5230 Odense, Denmark

* Correspondence: ahmedal.marouf@ucalgary.ca

Abstract: Classifying rice leaf diseases in agricultural technology helps to maintain crop health and to ensure a good yield. In this work, deep learning algorithms were, therefore, employed for the identification and classification of rice leaf diseases from images of crops in the field. The initial algorithmic phase involved image pre-processing of the crop images, using a bilateral filter to improve image quality. The effectiveness of this step was measured by using metrics like the Structural Similarity Index (SSIM) and the Peak Signal-to-Noise Ratio (PSNR). Following this, this work employed advanced neural network architectures for classification, including Cascading Autoencoder with Attention Residual U-Net (CAAR-U-Net), MobileNetV2, and Convolutional Neural Network (CNN). The proposed CNN model stood out, since it demonstrated exceptional performance in identifying rice leaf diseases, with test Accuracy of 98% and high Precision, Recall, and F1 scores. This result highlights that the proposed model is particularly well suited for rice leaf disease classification. The robustness of the proposed model was validated through k-fold cross-validation, confirming its generalizability and minimizing the risk of overfitting. This study not only focused on classifying rice leaf diseases but also has the potential to benefit farmers and the agricultural community greatly. This work highlights the advantages of custom CNN models for efficient and accurate rice leaf disease classification, paving the way for technology-driven advancements in farming practices.

Keywords: neural network architectures; rice leaf disease; cascading autoencoder with attention residual U-net (CAAR-U-Net); mobilenetv2; convolutional neural network (CNN)



Citation: Dutta, M.; Islam Sujan, M.R.; Mojumdar, M.U.; Chakraborty, N.R.; Marouf, A.A.; Rokne, J.G.; Alhajj, R. Rice Leaf Disease Classification—A Comparative Approach Using Convolutional Neural Network (CNN), Cascading Autoencoder with Attention Residual U-Net (CAAR-U-Net), and MobileNet-V2 Architectures. *Technologies* **2024**, *12*, 214. <https://doi.org/10.3390/technologies12110214>

Academic Editor: Pedro Antonio Gutiérrez

Received: 13 September 2024

Revised: 23 October 2024

Accepted: 25 October 2024

Published: 29 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Rice, a staple food for over half of the world's population, is grown in over 100 countries. Asia dominates cultivation, contributing to approximately 90% of global rice production [1]. This prominence underscores the need to effectively manage rice diseases, such as Bacterial Blight (BB), Blast (BL), Brown Spot (BS), and Tungro (TU). These diseases pose significant threats to yield and food security. Each rice leaf disease is caused by a different pathogen and can lead to considerable economic loss due to low yield and to nutritional deficit due to low quality of crop. This has the effect of potentially exacerbating food insecurity in vulnerable regions [2].

Bacterial Blight is a disease caused by *Xanthomonas oryzae* pv. *oryzae*. It is recognized for its significant impact on rice yields, contributing to 20–30% losses in many rice-growing

areas [3]. Similarly, Rice Blast, which is attributed to the pathogen *Magnaporthe oryzae*, presents a substantial threat to rice cultivation, potentially devastating entire fields if not adequately managed [4]. Brown Spot, historically linked to the Bengal Famine, is caused by the fungus *Cochliobolus miyabeanus*. It continues to be a concern, due to its resilience and widespread occurrence [5]. Tungro, a viral disease (Rice tungro bacilliform virus (RTBV) and Rice tungro spherical virus (RTSV)), further complicates rice cultivation, particularly in Southeast Asia [6,7].

Applying deep learning to disease detection and classification presents a transformative approach to these challenges. Researchers have applied many different methodologies for rice leaf disease classification, using different approaches. An example is T. Daniya and S. Vigneshwari et al. [8] who developed the RWW-NN algorithm for early detection of leaf diseases in rice plants. Their methodology encompassed an initial image pre-processing step utilizing histogram equalization and segmentation through SegNet. They then employed CNN features to enhance their disease-recognition capabilities. Based on the rice plant disease dataset, the RWW-NN model achieved remarkable performance metrics, such as high Accuracy (90.8%), F-measure (90.7%), Sensitivity (86.2%), and Specificity (94.7%) in identifying various diseases in rice. The study by P. Sobiya and colleagues [9] emphasized the significance of early disease detection in safeguarding paddy crops. Their study employed a machine learning methodology based on image processing for detecting and classifying diseases in rice (*Oryza sativa*). The dataset for this study consisted of images of diseased leaves and stems collected from paddy fields. The conditions examined were Rice Blast, Bacterial Leaf Blow, and Sheath Blow, as well as images of healthy leaves. Based on their experiments, they discovered that CNN was the most effective classifier, demonstrating an Accuracy rate of 93%.

L. Yang and team [10] focused on the critical task of the early identification and management of rice leaf diseases in natural environments. To tackle this challenge, they presented rE-GoogLeNet, a CNN model built upon the GoogLeNet architecture. Based on their findings, rE-GoogLeNet outperformed traditional and advanced multiscale models, achieving an impressive Accuracy rate of 99.58%, on average. This study showed a significant 1.72% improvement over the original GoogLeNet model, thus demonstrating the effectiveness of this model for rice leaf disease detection and control. According to the study conducted by J. Pan et al. [11], rice diseases can be identified in real time, using field images with complex backgrounds. In the initial phase of the research, the YoloX model demonstrated remarkable performance, with a mean Average Precision (mAP) of 95.58% in detecting rice disease images, surpassing other models' detection capabilities.

Regarding disease identification, the Siamese Network stands out, achieving 99.03% Accuracy and outperforming alternative models. Anshuman Nayak et al. [12] focused on plant disease early detection through mobile phones and cameras, enhanced with improved image acquisition modes (CNN). The study involved the application of various image segmentation techniques, such as foreground extraction, to pinpoint affected areas accurately. The top-performing models for image classification were DenseNet201 and MobileNetV2, with validation accuracies of 0.9718% and 0.9803%, respectively. In their study, Md Taimur Ahad et al. [13] conducted a performance comparison of various CNN paradigms, including aspects of transfer learning and ensemble models, for detecting and localizing rice diseases. Their study evaluated six CNN-based deep learning architectures, two of which were DenseNet121 and Inceptionv3. Moreover, they implemented transfer learning for several of these models and introduced a novel ensemble model. Their research indicated that the ensemble framework (DEX) achieved the highest Accuracy of 98%. Mainak Deb et al. [14] conducted a study in the agricultural domain, with the primary objective of plant disease precise detection for mitigating productivity loss. The study addressed paddy disease identification using CNNs. Five well-known models were used to evaluate classical CNN models: Inception-V3, VGG-16, AlexNet, MobileNet V2, and ResNet-18. Based on the experimental results, Inception-V3 was the best-performing CNN model, achieving an impressive 96.23% Accuracy rate.

