



Article

Classification of Typical Pests and Diseases of Rice Based on the ECA Attention Mechanism

Hongjun Ni ¹, Zhiwei Shi ¹, Stephen Karungaru ², Shuaishuai Lv ^{1,*}, Xiaoyuan Li ¹, Xingxing Wang ¹ 
and Jiaqiao Zhang ³ 

¹ School of Mechanical Engineering, Nantong University, Nantong 226019, China; ni.hj@ntu.edu.cn (H.N.); 2109310033@stmail.ntu.edu.cn (Z.S.); 2210310046@stmail.ntu.edu.cn (X.L.); wangxx@ntu.edu.cn (X.W.)

² Graduate School of Advanced Technology and Science, Tokushima University, Tokushima 770-8506, Japan; karungaru@tokushima-u.ac.jp

³ School of Mechanical Engineering, Southeast University, Nanjing 211189, China; 230218554@seu.edu.cn

* Correspondence: lvshuaishuai@ntu.edu.cn

Abstract: Rice, a staple food crop worldwide, is pivotal in agricultural productivity and public health. Automatic classification of typical rice pests and diseases is crucial for optimizing rice yield and quality in practical production. However, infrequent occurrences of specific pests and diseases lead to uneven dataset samples and similar early-stage symptoms, posing challenges for effective identification methods. In this study, we employ four image enhancement techniques—flipping, modifying saturation, modifying contrast, and adding blur—to balance dataset samples throughout the classification process. Simultaneously, we enhance the basic *RepVGG* model by incorporating the ECA attention mechanism within the Block and after the Head, resulting in the proposal of a new classification model, *RepVGG_ECA*. The model successfully classifies six categories: five types of typical pests and diseases, along with healthy rice plants, achieving a classification accuracy of 97.06%, outperforming *ResNet34*, *ResNeXt50*, *ShuffleNet V2*, and the basic *RepVGG* by 1.85%, 1.18%, 3.39%, and 1.09%, respectively. Furthermore, the ablation study demonstrates that optimal classification results are attained by integrating the ECA attention mechanism after the Head and within the Block of *RepVGG*. As a result, the classification method presented in this study provides a valuable reference for identifying typical rice pests and diseases.

Keywords: rice; pest and disease classification; ECA; attention mechanism; deep learning



Citation: Ni, H.; Shi, Z.; Karungaru, S.; Lv, S.; Li, X.; Wang, X.; Zhang, J. Classification of Typical Pests and Diseases of Rice Based on the ECA Attention Mechanism. *Agriculture* **2023**, *13*, 1066. <https://doi.org/10.3390/agriculture13051066>

Academic Editor: Maciej Zaborowicz

Received: 29 March 2023

Revised: 11 May 2023

Accepted: 15 May 2023

Published: 16 May 2023



Copyright: © 2023 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

As global economic development progresses and living standards rise, the demand for high-quality rice crops grows [1]. However, rice production faces numerous challenges from diseases caused by bacteria, fungi, and other organisms, such as blasts, brown spots, and dead hearts, which can significantly affect crop health, yield, and quality. Failure to detect and control these pests and diseases could lead to substantial economic losses in agricultural production [2–4]. Thus, farmers and the agricultural sector must develop effective scientific methods for classifying and detecting rice pests and diseases.

Historically, the classification and detection of rice pests and diseases have relied on manual visual inspection, which is time-consuming, inefficient, and inaccurate [5]. Such ineffective traditional manual inspection methods inhibit sufficient rice production from meeting societal needs, necessitating the development of novel detection techniques.

Advancements in computer technology have given rise to traditional image-processing techniques, including image pre-processing, feature extraction, dimensionality reduction, and recognition. These methods have been applied to agricultural images for various purposes, for example, classifying crop pest and disease locations and types [6–8]. Nan Xu [9] utilized image processing and traditional machine recognition techniques for crop pest detection and analyzed their effectiveness. The results show that the recognition rates

of the method are 86%, 89%, 91%, 83%, 78%, and 79%, respectively. Azim et al. [10] propose segmentation of disease-affected areas using hue thresholding by background saturation threshold removal of the image. Local and global statistical information for such images is reliably described. An accuracy of 86.58% was achieved on UCI's rice leaf disease dataset. Shrivastava et al. [11] explored 14 color spaces, with four features extracted from each color channel, yielding 172 features. The performance of seven different classifiers was also compared, demonstrating a classification accuracy of up to 94.65% using a Support Vector Machine (SVM) classifier. The method has high accuracy and is easy to implement. However, these approaches still fall short of societal needs due to poor migration ability between rice pests and diseases and suboptimal recognition results.

Deep learning-based classification and target recognition algorithms have recently emerged in mainstream research, replacing traditional image recognition techniques [12–14]. These algorithms allow machines to adaptively learn image features without manual feature extraction, thereby enabling efficient completion of classification and detection tasks. Deep convolutional networks have effectively improved migration between rice pests and diseases [15]. Sagarika et al. [16] used a CNN model to build a classification detection system for three types of rice pests and diseases with an accuracy of 94.12%. Wang Yibin et al. [17] proposed an attention-based deep separable network model to achieve classification detection for four rice pest types with an accuracy of 94.65%. Patil et al. [18] proposed a novel multimodal data fusion framework called Rice-Fusion to diagnose rice diseases, which fuses agrometeorological data extracted from sensors with images captured by cameras to provide a single output for diagnosing rice diseases. Analysis of the experimental results shows that the proposed Rice-Fusion multimodal data fusion framework outperforms the unimodal framework. Liang K et al. [19] reduced the model parameters by decreasing the number of channels and introducing depth-separable convolution based on the VGG16 model. A lightweight model was also constructed by introducing the SE attention mechanism. The results show that the model achieves a classification accuracy of 95.09%. Yi Lu et al. [20] first preprocessed the images using median filtering and histogram equalization. They performed edge segmentation using the Sobel operator to extract parameters based on color and texture features. Subsequently, a BP neural network was built to train and test the pre-processed images. A recognition accuracy of 85.8% was achieved. Burhan et al. [21] conducted a comparative study of classification performance using five different models on their rice dataset after pre-processing to remove background and shadows, and the results showed that ResNet101 V2 was the best-performing model with an accuracy of 86.799%. Hu Y et al. [22] presented an advanced YOLO-GBS model for accurate rice pest detection. The model achieved a mean average precision of 79.8% on a self-made rice pest dataset, outperforming the original YOLOv5s by 5.4%.