An analysis of 5547 images was conducted by Swathy et al. [15], in order to classify two agricultural diseases, namely, Brown Spot and Leaf Blast. They assessed three machine learning models for the classification: a Random Forest Classifier, a CNN, and their purpose-built system. Based on their findings, the Random Forest Classifier achieved 92.77% disease detection Accuracy, the Neural Network achieved 96.27%, and their system reached 97%. Moreover, in the study they also analyzed Precision and Recall metrics for images for each disease. The Brown Spot scored 0.98 in Precision and 0.96 in Recall, while the Leaf Blast was rated 0.96 in Precision and 0.98 in Recall. Using 32 sophisticated pre-trained image analysis models, Meenakshi Aggarwal et al. [16] developed a system that effectively diagnoses agricultural diseases such as Bacterial Blight, Blast, and Brown Spot. They achieved 90–91% Accuracy, exceeding traditional diagnostic methods. After incorporating image-segmentation techniques into their methodology, the resulting Accuracy increased to 93–94%. Sharma et al. [17] confronted the issue of plant disease detection in remote regions, emphasizing the necessity for economical and reliable methods. Their research encompassed the analysis of image datasets, including 5932 rice and 1500 potato leaves affected by multiple diseases. They employed machine learning techniques such as CNN, SVM, KNN, DT, and RF. Remarkably, the CNN model excelled at identifying diseases in rice and potato leaves, registering identification Accuracy of 99.58% and 97.66%, respectively.

Singh et al. [18] employed a variety of models to detect diseases in rice, utilizing a dataset comprising 7332 images focusing on four specific types of disease. They used custom CNN architectures to improve disease detection and achieved 99.66% Accuracy with SGDM optimization. They improved this to even higher Accuracy of 99.83% when using the Adam optimizer [19].

Islam et al. [20] emphasized that the timely detection of rice diseases is crucial for protecting farmers from significant losses and suggested that machine learning models would be a possible solution for developing procedures to minimize these losses. Their research identified six rice diseases: Bacterial Leaf Blight, Leaf Blast, Tungro, Sheath Blight, Leaf Smut, and Brown Spot. Their team utilized three datasets and employed CNN with three different architectures—VGG, ResNet, and DenseNet—for disease detection. DenseNet121 turned out to be the most effective algorithm, achieving a notable Accuracy rate of 91.67%.

Dhar et al. [21] conducted a thorough analysis of a range of plant diseases, encompassing three diseases in apple leaves, one in cherry leaves, three in corn leaves, three in grape leaves, one in peach leaves, one in pepper bell leaves, two in potato leaves, one in strawberry leaves, four in rice leaves, and eight in tomato leaves, making up a total of 10 distinct plant diseases. Their research utilized three machine learning models: SVM, KNN, and AdaBoost. Notably, the KNN model emerged as the most accurate, particularly in diagnosing rice foliar diseases, achieving an impressive Accuracy rate of 99.6%.

Hassan et al. [22], in their study, emphasized the importance of early disease detection in plants. Their paper explored various machine learning models employed for detecting different diseases. The machine learning models included VGG16, VGG19, InceptionV3, ResNet50, and DenseNet201, which were carefully compared with the newly proposed Novel CNN model. Notably, the Novel CNN model achieved the highest Accuracy among the three datasets. Specifically, applying the model to a plant village dataset achieved Accuracy of 99.39%, for a rice disease dataset the Accuracy was 99.66%, and it was 76.59% for a cassava dataset.

Deep learning, a branch of machine learning, can be used to diagnose plant diseases from complex visual input [23,24]. This was done in study [25] and in similar studies like [26–28], which aspired to leverage deep learning techniques, such as CAAR-U-Net and MobileNet-V2, and, specifically, CNNs, to accurately classify and diagnose rice leaf diseases. In the last decade, CNNs have been increasingly employed in the plant phenotyping community. They have been very effective in modeling complicated concepts, owing to their ability to distinguish patterns and extract regularities from data. Examples range from variety identification in seeds [29] up to leaf water status information [30].

Several studies, including a smartphone-based citizen science tool for plant disease and insect pest identification [26], tongue disease prediction using machine learning [27], and an autoencoder approach for skeleton-based human activity recognition [28] have laid the foundation for using these deep learning architectures for similar problems. By doing a comparative analysis, our proposed study aimed to provide a robust, scalable, and efficient disease detection tool for farmers and agricultural professionals, thus contributing significantly to disease management and crop yield optimization. In particular, it focused on leveraging deep learning techniques for rice leaf disease classification. This approach will enhance the effectiveness of disease management strategies. The study combined technological advancements with agricultural practices, to classify the impact of disease on rice production. Its objective was to contribute to global food security.

In this paper, the contributions are as follows:

- Image pre-processing techniques—applying a bilateral filter and assessing image quality, using metrics such as SSIM and PSNR—were analyzed for use as crucial parts of a methodology for developing a robust implementation.
- Based on the results of the analysis, Convolutional Neural Network (CNN) was the most influential performer, exhibiting remarkable test Accuracy of 0.98 and high Precision, Recall, and F1 scores.
- As a result of the implementation of k-fold cross-validation ($k = 2, 4, 6$), the model is reliable and generalizable, effectively addressing concerns regarding overfitting and underfitting.

2. Materials and Methods

In the initial phase, data acquisition was initiated by gathering a dataset from the Kaggle website, a recognized online repository. Image processing, specifically utilizing a bilateral filter, was then applied, to improve its overall quality. Following this enhancement, a range of deep learning algorithms, namely, CAAR-U-Net, MobileNetV2, and CNN, were applied for thorough data analysis. Figure 1 depicts the comprehensive methodology employed in this study. Detailed insights into each facet of this methodology are provided in the following subsections.

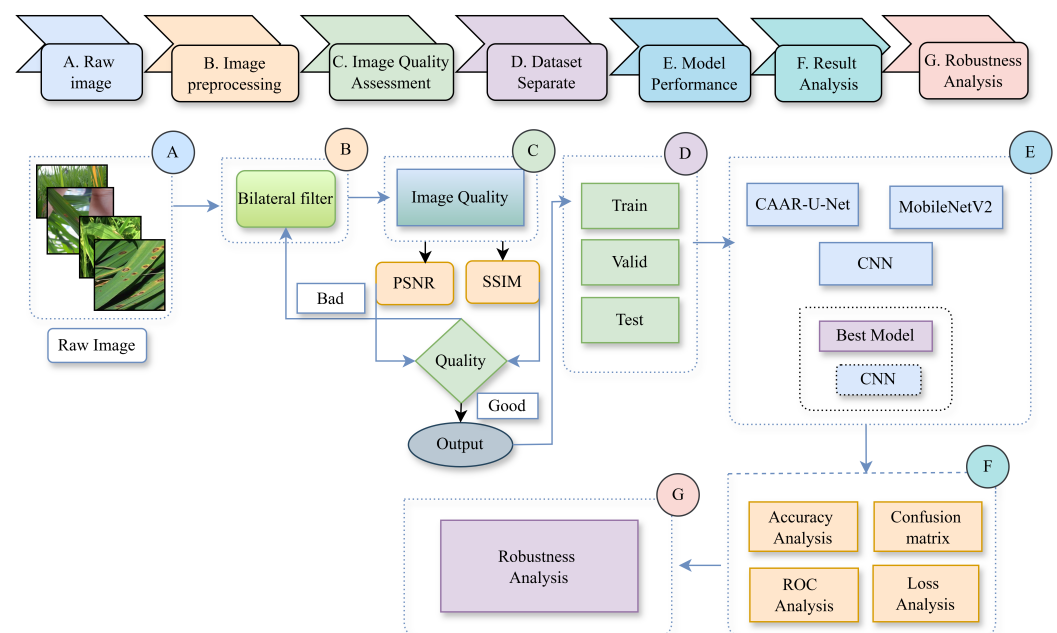


Figure 1. Proposed methodology for rice leaf disease classification.

2.1. Dataset Description

The dataset was collected from the Kaggle data repository [31], which has a total of 5932 images of rice leaf diseases. There were four classes: Bacterial Blight, Blast, Brown Spot, and Tungro. These classes contained 1584, 1440, 1600, and 1308 images, respectively. The images each had 300 * 300 pixels, and the image format was jpg [31] (see Figure 2 for the sample data).

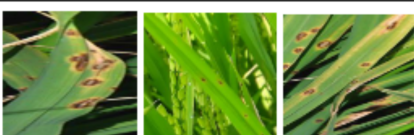
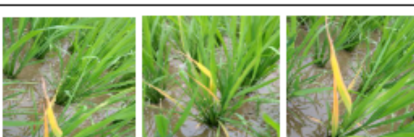
SN	Class Name	Description	Sample Images
1	Bacterial Blight	Dimension - 300*300	
		Formats - jpg	
		Quantity - 1584	
2	Blast	Dimension - 300*300	
		Formats - jpg	
		Quantity - 1440	
3	Brown Spot	Dimension - 300*300	
		Formats - jpg	
		Quantity - 1600	
4	Tungro	Dimension - 300*300	
		Formats - jpg	
		Quantity - 1308	

Figure 2. Sample of dataset.

As I was the set of all images in the dataset and C the set of class labels $C = \{C1, C2, C3, C4\}$ corresponding to Bacterial Blight, Blast, Brown Spot, and Tungro, respectively, then the dataset could be denoted by D , where $D \subseteq I * C$.

For each class C_i , there was a corresponding number of images n_i .

Each image $x \in I$ was a 300 * 300-pixel image in JPG format, which could be represented as a 3-dimensional array in the RGB color space: $x \in R^{300*300*3}$. The dataset could then be written as

$$D = \{(x_i, y_i) | x_i \in R^{300*300*3}, y_i \in C, i = 1 \dots N\} \quad (1)$$

2.2. Pre-Processing Steps

The bilateral filter, recognized as a non-linear, noise-reducing, edge-preserving image-smoothing filter, operated by substituting the intensity of each pixel for a weighted average of intensity values from adjacent pixels. The determination of this weight often relied on a Gaussian distribution. This process preserved edges by systematically choosing the weights to avoid blurring the edges. The bilateral filter [32] for each pixel P could be defined as

$$B(P) = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I(p) - I(q)|) \quad (2)$$

where $B(P)$ was the intensity of pixel P after applying the bilateral filter, S was the set of pixels in the neighborhood of pixels P , $I(p)$ was the intensity of the pixel P before filtering, $I(q)$ was the intensity of the pixel q , where q was a neighbor pixel of p , $G_{(\sigma_s)}$ was a spatial Gaussian that decreased the weight of distant pixels, with σ_s being the spatial

standard deviation, $G_{\sigma s}$ was a range Gaussian that reduced the weight of pixels with different intensity values $I(p)$, with σr being the range standard deviation, $\|p - q\|$ was the Euclidean distance [33] between pixels p and q , and W_p was a normalization factor $W_p \sum_{q \in S} G_{\sigma s}(\|p - q\|) G_{\sigma r}(|I(p) - I(q)|)$ to ensure that the weights summed up to 1. The parameters σs and σr corresponded (Table 1) to σ_{color} and σ_{space} , respectively, and the diameter of the pixel neighborhood was defined by the diameter parameter.

Table 1. The parameters and values of the image pre-processing.

Parameter	Value
Diameter	9
sigma_color	75
sigma_space	75

2.3. Image Resize

The images were resized to $128 * 128$ pixels, to create a uniform input shape for the neural network. For this, we let I_{orig} denote the set of original images and $I_{resized}$ the set of resized images. The resizing operation [34] could be represented as a function, $R : I_{orig} \rightarrow I_{resized}$ or applied as the resizing operation for each image as $R^{300*300*3} \rightarrow R^{128*128*3}$.

2.4. Image Quality Performance

The study assessed image qualities through the SSIM [35] and PSNR [36] metrics applied to each of the image sets. The results are shown in Table 2. The assessments were as follows: Image-01 (Bacterial Blight) had SSIM 0.9730 and PSNR 0.4254, indicating good quality; Image-02 (Blast) scored SSIM 0.9825 and PSNR 0.4560, also showing good quality; Image-03 (Brown Spot) had slightly lower SSIM 0.9389 and PSNR 0.3543, but quality remained satisfactory; Image-04 (Tungro) had SSIM 0.9210 and PSNR 0.4133, which were the weakest results among the image sets, but still satisfactory. Overall, pre-processing therefore maintained satisfactory image quality, which is crucial for accurate disease classification.

Table 2. The PSNR and SSIM values of the image quality are presented.

SL	SSIM	PSNR	Performance
image-01	0.9730	0.4254	The image quality was satisfactory.
image-02	0.9825	0.4560	The image quality was satisfactory.
image-03	0.9389	0.3543	The image quality was slightly reduced but acceptable.
image-04	0.9210	0.4133	The image quality was satisfactory.

2.5. Encoding

The class labels y_i were encoded into integer values by the encoding function $E : \{C_1, C_2, C_3, C_4\} \rightarrow \{0, 1, 2, 3\}$. The integer-encoded labels were then converted into a one-hot encoded format. We let $O : 0, 1, 2, 3 \rightarrow R^4$ be the one-hot encoding function [37], where each integer label was mapped to a vector R^4 . After encoding, the dataset was split into training, validation, and test sets. We let α be the proportion of the dataset used for training (70%) and $1 - \alpha$ be the proportion for validation and testing (30%). The validation and test sets were split equally, each receiving half of $1 - \alpha$ [38].

3. Model Architecture

The study applied various neural networks, such as Convolutional Neural Network (CNN), Cascading Autoencoder with Attention Residual U-Net (CAAR-U-Net), and MobileNet-V2 for disease classification.

3.1. Convolutional Neural Network (CNN)

The architecture of the described classification model was designed to analyze images of (128×128) pixels across three RGB channels, using a CNN approach. The model initiated with an input layer suited to the specified dimensions, followed by a convolutional layer with 32 filters with a (3×3) kernel size, employing Relu activation and using the same padding to preserve spatial dimensions. A subsequent max pooling layer with a (2×2) window reduced the spatial dimensions by half, acting as a downsampling approach for the feature maps. The model transitioned to a dense phase by flattening the 2D feature maps into a 1D feature vector, a critical step for moving from convolutional to dense layers. It included a dense layer with 128 units and Relu activation to add non-linearity and to support learning complex patterns. To mitigate overfitting, a dropout layer with a rate of 0.5 randomly deactivated specific input units during training [39]. The final part of the model consisted of a dense output layer aligned with the number of classes, using a softmax activation function to produce a probability distribution among the classes. The model employed a custom learning rate schedule starting at $1 \times e^{-3}$.

and adjusted based on the epoch number to enhance learning, alongside early stopping based on validation loss to avoid overfitting and model check-pointing to preserve the best model based on validation loss. The compilation process utilized the Adam optimizer [19] with a starting learning rate of $1 \times e^{-3}$ focusing on minimizing categorical cross-entropy Loss and maximizing Accuracy (see Figure 3) [40,41].