Additionally, it demonstrated robustness and generalizability on larger-scale pest datasets, suggesting potential for broader crop pest detection applications. Chowdhury R. Rahman et al. [23] employed large-scale architectures, including VGG16 and InceptionV3, and a two-stage small CNN architecture for detecting diseases and pests from rice plant images and achieved a notable accuracy of 93.3% with the proposed small CNN model on a dataset of 1426 images. Dengshan Li et al. [24] developed a deep learning-based video detection architecture to identify three rice pests and diseases: rice blight, rice brown spot and rice stem borer symptoms. The researchers' customized backbone network outperforms other models such as VGG16, ResNet-50, ResNet-101 and YOLOv3, particularly in detecting slightly blurred images, demonstrating its potential for wider application in crop pest and disease detection. Finally, Zhiyong Li et al. [25] demonstrated the effectiveness of transfer learning and ARGAN data augmentation for rice pest identification. VGG16 achieved the highest accuracy among the three classification networks, with 84.39% on the original dataset and 88.68% on the augmented dataset. In this paper, we address the issue of training sample imbalance during classification by applying image augmentation techniques, including flipping, modifying saturation and contrast, and adding a blur. We employ a *RepVGG_ECA* model with an integrated attention mechanism to classify six rice images, achieving a classification accuracy of 97.06%. This method offers valuable insights for classifying typical rice pests and diseases.

The remainder of the paper is structured as follows: Section 2 describes the experimental methods, Section 3 presents the experimental results, Section 4 discusses the proposed method based on the experimental results, and Section 5 offers concluding remarks.

2. Methods

2.1. RepVGG

In 2014, the Visual Geometry Group at the University of Oxford introduced the VGG model [26], which exhibited strong performance in various computer vision tasks. The primary attribute of the VGG model is its depth, with networks typically comprising 16 (VGG-16) or 19 (VGG-19) successive convolutional and fully connected layers. While this allows the network to learn more complex features, the increased depth also exacerbates the vanishing gradient problem during training, leading to higher computational complexity and a more significant number of parameters. In addition, the fully connected layers, in particular, require substantial computational resources and training time. Consequently, VGG networks have been progressively superseded by more advanced network architectures.

In 2021, Ding et al. [27] drew inspiration from residual structures and proposed the RepVGG [28] model. The defining characteristic of residual structures is their skip connections, where the input is directly added to the output. This design mitigates the vanishing gradient problem, enabling deeper network training. RepVGG, in turn, features a multi-branch structure akin to residual structures during the training phase. Specifically, RepVGG's multi-branch structure comprises a 1×1 convolutional branch, a 3×3 convolutional branch, and a constant mapping branch, as illustrated in Figure 1. For example, given input feature map X , the 3×3 convolutional branch output is A , while the 1×1 convolutional branch output is B . The constant mapping branch output is C . The output of the fundamental building block can then be expressed as:

$$Y = A + B + C = BN(W_{3 \times 3}(X)) + BN(W_{1 \times 1}(X)) + BN(X), \quad (1)$$

where $W_{3 \times 3}$ and $W_{1 \times 1}$ denote the 3×3 and 1×1 weight matrices. BN denotes the batch normalization operation. The main role of the BN layer is to normalize the output of the convolutional layer. Specifically, the BN layer calculates the mean and variance of the output of the convolutional layer and normalizes the output. Let the desired mean be μ , and the variance be σ^2 . The BN layer is given by:

$$BN(x) = \gamma \times \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \quad (2)$$

where γ and β are learnable parameters to control the scaling and translation of the output features, respectively. x is a sample of the output of the convolution layer. ϵ is a small value to maintain numerical stability and is usually set to e^{-5} .

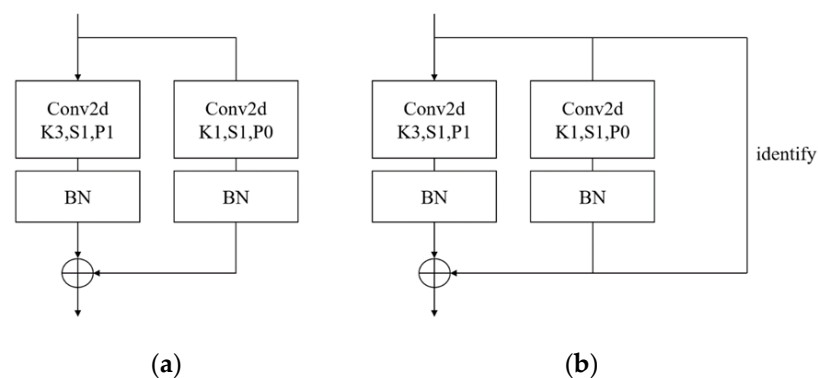


Figure 1. Structure of the RepVGG block: (a) RepVGG block A; (b) RepVGG block B.

This can effectively reduce internal covariate bias, improve training stability, and speed up the convergence process while helping to reduce the sensitivity of the model to parameter initialization. In the *RepVGG* block, the BN layer follows the convolutional layer, and this design also helps to improve the training effect and inference performance of the model. Figure 2a below shows the structure of the model during the training process.

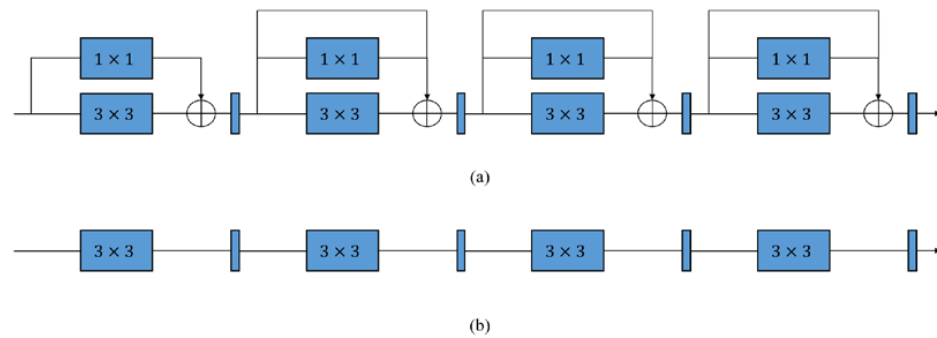


Figure 2. The structure of *RepVGG* in different states: (a) Structure during training; (b) Structure during inference.