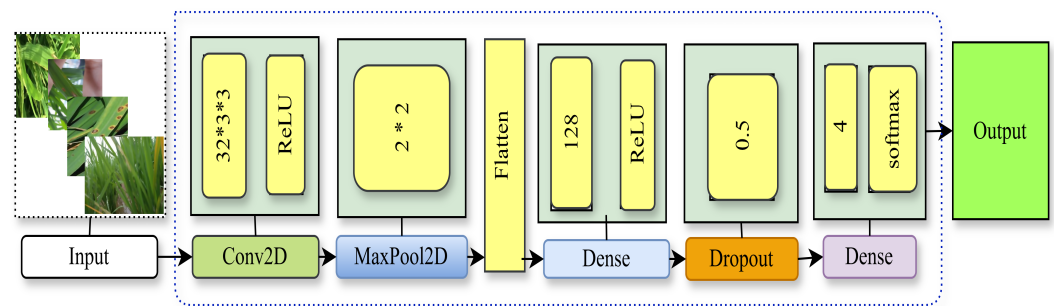


Figure 3. The CNN model architecture visualization.

3.2. Cascading Autoencoder with Attention Residual U-Net (CAAR-U-Net)

The CAAR-U-Net model, designed for image segmentation tasks, utilized a tailored architecture for processing 128×128 pixel images across three color channels, a standard setup in deep learning applications. The model's construction included convolutional, batch normalization, and pooling layers organized into an encoder–decoder framework. Initially, the encoder captured essential features through convolutional layers with 64 and then 128 filters, each followed by batch normalization and max pooling to ensure learning stability and reduce spatial dimensions. Subsequently, the decoder reversed this process, employing up-sampling layers to restore the feature maps to their original size, enabling precise segmentation. Moreover, the model's output was determined by a convolutional layer with three filters and a softmax activation function designed to classify each pixel into one of three categories, which was suitable for a four-class segmentation challenge. The compilation of the model used the Adam optimizer [19] and a mean squared error Loss function, complemented by Accuracy metrics and a custom learning rate schedule to optimize the training phase and enhance convergence. Furthermore, to combat overfitting and reduce generalization to new data, early stopping halted training when no improvement in validation loss was observed, while model check-pointing saved the weights of the best-performing instance. Hence, the CAAR-U-Net model encapsulated a comprehensive approach to image segmentation, integrating advanced features and mechanisms to improve Accuracy and model performance (see Figure 4) [42].

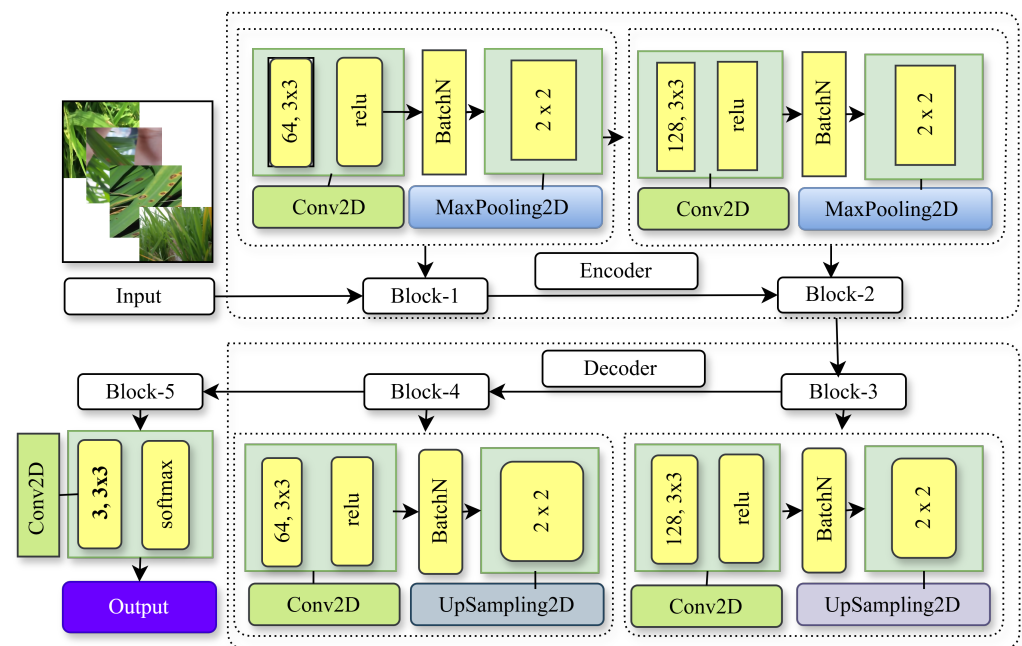


Figure 4. The CAAR-U-Net model architecture.

3.3. MobileNet-V2

A MobileNet-V2 model, pre-trained on ImageNet, was initially employed for transfer learning. The base model's architecture was retained, but its weights were frozen to prevent further training and preserve the pre-learned features [43]. Custom layers were added to the output of the base model. These layers included global average pooling, a dense layer with 512 units and ReLU activation, dropout with a rate of 0.4 for regularization, an additional dense layer with 256 units and ReLU activation, and, finally, an output layer with a softmax activation function, configured to have as many units as the specified number of classes [39,40,43,44]. The model was compiled with the Adam optimizer [19], a categorical cross-entropy Loss function (suitable for multi-class classification (4)), and Accuracy as the evaluation metric, to enable training the model for subsequent training on the dataset of interest with the specified optimization and Loss settings. The resulting model was a composite of the base MobileNetV2 architecture (see Figure 5) with custom layers, forming an end-to-end neural network for image classification [43].

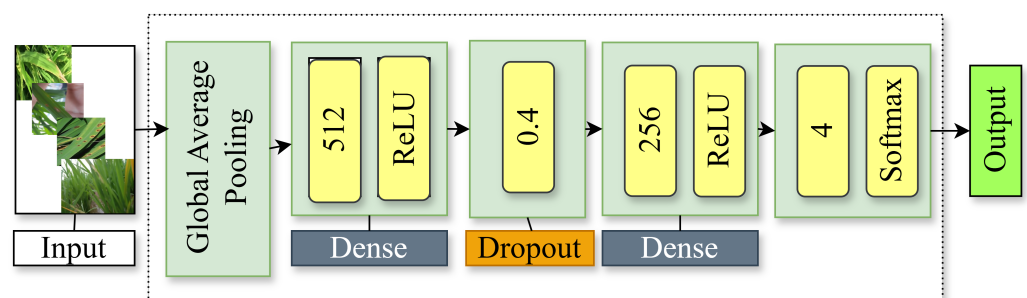


Figure 5. The MobileNetV2 model architecture.

4. Results

A Core i5-10210U CPU with a 1.60 GHz speed and graphics NVIDIA GeForce MX110 with 8GB RAM were used to implement the research described in this paper. The models were developed through Python programming libraries such as TensorFlow [45,46] and Keras [47]. The Precision, Recall, F1 score, and confusion matrix were calculated to evaluate the model performance, and the model's robustness was checked. In the study, we experimented

with various image quality assessment methods and Convolutional Neural Network (CNN) architectures. The study aimed to address the following research questions:

1. Is the image of acceptable quality after image pre-processing?
2. Can the CAAR-U-Net architecture segment images effectively?
3. Among the CNN models examined, which detects rice leaf disease classification better?
4. Does the final model exhibit any signs of overfitting or underfitting?

The performance matrices used to compare the proposed methods were Accuracy, Precision, Recall, F1 score, and ROC. Accuracy defines the proportion of correctly classified instances out of the total cases in the dataset [48]. Mathematically, it is calculated as

$$\text{Accuracy} = \text{Total Number of predictions} / \text{Number of correct predictions} \quad (3)$$

Precision quantifies the Accuracy of the model's positive predictions [49]. Mathematically, it is defined as the ratio of True Positive predictions, TP, to the total number of positive predictions TP + FP, where FP is False Positive predictions. The Precision [48] formula is, therefore,

$$\text{Precision} = TP / (TP + FP) \quad (4)$$

High Precision in a model indicates a low rate of False Positive errors. This suggests that it will likely be correct when the model predicts an instance as positive.