Upon completing *RepVGG* training, we employ a structural reparameterization strategy to transform the original multi-branch structure into a simplified, continuous convolution operation, enhancing computational efficiency during inference. First, we fuse the convolutional layer with the batch normalization (BN) layer. Then, in Equation (2), x represents the output value following the convolutional layer operation. By substituting the convolutional layer output, we can transform Equation (2) accordingly.

$$BN(Conv_x) = \gamma \times \frac{wx - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta = \frac{\gamma \times w}{\sqrt{\sigma^2 + \epsilon}}x + \beta - \frac{\gamma \times \mu}{\sqrt{\sigma^2 + \epsilon}}, \tag{3}$$

Equation (3) represents a new fused convolutional layer, with the updated convolution kernel weight denoted as W' and the bias as β' . Specifically:

$$W' = \frac{\gamma \times w}{\sqrt{\sigma^2 + \epsilon}}, \tag{4}$$

$$\beta' = \beta - \frac{\gamma \times \mu}{\sqrt{\sigma^2 + \epsilon}}, \tag{5}$$

For the 3×3 convolutional layer, fusion with the BN layer can be achieved by directly substituting into Equation (3), resulting in a new 3×3 weight, $W'_{3 \times 3}$, and a bias, $\beta'_{3 \times 3}$. For a 1×1 convolutional layer, it must first be transformed into a 3×3 convolutional layer by padding zeros around the 1×1 convolution kernel to create a 3×3 kernel. Substituting this into Equation (3) provides the weight $W'_{1 \times 1}$ and bias $\beta'_{1 \times 1}$ for a new 1×1 convolution branch. For the Identity branch, a 3×3 convolution kernel is established with all nine positions set to 1 to complete the identity mapping. This yields the branch's new weight, W'_0 , and new bias, β'_0 .

Upon transforming all three branches into 3×3 convolutional layers, the weights and biases from each branch are superimposed separately, thus forming a new, fused convolutional operation.

By integrating the BN layers into the convolutional weights and subsequently performing structural reparameterization, a simplified *RepVGG* model is obtained. This model retains the representative capacity of the original multi-branch structure while offering enhanced computational efficiency. Figure 2b illustrates the model's structure following structural reparameterization.

In their paper, the model authors propose an array of *RepVGG* networks. For our purposes, we selected the *RepVGG_{a0}* network structure.

2.2. ECA Attention Mechanism

The attention mechanism emulates human visual or cognitive focus in deep learning models, selectively emphasizing specific input data components. Rather than uniformly weighting all parts during input data processing, each part is assigned a weight reflecting the model’s current attention allocation. Consequently, the model prioritizes task-relevant information, thereby enhancing classification accuracy. Widely employed attention mechanisms include Squeeze-and-Excitation (SE) [29] and Convolutional Block Attention Module (CBAM) [30]. In addition, the Efficient Channel Attention (ECA) [31], depicted in Figure 3, constitutes an effective channel attention strategy designed to augment convolutional neural networks’ feature representation by capturing inter-channel dependencies.

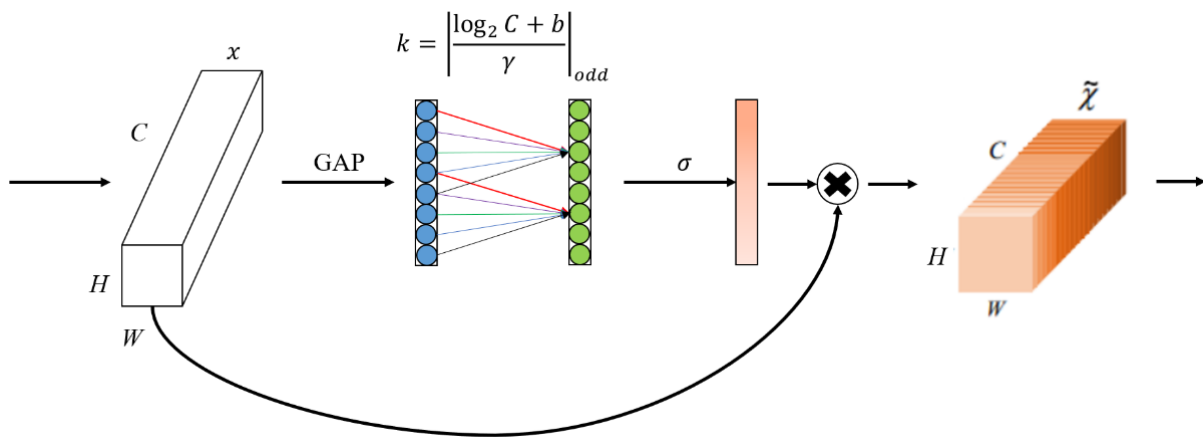


Figure 3. Structural diagram of the ECA attention mechanism.

Given a single input image $X[C, H, W]$, where C represents the number of channels and H and W denote the feature map’s height and width, respectively, the ECA attention mechanism initially conducts Global Average Pooling (GAP) to capture global contextual information for each channel. Specifically, the GAP procedure can be expressed as:

$$GAP(X) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X_{cij}, \forall c \in \{1, \dots, C\}, \tag{6}$$

In this context, X_{cij} represents the (i, j) th element of channel c in the input feature map X . The GAP output is a vector with C dimensions, reflecting the average response of each channel.

Subsequently, the ECA attention mechanism captures local dependencies between channels via a one-dimensional convolution layer (1D Convolution Layer). Denoting a one-dimensional convolution operation with a convolution kernel of size k as F_{conv1D} , the ECA attention mechanism dynamically determines the size k of the 1D Convolution Layer’s convolution kernel. This adaptability enables ECA to accommodate varying numbers of channels and more effectively capture local dependencies between channels. The convolution kernel size k can be computed according to the following equation:

$$k = \left\lfloor \frac{\log_2 C + b}{\gamma} \right\rfloor_{odd} \tag{7}$$

Here, γ serves as a hyperparameter governing the scope of local dependencies. A smaller γ value yields a larger convolution kernel size, encompassing a broader range of channel dependencies.

The output of a one-dimensional convolutional layer can be expressed as:

$$Conv1D(GPA(X)) = F_{conv1D}(GPA(X)), \tag{8}$$

To facilitate adaptive learning of inter-channel correlations within the model, the output of the 1D convolutional layer undergoes nonlinear transformation via a Sigmoid activation function, yielding a vector of attention weights:

$$A = \sigma(\text{Conv1D}(\text{GPA}(X))) \quad (9)$$

Ultimately, the attention weight vector A is multiplied element-wise by the original input feature map X , effectuating channel attention rescaling. The recalibrated feature map Y_{cij} is computed as follows:

$$Y_{cij} = A_c \cdot X_{cij}, \forall c \in \{1, \dots, C\}, i \in \{1, \dots, H\}, j \in \{1, \dots, W\}, \quad (10)$$

The ECA attention mechanism bolsters the feature representation of convolutional neural networks by capturing inter-channel dependencies. Its merits include high computational efficiency, a low parameter count, and seamless compatibility with existing convolutional neural networks. In addition, by incorporating localized adaptive channel attention, the ECA attention mechanism enhances feature representation and elevates model performance.