The Recall metric defines the model's ability to identify all positive instances [49]. It represents the proportion of true, accurate predictions relative to the number of positive instances. The formula for calculating Recall is as follows:

$$\text{Recall} = TP / (TP + FN) \quad (5)$$

where FN is the False Negative errors. High Recall means a low rate of False Negative errors, implying that the model successfully identifies the most positive instances.

The F1 score represents the harmonic mean of Precision and Recall [49]. It is a way to combine both metrics into a single, comprehensive metric. The F1 score [50] formula is

$$\text{F1-score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (6)$$

The F1 score is particularly useful when balancing Precision and Recall and there is uneven class distribution.

4.1. Performance Analysis

The study compared the performance metrics of the MobileNetV2 and CNN models, as outlined in Table 3. The CNN model consistently outperformed MobileNetV2 across all classes—Bacterial Blight, Blast, Brown Spot, and Tungro—in Accuracy (Equation (3)), Precision (Equation (4)), Recall (Equation (5)), and F1 measure (Equation (6)). While MobileNetV2 (MBV2) showed decent results, the CNN model demonstrated superior performance, even achieving perfect Recall in the Brown Spot class. The comprehensive data in Tables 3 and 4 highlight CNN's dominance in Accuracy and Loss during the training and validation phases of the study.

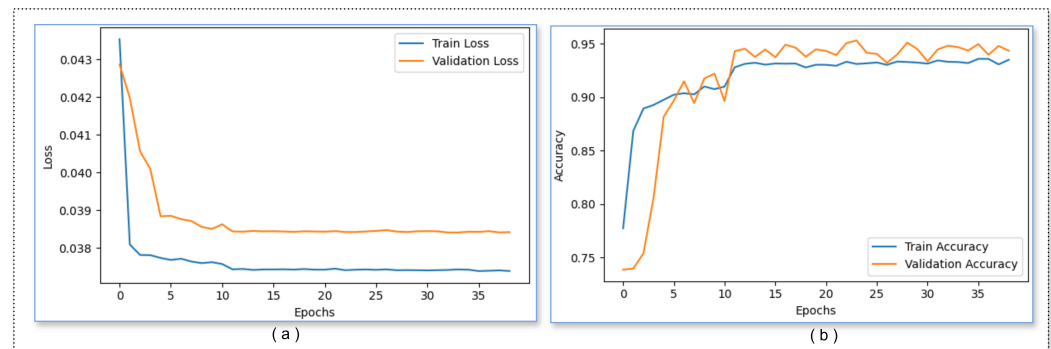
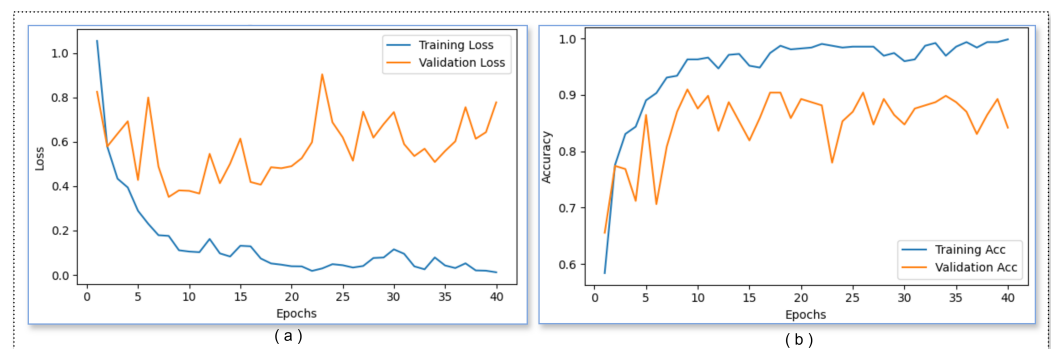
Table 3. Models performance analysis using Precision, Recall, and F1 measure.

Class	Precision		Recall		F1 Measure	
	MobileNetV2	CNN	MobileNetV2	CNN	MobileNetV2	CNN
Bacterial Blight	0.81	0.98	0.95	0.99	0.88	0.97
Blast	0.89	0.99	0.94	0.98	0.92	0.98
Brown Spot	0.86	0.97	0.79	1.0	0.83	0.99
Tungro	0.95	0.99	0.83	0.97	0.89	0.98

Table 4. Models performance analysis using Accuracy and Loss results.

Model	Accuracy			Loss	
	Accuracy	Training	Validation	Training	Validation
CAAR-U-Net	0.9541	0.9553	0.9562	0.0376	0.0372
MobileNetV2	0.8764	0.9983	0.8418	0.0044	0.7780
Convolutional Neural Network	0.9808	0.9992	0.9752	0.0095	0.0987

The results in Figure 6 prominently display the training and validation Accuracy curves and the Loss curve derived from the Segmentation model [51]. Figure 7a shows the Loss curve, while the x-axis corresponds to the total number of epochs. Figure 7b continues to provide insights into the model's performance, with the y-axis representing the Accuracy curve and the x-axis denoting the total number of epochs. In Figure 8a, the y-axis signifies the Loss curve, while the x-axis displays the total number of epochs. Conversely, in Figure 8b the y-axis corresponds to the Accuracy curve, and the x-axis indicates the total number of epochs. Collectively, these figures offer a comprehensive visual representation of the training and validation processes, providing valuable insights into the model's performance throughout successive epochs, as presented in the study.

**Figure 6.** The CAAR-U-Net model's (a) training and validation loss and (b) training and validation accuracy curves.**Figure 7.** The MobileNetV2 model's (a) training and validation loss and (b) training and validation accuracy curves.

In Figure 9, the confusion matrices [48,52] are compared for the two models, (a) CNN and (b) MobileNetV2, across four conditions. Model (a) shows high Accuracy, with 97% for 'Bacterial Blight', 98% for 'Blast', 100% for 'Brown Spot', and 97% for 'Tungro'. Model (b) has slightly lower Accuracy for 'Bacterial Blight' and 'Tungro' at 96%, a perfect score for 'Blast', but only 62% for 'Brown Spot'. Overall, model (a) demonstrated superior predictive Accuracy, as seen in its confusion matrix.

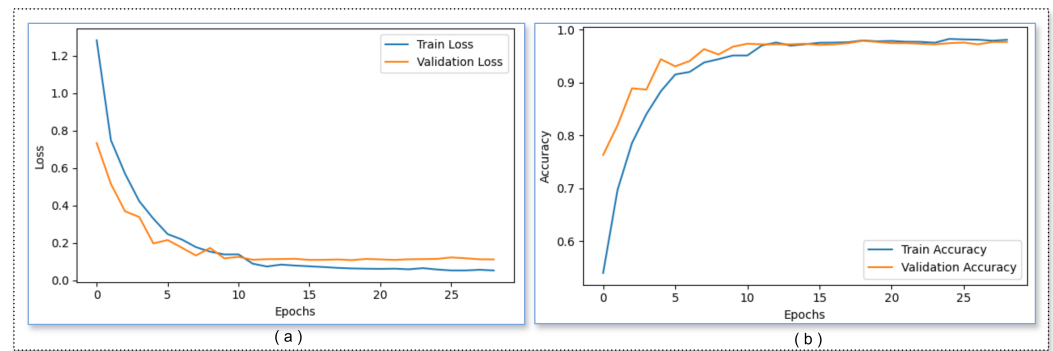


Figure 8. The CNN model's (a) training and validation loss and (b) training and validation accuracy curves.