2.3. Proposed Methods

In deep learning-based image classification, recent research has demonstrated that incorporating attention mechanisms can significantly enhance a model's classification accuracy. The *RepVGG* model, which has already demonstrated its robust classification performance on ImageNet, is an ideal candidate for addressing the unique challenges associated with rice pest and disease classification. These challenges include the similarity of early-stage symptoms across various pests and diseases, the infrequent appearance of certain pests and diseases, the impact of weather and lighting conditions on image quality, and the cluttered rice background that complicates feature learning. The Efficient Channel Attention (ECA) module adaptively recalibrates channel-wise feature responses and effectively addresses these challenges by explicitly modeling interdependencies between channels, thereby improving the model's generalization capabilities. Moreover, the ECA module's lightweight and efficient design enables performance enhancement without substantially increasing the model's complexity. Consequently, integrating the ECA module into the *RepVGG* architecture proves to be a logical and beneficial strategy for improving the classification of rice pests and diseases.

Our design amalgamates *RepVGG* blocks and ECA modules to construct the overall model. As depicted in Figure 4, we have innovatively integrated the ECA module into two *RepVGG* blocks, referred to as Block A_ECA and Block B_ECA, and incorporated the ECA module after the Head. "Head" refers to the first layer of the *RepVGG_ECA* model architecture. This aims to enhance classification performance by emphasizing crucial features within the input data and guides the model to concentrate on specific regions or channels. Furthermore, to mitigate overfitting during training, we employ the L2 regularization method. This paper's fourth section will juxtapose our proposed approach against alternative models to validate its superiority.

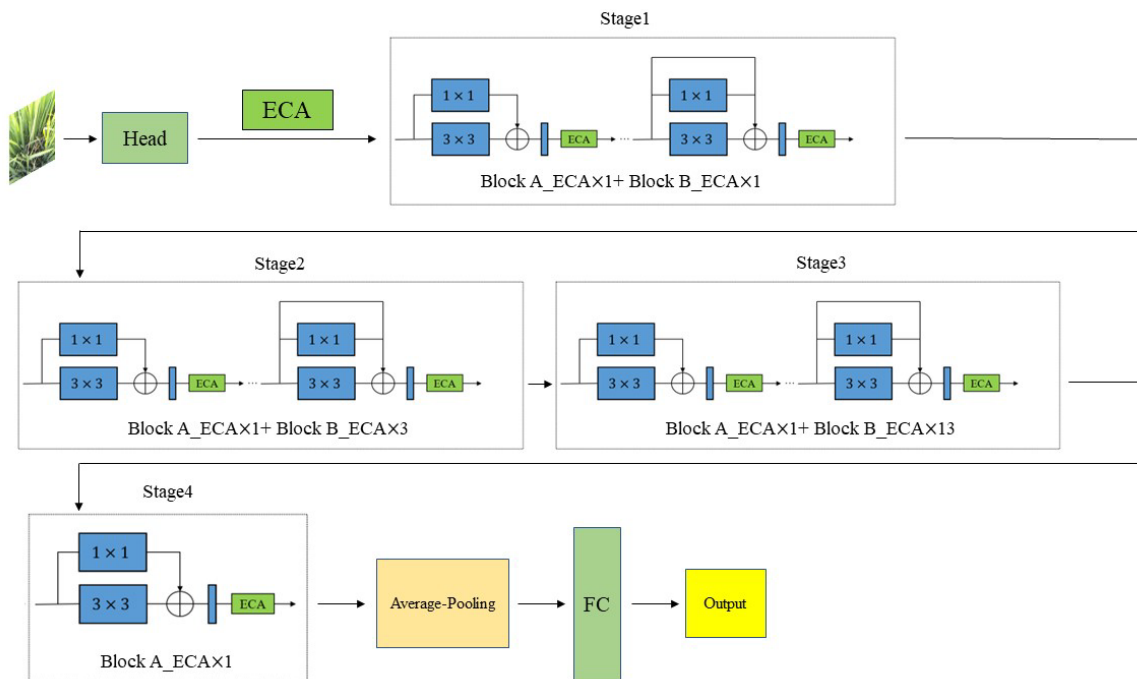


Figure 4. The structures we have designed.

3. Experiments and Results

The hardware and software environment for this experiment is as follows: Microsoft Windows 11 operating system, CPU: AMD Ryzen 7 5800H, GPU: NVIDIA GeForce RTX 3060 Laptop (6 GB video memory), CUDA 11.1 and cuDNN 8.0.5, PyTorch 1.12.1 deep learning framework.

3.1. Dataset

The dataset employed in this experiment is sourced from “Paddy Doctor: A Visual Image Dataset for Automated Paddy Disease Classification and Benchmarking [32].” In Figure 5, we have selected five types of rice pests and diseases, along with healthy rice, each forming a category for our classification study. The dataset encompasses Blast, Brown_spot, Dead_heart, Hispa, Tungro, and Normal categories. The images in the dataset are color RGB images with a resolution of 480×640 .

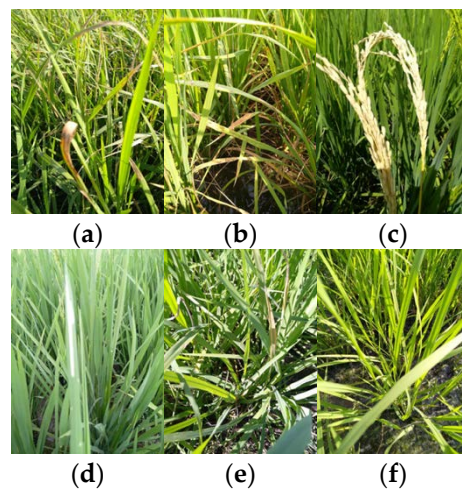


Figure 5. The datasets used in the experiments were divided into six categories. (a) Blast; (b) Brown_spot; (c) Dead_heart; (d) Hispa; (e) Tungro; (f) Normal.

The dataset, upon completion of data augmentation, comprised a total of 11,907 samples and was categorized into three distinct subsets: training, validation, and test sets. A random selection of 10% of the augmented dataset constituted the test set. The remaining samples were subsequently partitioned, with 80% allocated to the training set and 20% to the validation set. Table 1 delineates the specific groupings for each subset.

Table 1. Specific groupings.

Type	Training Set	Validation Set	Testing Set
Blast	1482	370	206
Brown_spot	1386	346	193
Dead_heart	1413	353	196
Hispa	1436	359	199
Tungro	1414	354	196
Normal	1442	362	200

3.2. Data Augmentation

As evident from Table 2, the images in this dataset exhibit overall diversity; however, the sample sizes for Brown_spot and Tungro pests are considerably smaller than those of other types. For example, Table 2 reveals that the normal category has the largest sample size at 20.5%, significantly exceeding the smallest category, Brown_spot, which accounts for only 11.2%. This suggests that certain rice pests and diseases may occur less frequently in actual production, leading to an imbalanced dataset. Consequently, during training, the model may overly concentrate on more abundant categories while neglecting less frequent ones, ultimately diminishing classification accuracy.