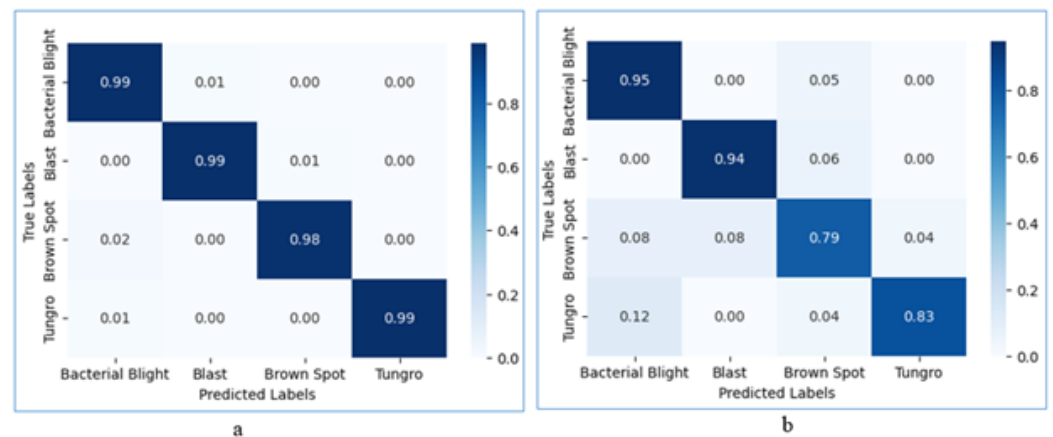


Figure 9. Confusion matrix for the CNN (a) and MobileNetV2 (b) models.

4.2. Receiver Operating Characteristics (ROC) Curve Analysis

The ROC curves analyzed in the study compared model performance for disease classification in various conditions [53,54]. The CNN and MobileNetV2 models showed high Accuracy with accurate ROC area scores of 1.0 for 'Bacterial Blight'. This performance level was consistent for diseases like 'Blast' and 'Brown Spot'. However, for 'Tungro', the CNN model scored perfectly at 1.0, while MobileNetV2 scored slightly less at 0.99. Overall, as shown in panel (a) of Figure 10, the CNN model demonstrated superior performance across the conditions tested, as confirmed by ROC analysis.

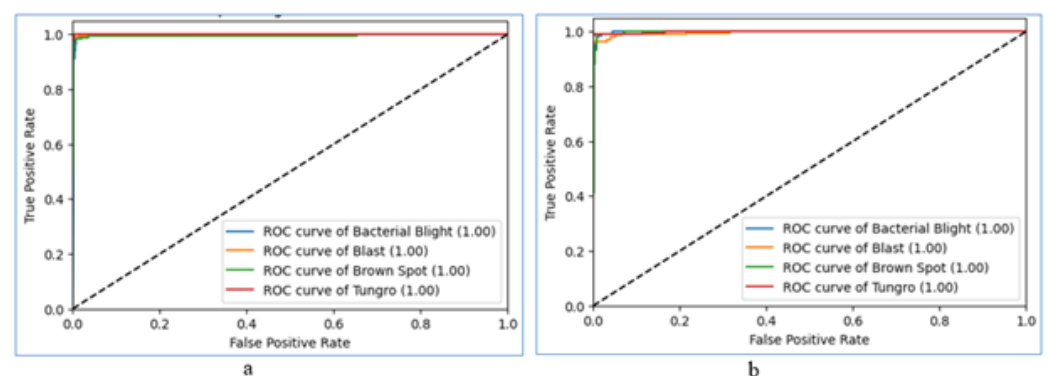


Figure 10. ROC for CNN (a) and MobileNetV2 (b) models.

4.3. Checking Robustness

The study investigated the balance between overfitting and underfitting in deep learning models by analyzing their performance with varying cross-validation folds [55]. It

observed high training Accuracy but lower validation Accuracy, suggesting a propensity for overfitting. The 2-fold, 4-fold, and 6-fold cross-validation analysis showed that the model performed well on training data but less on unused, unseen data (see Table 5). Increasing the cross-validation folds led to a minor improvement in generalization, indicating a slight reduction in overfitting.

Table 5. The average cross-validation result of the classification model.

Accuracy	2-Fold	4-Fold	6-Fold
Training	0.9987	0.9990	0.9983
Validation	0.9749	0.9753	0.9736

4.4. Comparative Analysis

The studies focused on different proposed architectures and their respective Accuracy in handling various classes. T. Daniya et al. [8] introduced an RWW and NN architecture in May 2023, achieving a 90% Accuracy rate across 3 classes. In contrast, L. Yang et al.'s [10] GoogLeNet architecture, introduced in May 2022 for 8 classes, reached the highest Accuracy of 99%, matched by J. Pan et al.'s [11] Siamese Network for 4 classes in December 2022. A notable trend was the diverse approach towards architecture design, ranging from CNN by P. Sobiya et al. [9] and Mainak Deb et al. [14], achieving 93% and 96% Accuracy, respectively, to more complex models like DenseNet201 by A. Nayak et al. [12] for 13 classes with 98% Accuracy, and an ensemble model by Md Taimur Ahad et al. [13] with 98% Accuracy for 9 classes. The proposed model employed a novel combination of CAAR-U-Net and CNN, attesting to the ongoing innovation in the field, achieving a 98% Accuracy rate for 4 classes, highlighting the continuous evolution and refinement of deep learning architectures for classification purposes (see Table 6).

On a commercial scale, evidently, a capital investment is initially required for adopting the employed approach [56,57]. Nevertheless, the wide-ranging large-scale commercial applications could provide high returns through considerable improvements in shortening process enhancement and cost reduction.

Table 6. Comparison of proposed methods with existing studies.

Author's	Publish Year	Proposed Architecture	Accuracy
T. Daniya and S. Vigneshwari et al. [8]	May 2023	RWW + NN	0.90
P. Sobiya et al. [9]	May 2022	CNN	0.93
L. Yang et al. [10]	May 2022	rE-GoogLeNet	0.99
J. Pan et al. [11]	Dec. 2022	Siamese Network	0.99
A. Nayak et al. [12]	Feb. 2023	DenseNet201,	0.98
Md Taimur Ahad et al. [13]	July 2023	Ensemble model (Densenet121, EfficientNetB7, and Xception)	0.98
Mainak Deb et al. [14]	October 2021	CNN	0.96
Proposed Model		CNN	0.98

5. Conclusions

This study presented an enhanced rice leaf disease identification model. The study involved the collection of images of infected rice leaves, encompassing diseases like Bacterial Blight, Blast, Brown Spot, and Tungro. These images were collected from sources such as Kaggle. Image pre-processing techniques were systematically applied to prepare the data, including bilateral filtering, resizing, encoding, and normalization. Approximately 70% of the dataset was used for training, 20% for validation, and 10% for testing. The subsequent

research phase utilized CAAR-U-Net for image segmentation, achieving an impressive 95.41% Accuracy rate. Further advancements were made by applying MobileNet-V2 and Convolutional Neural Network (CNN) to classification tasks. Notably, the CNN classification algorithm yielded outstanding results, attaining an Accuracy rate of 0.98%, and the model's robustness was checked, using K-fold cross-validation. This research underscores the effectiveness of the proposed approach in accurately identifying and classifying rice leaf diseases. It also demonstrates its potential for practical applications in agriculture and disease management.

However, acknowledging the study's limitations is essential. One key area for improvement is the size and diversity of the dataset utilized. In our future work, we will prioritize comprising larger and more diverse datasets. This approach will significantly enhance the model's generalizability and robustness, allowing us to better address a wider range of rice leaf disease cases and other crops.

Additionally, the model's practical applications go beyond classifying rice leaf diseases. On a commercial level, this approach could be incorporated into mobile applications, enabling farmers to diagnose plant diseases in real time and reducing the reliance on expert diagnoses. Furthermore, the model could be integrated into IoT devices or drones for automated field monitoring, offering real-time disease detection and facilitating timely interventions in crop management.

This study not only advances rice leaf disease classification but also holds the potential to significantly benefit the agricultural community. By improving the accuracy and efficiency of disease diagnosis, the proposed model can contribute to better crop management, ultimately enhancing yields and reducing the economic impact of crop losses.

Author Contributions: Conceptualization, M.D., M.R.I.S. and M.U.M.; methodology, M.D., M.R.I.S. and M.U.M.; software, M.D. and M.R.I.S.; validation, M.U.M., N.R.C. and A.A.M.; formal analysis, M.D., M.R.I.S.; investigation, M.U.M., N.R.C. and A.A.M.; resources, M.U.M., N.R.C. and A.A.M.; data curation, M.D. and M.R.I.S.; writing—original draft preparation, M.D. and M.R.I.S.; writing—review and editing, A.A.M. and J.G.R.; visualization, M.D. and M.R.I.S.; supervision, M.U.M., N.R.C. and R.A.; project administration, J.G.R. and R.A.; funding acquisition, R.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available from the corresponding author and can be shared with anyone upon reasonable request.

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

References

1. Fukagawa, N.K.; Ziska, L.H. Rice: Importance for Global Nutrition. *J. Nutr. Sci. Vitaminol.* **2019**, *65*, S2–S3. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Khush, G.S. What it will take to Feed 5.0 Billion Rice consumers in 2030. *Plant Mol. Biol.* **2005**, *59*, 1–6. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Niño-Liu, D.O.; Ronald, P.C.; Bogdanove, A.J. Xanthomonas oryzae pathovars: Model pathogens of a model crop. *Mol. Plant Pathol.* **2006**, *7*, 303–324. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Dean, R.; Van Kan, J.A.; Pretorius, Z.A.; Hammond-Kosack, K.E.; Di Pietro, A.; Spanu, P.D.; Rudd, J.J.; Dickman, M.; Kahmann, R.; Ellis, J.; et al. The Top 10 fungal pathogens in molecular plant pathology. *Mol. Plant Pathol.* **2012**, *13*, 414–430. [\[CrossRef\]](#)
5. Talbot, N.J. On the Trail of a Cereal Killer: Exploring the Biology of Magnaporthe grisea. *Annu. Rev. Microbiol.* **2003**, *57*, 177–202. [\[CrossRef\]](#)
6. Pangga, I.B.; Cruz, F.C.S. Rice tungro disease. In *Viral Diseases of Field and Horticultural Crops*; Elsevier: Amsterdam, The Netherlands, 2024; pp. 81–86. [\[CrossRef\]](#)
7. Dahal, G.; Druka, A.; Burns, T.M.; Villegas, L.C.; Fan, Z.; Shrestha, R.A.; Hull, R. Some biological and genomic properties of rice tungro bacilliform badnavirus and rice tungro spherical waikavirus from Nepal. *Ann. Appl. Biol.* **1996**, *129*, 267–287. [\[CrossRef\]](#)
8. Daniya, T.; Vigneshwari, S. Rider Water Wave-enabled deep learning for disease detection in rice plant. *Adv. Eng. Softw.* **2023**, *182*, 103472. [\[CrossRef\]](#)

9. Sobiyaa, P.; Jayareka, K.S.; Maheshkumar, K.; Naveena, S.; Rao, K.S. Paddy disease classification using machine learning technique. *Mater. Today Proc.* **2022**, *64*, 883–887. [\[CrossRef\]](#)
10. Yang, L.; Yu, X.; Zhang, S.; Long, H.; Zhang, H.; Xu, S.; Liao, Y. GoogLeNet based on residual network and attention mechanism identification of rice leaf diseases. *Comput. Electron. Agric.* **2023**, *204*, 107543. [\[CrossRef\]](#)
11. Pan, J.; Wang, T.; Wu, Q. RiceNet: A two stage machine learning method for rice disease identification. *Biosyst. Eng.* **2023**, *225*, 25–40. [\[CrossRef\]](#)
12. Nayak, A.; Chakraborty, S.; Swain, D.K. Application of smartphone-image processing and transfer learning for rice disease and nutrient deficiency detection. *Smart Agric. Technol.* **2023**, *4*, 100195. [\[CrossRef\]](#)
13. Ahad, M.T.; Li, Y.; Song, B.; Bhuiyan, T. Comparison of CNN-based deep learning architectures for rice diseases classification. *Artif. Intell. Agric.* **2023**, *9*, 22–35. [\[CrossRef\]](#)
14. Deb, M.; Dhal, K.G.; Mondal, R.; Gálvez, J. Paddy Disease Classification Study: A Deep Convolutional Neural Network Approach. *Opt. Mem. Neural Netw.* **2021**, *30*, 338–357. [\[CrossRef\]](#)
15. Swathy, K.; Anish Babu, K.K. Classification of Paddy Diseases Using Ellipse Fitting. In Proceedings of the 2022 Second International Conference on Next Generation Intelligent Systems (ICNGIS), Kottayam, India, 29–31 July 2022. [\[CrossRef\]](#)
16. 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. [\[CrossRef\]](#)
17. Sharma, R.; Singh, A.; Kavita; Jhanjhi, N.Z.; Masud, M.; Sami, Jaha, E.; Verma, S. Plant Disease Diagnosis and Image Classification Using Deep Learning. *Comput. Mater. Contin.* **2022**, *71*, 2125–2140. [\[CrossRef\]](#)
18. Singh, S.P.; Pritamdas, K.; Devi, K.J.; Devi, S.D. Custom Convolutional Neural Network for Detection and Classification of Rice Plant Diseases. *Procedia Comput. Sci.* **2023**, *218*, 2026–2040. [\[CrossRef\]](#)
19. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
20. Islam, A.; Islam, R.; Haque, S.M.R.; Islam, S.M.M.; Khan, M.A.I. Rice Leaf Disease Recognition using Local Threshold Based Segmentation and Deep CNN. *Int. J. Intell. Syst. Appl.* **2021**, *13*, 35–45. [\[CrossRef\]](#)
21. Dhar, P.; Rahman, M.S.; Abedin, Z. Classification of Leaf Disease Using Global and Local Features. *Int. J. Inf. Technol. Comput. Sci.* **2022**, *14*, 43–57. [\[CrossRef\]](#)
22. Hassan, S.M.; Maji, A.K. Plant Disease Identification Using a Novel Convolutional Neural Network. *IEEE Access* **2022**, *10*, 5390–5401. [\[CrossRef\]](#)
23. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Kansizoglou, I.; Bampis, L.; Gasteratos, A. Deep Feature Space: A Geometrical Perspective. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6823–6838. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2012**, *60*, 84–90. [\[CrossRef\]](#)
26. Christakakis, P.; Papadopoulou, G.; Mikos, G.; Kalogiannidis, N.; Ioannidis, D.; Tzovaras, D.; Pechlivani, E.M. Smartphone-Based Citizen Science Tool for Plant Disease and Insect Pest Detection Using Artificial Intelligence. *Technologies* **2024**, *12*, 101. [\[CrossRef\]](#)
27. Hassoon, A.R.; Al-Naji, A.; Khalid, G.A.; Chahl, J. Tongue disease prediction based on machine learning algorithms. *Technologies* **2024**, *12*, 97. [\[CrossRef\]](#)
28. Hossen, M.A.; Naim, A.G.; Abas, P.E. Deep Learning for Skeleton-Based Human Activity Segmentation: An Autoencoder Approach. *Technologies* **2024**, *12*, 96. [\[CrossRef\]](#)
29. Taheri-Garavand, A.; Nasiri, A.; Fanourakis, D.; Fatahi, S.; Omid, M.; Nikoloudakis, N. Automated in situ seed variety identification via deep learning: A case study in chickpea. *Plants* **2021**, *10*, 1406. [\[CrossRef\]](#)
30. Fanourakis, D.; Papadakis, V.M.; Machado, M.; Psyllakis, E.; Nektarios, P.A. Non-invasive leaf hydration status determination through convolutional neural networks based on multispectral images in chrysanthemum. *Plant Growth Regul.* **2024**, *102*, 485–496. [\[CrossRef\]](#)
31. Rice Leafs Disease Dataset [Online]. Available online: <https://www.kaggle.com/datasets/maimunulkjisan/rice-leaf-dataset-from-mendeley-data> (accessed on 22 July 2024).
32. Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, 7 January 1998; pp. 839–846. Available online: <https://ieeexplore.ieee.org/document/710815> (accessed on 3 March 2022).
33. Gower, J.C. Properties of Euclidean and non-Euclidean distance matrices. *Linear Algebra Its Appl.* **1985**, *67*, 81–97. [\[CrossRef\]](#)
34. Upadhyay, S.K.; Kumar, A. A novel approach for rice plant diseases classification with deep convolutional neural network. *Int. J. Inf. Technol.* **2022**, *14*, 185–199.
35. Brunet, D.; Vrscay, E.R.; Wang, Z. On the Mathematical Properties of the Structural Similarity Index. *IEEE Trans. Image Process.* **2012**, *21*, 1488–1499. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Wang, Y.; Li, J.; Lv, Y.; Yao, F.; Jiang, Q. Image quality evaluation based on image weighted separating block peak signal to noise ratio. In Proceedings of the International Conference on Neural Networks and Signal Processing, Nanjing, China, 14–17 December 2003.
37. Mohapatra, S.; Marandi, C.; Sahoo, A.; Mohanty, S.; Tudu, K. Rice Leaf Disease Detection and Classification Using a Deep Neural Network. In *Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2022; pp. 231–243. [\[CrossRef\]](#)