Table 2. Number and percentage of images of each type of rice.

Type	Blast	Brown_spot	Dead_heart	Hispa	Normal	Tungro
Number	1738	965	1442	1594	1764	1088
Proportion (%)	20.2	11.2	16.8	18.6	20.5	12.7

In deep learning-based image classification, image enhancement plays a crucial role. Primarily, data augmentation increases the dataset's diversity by transforming the original dataset to generate new samples. This enables the model to learn more features and perform effectively in various situations during training, consequently improving its generalization capacity. Moreover, data augmentation can create more samples by transforming smaller categories, thereby alleviating data imbalance issues, and enhancing the model's performance across all categories. Furthermore, by perturbing the original image to different extents, data augmentation exposes the model to more transformed samples during training, helping it learn more robust features and improving performance on noisy or perturbed data. Lastly, data augmentation techniques can expand the number of training samples and prevent overfitting issues in small datasets. In this study, we employ four image-processing techniques to enrich the initial dataset:

Flipping: Inverting an image, the model can learn features and corresponding classification information in different orientations, enhancing its robustness when processing mirrored or flipped samples. For numerous vision tasks, such as object recognition and classification, flipping usually does not alter the image's semantic information.

Modifying saturation: Adjusting an image's saturation lets the model learn features at varying color purities. This maintains better performance in lighting or color deviations and bolsters the model's color robustness.

Modifying contrast: By altering an image's contrast, the model can learn feature information under different contrast conditions, which in turn aids in improving the classification performance on images with variable lighting or low contrast.

Adding blur: Blurring simulates image distortion due to factors like focus and motion during the capturing process. By introducing blur to an image, the model can learn feature information at different levels of blurriness, thereby enhancing its robustness when handling images with varying degrees of sharpness or focus.

Table 3, presented below, enumerates the quantity of each image type following the application of data enhancement techniques. Concurrently, Figure 6 illustrates the effect of these four processing techniques on a single representative image.

Table 3. The sample size of various classes in the data set after data augmentation.

Type	Blast	Brown_spot	Dead_heart	Hispa	Normal	Tungro
Number	2058	1925	1962	1994	1964	2004
Proportion (%)	17.3	16.2	16.5	16.7	16.5	16.8

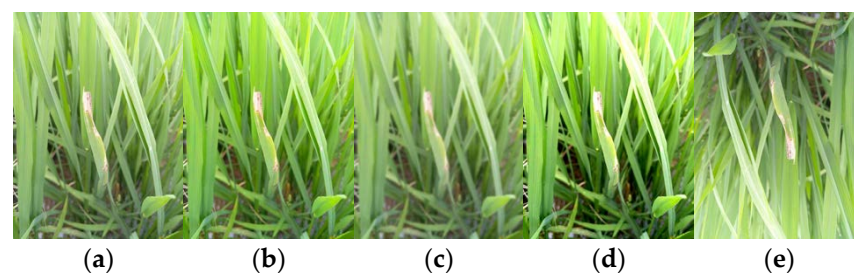


Figure 6. Data to enhance contrast. (a) original image; (b) modify saturation; (c) modify contrast; (d) add blur; (e) flip.

3.3. Pest and Disease Classification

In the context of image classification using a model, parameter selection plays a crucial role in enhancing the model's classification accuracy. T2 regularization is employed in conjunction with the Adam optimizer, a learning rate of 0.0001, and a maximum epoch count of 100, to mitigate overfitting.

Figure 7 depicts the variation curves for four key parameters. The parameter 'train_loss' represents the model's loss during training on the training dataset, while 'train_accuracy' signifies the proportion of accurate predictions made using the training dataset. Conversely, 'val_loss' corresponds to the model's loss on the validation dataset, a metric for evaluating the model's performance on previously unseen data. Lastly, 'val_accuracy' indicates the percentage of accurate predictions made by the model on the validation dataset.

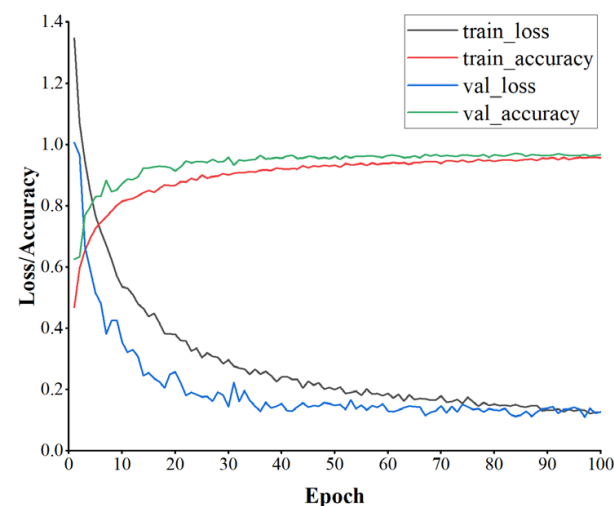


Figure 7. Accuracy and curves of loss during training.

Figure 7 clearly illustrates that ‘train_loss’ experiences a rapid decrease initially, particularly within the first 20–30 epochs. Subsequently, the decline becomes more gradual, interspersed with minor fluctuations during specific periods. Despite these fluctuations, the overall trend is downward, signifying the model’s learning and improvement in performance on the training dataset. Concurrently, ‘train_accuracy’ exhibits a swift increase during the initial 20–30 epochs, followed by a decelerated growth rate. The upward trend persists, albeit more gradually, with minimal fluctuations at certain intervals, indicating the model’s increased accuracy on the training dataset. ‘val_loss’ undergoes a sharp reduction within the first ten epochs, followed by a more moderate decline while demonstrating more frequent and pronounced fluctuations compared to ‘train_loss’.

Nevertheless, the overall trend is decreasing, suggesting that the model generalizes effectively to the validation dataset without overfitting the training data. ‘val_accuracy’ displays a substantial increase during the initial ten epochs, with the rate of growth diminishing after that. This trend implies that the model improves its performance on the validation dataset and generalizes well to unseen data. Consequently, we conclude that the model learns efficiently from the training data and has good generalization.

The results of the test were evaluated using the four indicators Accuracy, Macro-Precision, Macro-Recall, and Macro-F1. The specific formulae are shown below:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (11)$$

$$Macro - Precision = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i} \quad (12)$$

$$Macro - Recall = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FN_i} \quad (13)$$

$$Macro - F1 = \frac{1}{N} \sum_{i=1}^N \frac{2 \times P_i \times R_i}{P_i + R_i} \quad (14)$$

In the aforementioned equation, TP represents the True Positives, i.e., the correct prediction of positive samples as positive by the classifier. FP signifies the False Positives, i.e., the incorrect prediction of negative samples as positive. FN denotes the False Negatives, i.e., the incorrect prediction of positive samples as negative. TN refers to the True Negatives, i.e., the correct prediction of negative samples as negative. N corresponds to the number of defect types, which, in this study, amounts to 6. P_i and R_i denote the precision and recall for individual image types, respectively.

Table 4 presents the specific classification results. The classification accuracy of *RepVGG_ECA* stands at 97.06%, demonstrating the model’s effectiveness in classifying typical pests and diseases in rice. In addition, the model exhibits robustness and generalization capabilities. Figure 8 showcases the detailed classification in the form of a confusion matrix.

Table 4. Classification results.

Accuracy (%)	Macro-Precision (%)	Macro-Recall (%)	Macro-F1 (%)
97.06	97.13	97.08	97.09

Examining the confusion matrix, the model attains 100% accuracy for Dead_heart and Tungro. For Hispa, however, the model demonstrated the lowest accuracy at 94.6%, and the recall and F1 scores were similarly lower at 96% and 95.29%, respectively. This reduced performance can be attributed to the less distinctive features of Hispa, which shares more similarities with other pests and disease features. Despite this, the model achieves an overall classification accuracy of 97.06%. Consequently, our model delivers satisfactory classification results for typical rice pest images.

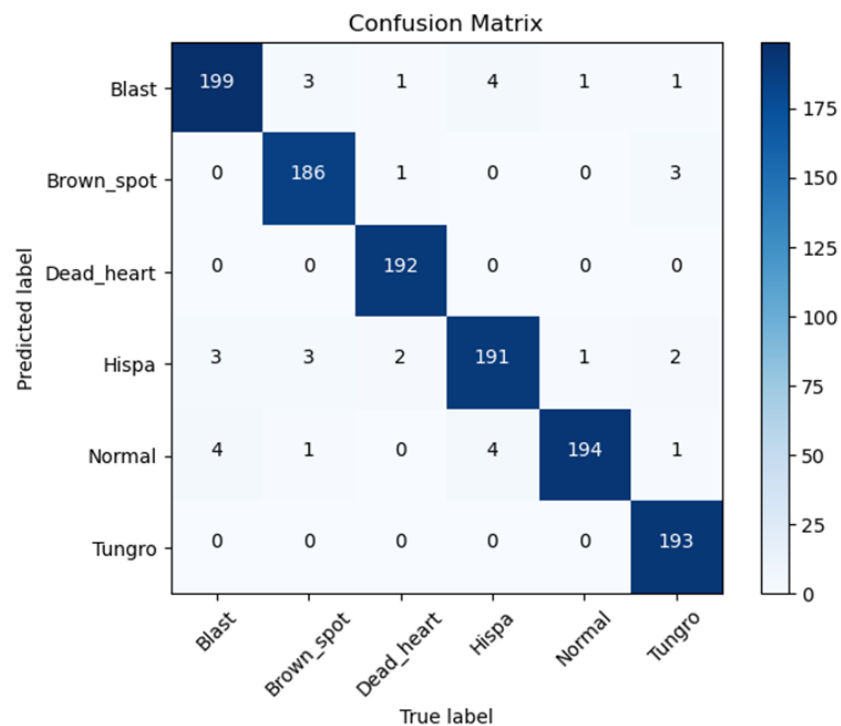


Figure 8. Confusion Matrix.

4. Discussion

4.1. Comparison with Other Models

To demonstrate the superior classification performance of our proposed model, we compared it with *ResNet34*, *ResNeXt50*, *ShuffleNet V2*, and the basic *RepVGG_{a0}* model for the classification of a typical rice pest dataset. Table 5 presents a comparison between the selected models and our proposed approach. As evident from the table, our method exhibits the highest classification accuracy, outperforming *ResNet34*, *ResNeXt50*, *ShuffleNet V2*, and the basic *RepVGG_{a0}* by 1.85%, 1.18%, 3.39%, and 1.09%, respectively. Additionally, our approach is optimal in terms of Macro-Precision, Macro-Recall, and Macro-F1, as shown by the three key performance parameters.

Table 5. Comparison between different models.

Methods	Accuracy (%)	Macro-Precision (%)	Macro-Recall (%)	Macro-F1 (%)
<i>ResNet34</i>	95.21	95.4	95.2	95.24
<i>ResNeXt50</i>	95.88	96.02	95.9	95.92
<i>ShuffleNet V2</i>	93.67	93.67	93.35	93.42
<i>RepVGG_{a0}</i>	95.97	95.48	95.98	96.01
<i>Our method</i>	97.06	97.13	97.08	97.09

Figure 9 presents the training set accuracy of all models compared after 100 epochs on the training set. As *ShuffleNet* is a lightweight network model with a small number of layers, it only achieved a maximum accuracy of 90.01%. In contrast, our method surpasses the accuracy of the other models, with the basic *RepVGG_{a0}* model outperforming other basic models, validating our choice of *RepVGG* as the base model. Concerning convergence speed, our method, even with a more complex network structure due to the addition of the ECA module, converges faster than the basic *RepVGG_{a0}* model and attains a satisfactory convergence speed. These aspects highlight the superiority of our designed method.