38. Ritharson, P.I.; Raimond, K.; Mary, X.A.; Robert, J.E.; Andrew, J. DeepRice: A deep learning and deep feature based classification of Rice leaf disease subtypes. *Artif. Intell. Agric.* **2024**, *11*, 34–49.
39. Garbin, C.; Zhu, X.; Marques, O. Dropout vs. batch normalization: An empirical study of their impact to deep learning. *Multimed. Tools Appl.* **2020**, *79*, 12777–12815. [\[CrossRef\]](#)
40. Kabani, A.; El-Sakka, M.R. Object Detection and Localization Using Deep Convolutional Networks with Softmax Activation and Multi-class Log Loss. In *Image Analysis and Recognition; Lecture Notes in Computer Science*; Springer: Cham, Switzerland, 2016; pp. 358–366. [\[CrossRef\]](#)
41. O'shea, K.; Nash, R. An Introduction to Convolutional Neural Networks. *arXiv* **2015**, arXiv:1511.08458. Available online: <https://arxiv.org/pdf/1511.08458.pdf> (accessed on 24 October 2024).
42. Abinaya, S.; Kumar, K.U.; Alphonse, A.S. Cascading Autoencoder with Attention Residual U-Net for Multi-Class Plant Leaf Disease Segmentation and Classification. *IEEE Access* **2023**, *11*, 98153–98170. [\[CrossRef\]](#)
43. MobileNetV2 Model for Image Classification [Online]. Available online: <https://ieeexplore.ieee.org/abstract/document/9422058> (accessed on 24 October 2024).
44. Agarap, A.F. Deep Learning using Rectified Linear Units (ReLU). *arXiv* **2018**, arXiv:1803.08375. Available online: <https://arxiv.org/pdf/1803.08375.pdf> (accessed on 24 October 2024).
45. TensorFlow. TensorFlow [Online]. TensorFlow. Google. 2019. Available online: <https://www.tensorflow.org/> (accessed on 24 October 2024).
46. Abadi, M. TensorFlow: Learning functions at scale. In Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, Nara, Japan, 18–24 September 2016; Volume 51, p. 1.
47. Ketkar, N. Introduction to Keras. In *Deep Learning with Python*; Apress: Berkeley, CA, USA, 2017; pp. 97–111.
48. Mojumdar, M.U.; Chakraborty, N.R. Orange & Orange leaves diseases detection using Computerized Techniques. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021. [\[CrossRef\]](#)
49. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
50. Naveen, A.; Manoj, B.; Akhila, G.; Nakarani, M.B.; Rathna Sreekar, J.; Beriwal, P.; Gupta, N.; Narayanan, S.J. Deep learning techniques for detection of COVID-19 using chest x-rays. *Adv. Syst. Sci. Appl.* **2021**, *21*, 42–57. [\[CrossRef\]](#)
51. Ahmed, T.U.; Jamil, M.N.; Hossain, M.S.; Andersson, K.; Hossain, M.S. An Integrated Real-Time Deep Learning and Belief Rule Base Intelligent System to Assess Facial Expression Under Uncertainty. In Proceedings of the 2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Kitakyushu, Japan, 26–29 August 2020. [\[CrossRef\]](#)
52. Marom, N.D.; Rokach, L.; Shmilovici, A. Using the confusion matrix for improving ensemble classifiers. In Proceedings of the 2010 IEEE 26-th Convention of Electrical and Electronics Engineers in Israel, Eilat, Israel, 17–20 November 2010. Available online: <https://ieeexplore.ieee.org/abstract/document/5662159> (accessed on 24 October 2024).
53. Kumar, R.; Indrayan, A. Receiver operating characteristic (ROC) curve for medical researchers. *Indian Pediatr.* **2011**, *48*, 277–287. [\[CrossRef\]](#)
54. Nainwal, A.; Malik, G.K.; Jangra, A. Convolution neural network based COVID-19 screening model. *Adv. Syst. Sci. Appl.* **2021**, *21*, 31–39. [\[CrossRef\]](#)
55. Rodriguez, J.D.; Perez, A.; Lozano, J.A. Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 569–575. [\[CrossRef\]](#) [\[PubMed\]](#)
56. Hridoy, R.H.; Arni, A.D.; Haque, A. Improved vision-based diagnosis of multi-plant disease using an ensemble of deep learning methods. *Int. J. Power Electron. Drive Syst.* **2023**, *13*, 5109–5117. [\[CrossRef\]](#)
57. Taheri-Garavand, A.; Mumivand, H.; Fanourakis, D.; Fatahi, S.; Taghipour, S. An artificial neural network approach for non-invasive estimation of essential oil content and composition through considering drying processing factors: A case study in *Mentha aquatica*. *Ind. Crops Prod.* **2021**, *171*, 113985. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.