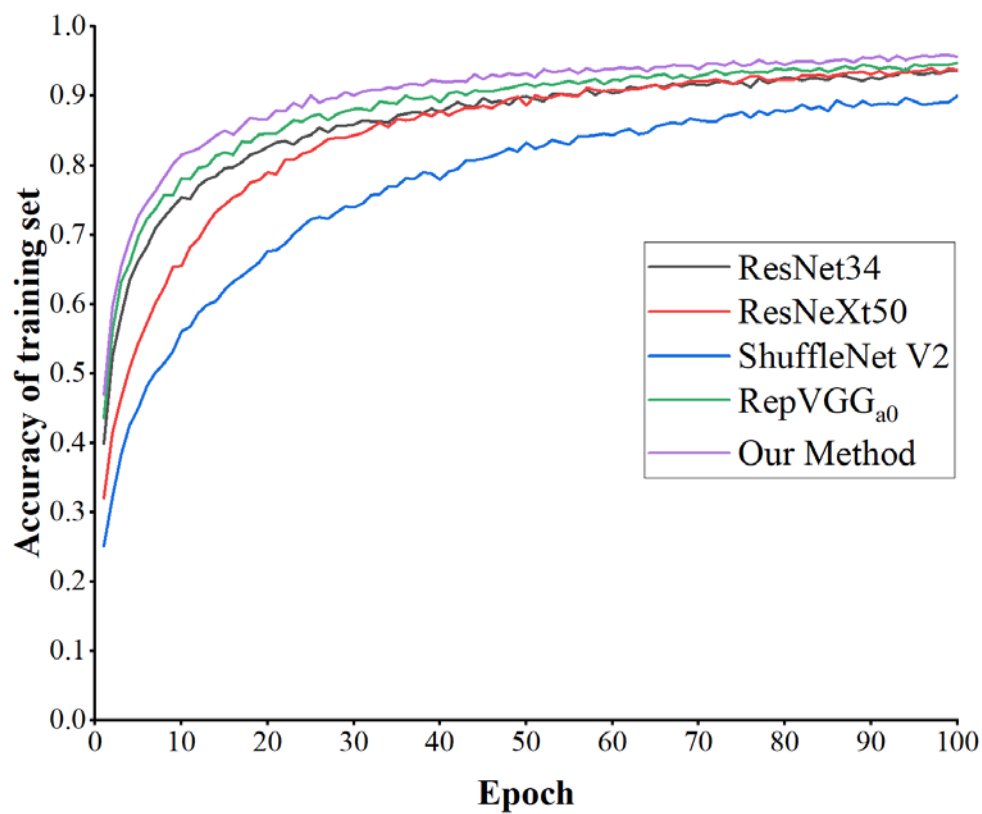


Figure 9. Comparison of the accuracy of the training set of each model.

Figure 10 illustrates the loss of all models compared to the validation set. *ResNeXt50* exhibits significant fluctuations and is the least stable. On the other hand, *ShuffleNet V2* demonstrates the smoothest and most stable curve with minimal fluctuations. Our method ranks second in stability, only surpassed by the *ShuffleNet V2* model, but achieves the lowest loss value. In comparison to *RepVGG_{a0}*, our method reduces the loss while maintaining stability.

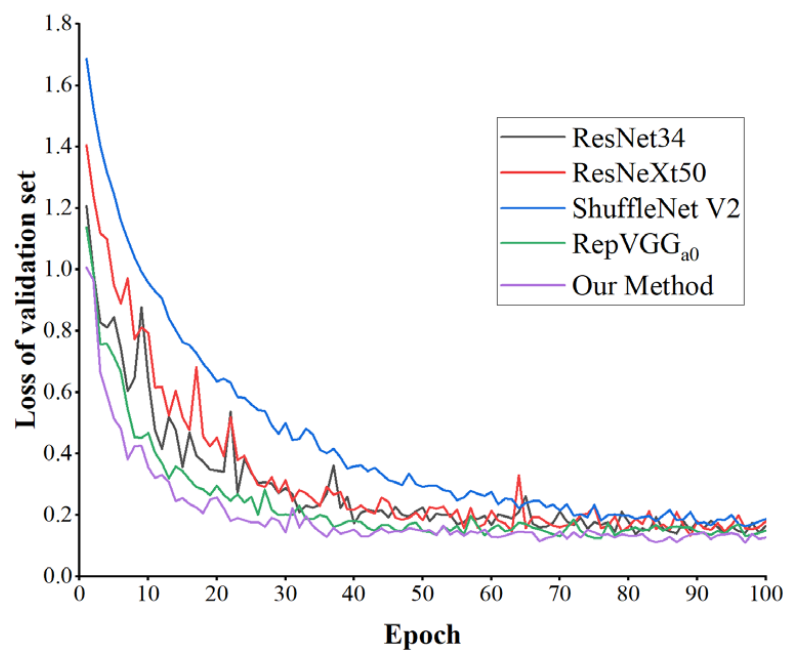


Figure 10. Comparison of the validation set losses across models.

In summary, our method exhibits good stability during the training process, ensuring the lowest loss value and the highest accuracy, making it the optimal method for rice pest and disease classification.

4.2. Ablation Study

To explore the impact of the ECA attention mechanism on *RepVGG*'s classification performance, we conducted an ablation study, the results of which are displayed in Table 6. Our findings indicate that incorporating additional ECA modules does not enhance the model's performance. Instead, optimal results are achieved when the ECA module is added both in the block and after the head, leading to a notably higher classification accuracy than other model configurations. These results suggest that the judicious insertion of ECA modules can yield improved classification outcomes, further reaffirming the superiority of our proposed method.

Table 6. Results of ablation study. “√” indicates the inclusion of the ECA attention mechanism after that position and “×” indicates that it is not inserted.

Block	Head	Stage 1	Stage 2	Stage 3	Stage 4	Accuracy (%)
√	×	×	×	×	×	96.55
√	√	×	×	×	×	97.06
√	√	√	×	×	×	95.97
√	√	√	√	×	×	96.22
√	√	√	√	√	×	96.47
√	√	√	√	√	√	96.31

5. Conclusions

In addressing the challenge of unusual training sample distribution in rice pest image datasets, this study puts forward four techniques: flipping, modifying saturation, modifying contrast, and adding a blur. These methods effectively simulate diverse rice environments under varying conditions, facilitating more efficient classification training for the model. In addition, we have proposed a novel *RepVGG_ECA* model for rice pest and disease classification. This model is based on the *RepVGG_{a0}* model, with an integrated ECA attention mechanism within the block and after the head. It achieves a remarkable accuracy of 97.06%, outperforming *ResNet34*, *ResNeXt50*, *ShuffleNet V2*, and the basic *RepVGG_{a0}* model.

Furthermore, the model excels in all three-evaluation metrics: macro precision, macro recall, and macro F1 score. The stability of the training process, free of overfitting issues, results in excellent classification results. An ablation study has revealed that while the ECA attention mechanism is beneficial, its excessive inclusion does not further enhance classification accuracy. The optimal results are obtained by strategically adding the mechanism within the block and after the head. This work lays a solid foundation for future research in this area.

While the *RepVGG_ECA* model demonstrates remarkable classification performance, it exhibits relatively poorer performance on Hispa images, which feature less prominent pests and diseases. Recognizing this, we plan to incorporate an optimization algorithm in our future work for fine-tuning hyperparameters and, simultaneously, enabling a more lightweight design without compromising on improved classification performance. This could significantly enhance the model's performance. In addition, including an optimization algorithm for hyperparameters adjustment could allow the model to better adapt to various data patterns and reduce bias, especially in classifying less prominent pests and diseases such as Hispa. This optimization approach will likely provide a more balanced and accurate model in future iterations.

Author Contributions: Conceptualization, H.N. and Z.S.; methodology, Z.S.; software, S.K.; validation, H.N., Z.S. and S.L.; formal analysis, H.N. and Z.S.; investigation, S.K. and X.L.; resources, H.N. and S.K.; data curation, H.N. and Z.S.; writing—original draft preparation, Z.S.; writing—review and editing, H.N., Z.S. and J.Z.; visualization, H.N. and S.L.; supervision, X.W.; project administration, H.N.; funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Priority Academic Program Development of Jiangsu Higher Education Institutions, grant number PAPD; Jiangsu Province Policy Guidance Program (International Science and Technology Cooperation) Project, grant number BZ2021045.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhao, X.; Ma, L.; Qu, P.; Yue, B.; Zhao, X.; Shang, X.; Zhao, Y.; Wu, Y. Total mercury and methylmercury in Chinese rice and dietary exposure assessment. *Food Addit. Contam. Part B* **2020**, *13*, 148–153. [CrossRef]
2. Li, F.; Zhang, W. Research on the Effect of Digital Economy on Agricultural Labor Force Employment and Its Relationship Using SEM and fsQCA Methods. *Agriculture* **2023**, *13*, 566. [CrossRef]
3. Tan, C.; Tao, J.; Yi, L.; He, J.; Huang, Q. Dynamic Relationship between Agricultural Technology Progress, Agricultural Insurance, and Farmers' Income. *Agriculture* **2022**, *12*, 1331. [CrossRef]
4. International Rice Research Institute. *Rice in the Global Economy: Strategic Research and Policy Issues for Food Security*; International Rice Research Institute: Los Baños, Philippines, 2010.
5. Xiaopeng, D.; Donghui, L. Research on Rice Pests and Diseases Warning Based on CBR. In Proceedings of the 2013 International Conference on Computational and Information Sciences, Shiyang, China, 21–23 June 2013; pp. 1740–1742. [CrossRef]
6. Prajapati, H.B.; Shah, J.P.; Dabhi, V.K. Detection and classification of rice plant diseases. *Intell. Decis. Technol.* **2017**, *11*, 357–373. [CrossRef]
7. Alfatni, M.S.M.; Khairunniza-Bejo, S.; Marhaban, M.H.B.; Saaed, O.M.B.; Mustapha, A.; Shariff, A.R.M. Towards a Real-Time Oil Palm Fruit Maturity System Using Supervised Classifiers Based on Feature Analysis. *Agriculture* **2022**, *12*, 1461. [CrossRef]
8. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [CrossRef] [PubMed]
9. Xu, N. Image Processing Technology in Agriculture. *J. Phys. Conf. Ser.* **2021**, *1881*, 32097. [CrossRef]
10. Azim, M.A.; Islam, M.K.; Rahman, M.; Jahan, F. An effective feature extraction method for rice leaf disease classification. *TELKOMNIKA* **2021**, *19*, 463–470. [CrossRef]
11. Shrivastava, V.K.; Pradhan, M.K. Rice plant disease classification using color features: A machine learning paradigm. *J. Plant Pathol.* **2021**, *103*, 17–26. [CrossRef]
12. Wang, P.; Fan, E.; Wang, P. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognit. Lett.* **2021**, *141*, 61–67. [CrossRef]
13. Lai, Y. A Comparison of Traditional Machine Learning and Deep Learning in Image Recognition. *J. Phys. Conf. Ser.* **2019**, *1314*, 12148. [CrossRef]
14. Hegde, R.B.; Prasad, K.; Hebbar, H.; Singh, B.M.K. Comparison of traditional image processing and deep learning approaches for classification of white blood cells in peripheral blood smear images. *Biocybern. Biomed. Eng.* **2019**, *39*, 382–392. [CrossRef]
15. Alfariy, A.A.; Chen, Q.; Guo, M. Deep learning based classification for paddy pests & diseases recognition. In Proceedings of the 2018 International Conference on Mathematics and Artificial Intelligence, Chengdu, China, 20–22 April 2018; pp. 21–25. [CrossRef]
16. Sagarika, G.K.; Prasad, S.J.K.; Kumar, S.M. Paddy Plant Disease Classification and Prediction Using Convolutional Neural Network. In Proceedings of the 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 12–13 November 2020; pp. 208–214. [CrossRef]
17. Wang, Y.; Wang, H.; Peng, Z. Rice diseases detection and classification using attention-based neural network and bayesian optimization. *Expert Syst. Appl.* **2021**, *178*, 114770. [CrossRef]
18. Patil, R.R.; Kumar, S. Rice-fusion: A multimodality data fusion framework for rice disease diagnosis. *IEEE Access* **2022**, *10*, 5207–5222. [CrossRef]
19. Liang, K.; Wang, Y.; Sun, L.; Xin, D.; Chang, Z. A Lightweight-Improved CNN Based on VGG16 for Identification and Classification of Rice Diseases and Pests. In Proceedings of the International Conference on Image, Vision and Intelligent Systems (ICIVIS 2021), Changsha, China, 21–23 May 2022; pp. 195–207. Available online: https://link.springer.com/chapter/10.1007/978-981-16-6963-7_18 (accessed on 28 March 2023).
20. Lu, Y.; Li, Z.; Zhao, X.; Lv, S.; Wang, X.; Wang, K.; Ni, H. Recognition of Rice Sheath Blight Based on a Backpropagation Neural Network. *Electronics* **2021**, *10*, 2907. [CrossRef]

21. Burhan, S.A.; Minhas, S.; Tariq, A.; Hassan, M.N. Comparative study of deep learning algorithms for disease and pest detection in rice crops. In Proceedings of the 2020 12th International Conference on Electronics, Computers and Artificial Intelligence, Bucharest, Romania, 25–27 June 2020; pp. 1–5. [[CrossRef](#)]
22. Hu, Y.; Deng, X.; Lan, Y.; Chen, X.; Long, Y.; Liu, C. Detection of Rice Pests Based on Self-Attention Mechanism and Multi-Scale Feature Fusion. *Insects* **2023**, *14*, 280. [[CrossRef](#)]
23. Rahman, C.R.; Arko, P.S.; Ali, M.E.; Khan, M.A.I.; Apon, S.H.; Nowrin, F.; Wasif, A. Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyst. Eng.* **2020**, *194*, 112–120. [[CrossRef](#)]
24. Li, D.; Wang, R.; Xie, C.; Liu, L.; Zhang, J.; Li, R.; Wang, F.; Zhou, M.; Liu, W. A Recognition Method for Rice Plant Diseases and Pests Video Detection Based on Deep Convolutional Neural Network. *Sensors* **2020**, *20*, 578. [[CrossRef](#)]
25. Li, Z.; Jiang, X.; Jia, X.; Duan, X.; Wang, Y.; Mu, J. Classification Method of Significant Rice Pests Based on Deep Learning. *Agronomy* **2022**, *12*, 2096. [[CrossRef](#)]
26. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
28. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making vgg-style convnets great again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
29. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [[CrossRef](#)] [[PubMed](#)]
30. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
31. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. Supplementary material for ‘ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13–19.
32. Kiruba, B.; Arjunan, P. Paddy Doctor: A Visual Image Dataset for Automated Paddy Disease Classification and Benchmarking. In Proceedings of the 6th Joint International Conference on Data Science & Management of Data, Mumbai, India, 4–7 January 2023; pp. 203–207. [[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.