



Article

Alpha-EIOU-YOLOv8: An Improved Algorithm for Rice Leaf Disease Detection

Dong Cong Trinh¹, Anh Tuan Mac², Khanh Giap Dang², Huong Thanh Nguyen², Hoc Thai Nguyen³ 
and Thanh Dang Bui^{1,2,*} 

¹ Institute for Control Engineering and Automation, Hanoi University of Science and Technology, Dai Co Viet Street, Hanoi 100000, Vietnam; dong.trinhcong@hust.edu.vn

² School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Dai Co Viet Street, Hanoi 100000, Vietnam; anh.mt192182@sis.hust.edu.vn (A.T.M.); khanh.gd192219@sis.hust.edu.vn (K.G.D.); huong.nguyenthanh3@hust.edu.vn (H.T.N.)

³ Faculty of Engineering, Vietnam National University of Agriculture, Trauquy, Gialam, Hanoi 131000, Vietnam; nguyenthaihoc@vnua.edu.vn

* Correspondence: thanh.buidang@hust.edu.vn; Tel.: +84-9158-97699

Abstract: Early detection of plant leaf diseases is a major necessity for controlling the spread of infections and enhancing the quality of food crops. Recently, plant disease detection based on deep learning approaches has achieved better performance than current state-of-the-art methods. Hence, this paper utilized a convolutional neural network (CNN) to improve rice leaf disease detection efficiency. We present a modified YOLOv8, which replaces the original Box Loss function by our proposed combination of EIoU loss and α -IoU loss in order to improve the performance of the rice leaf disease detection system. A two-stage approach is proposed to achieve a high accuracy of rice leaf disease identification based on AI (artificial intelligence) algorithms. In the first stage, the images of rice leaf diseases in the field are automatically collected. Afterward, these image data are separated into *blast leaf*, *leaf folder*, and *brown spot* sets, respectively. In the second stage, after training the YOLOv8 model on our proposed image dataset, the trained model is deployed on IoT devices to detect and identify rice leaf diseases. In order to assess the performance of the proposed approach, a comparative study between our proposed method and the methods using YOLOv7 and YOLOv5 is conducted. The experimental results demonstrate that the accuracy of our proposed model in this research has reached up to 89.9% on the dataset of 3175 images with 2608 images for training, 326 images for validation, and 241 images for testing. It demonstrates that our proposed approach achieves a higher accuracy rate than existing approaches.

Keywords: rice leaf diseases; blast leaf; leaf folder; brown spot; YOLOv8



Citation: Trinh, D.C.; Mac, A.T.; Dang, K.G.; Nguyen, H.T.; Nguyen, H.T.; Bui, T.D. Alpha-EIOU-YOLOv8: An Improved Algorithm for Rice Leaf Disease Detection. *AgriEngineering* **2024**, *6*, 302–317. <https://doi.org/10.3390/agriengineering6010018>

Academic Editors: Chiew Foong Kwong and Ray E. Sheriff

Received: 24 November 2023

Revised: 19 January 2024

Accepted: 22 January 2024

Published: 4 February 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 (*Oryza sativa*) is one of the most important staple crops in the world, not only providing a rich source of carbohydrates but also plenty of vitamins and minerals for human healthy growth and development. This adaptable grain is an essential element of diets and Asian cuisine [1]. Unfortunately, the recent successful cultivation of rice faces numerous challenges, and one of the most significant threats to its yield and quality comes in the form of rice leaf diseases. These diseases, caused by various pathogens, can severely impact the health of rice plants, resulting in reduced crop yields and low-quality harvests. Therefore, accurate and timely detection of rice leaf diseases is of paramount importance and has become a primary agricultural concern.

Nowadays, science and technology are growing rapidly with many breakthrough inventions that make people's lives easier and simpler. The adoption of modern technologies, including machine learning and deep learning, has opened up new avenues for tackling these challenges in the agriculture sector. One of the most popular technologies that can be

used to solve this problem is the use of deep learning technology on plants to identify and classify diseases. In recent years, more and more studies have focused on disease detection and crop management based on these technologies with more and more improvement in accurate detection rate [2–5].

Deep learning is a subset of machine learning that seeks to mimic the data processing and decision-making mechanisms of the human brain. Within deep learning, CNN stands as a vital element in various fields such as speech recognition, image recognition, natural language processing, and the field of genomics [6]. Its utility extends across diverse real applications such as health care, finance, and even education as in the E-Learning Modeling Technique and CNN as proposed in [7]. Notably, there have been encouraging outcomes in scientific investigations that employ deep learning to identify diseases in specific plant species. In this research area, several noteworthy studies have explored disease detection in plants. For instance, study [8] focused on disease detection in cucumber plants using the YOLOv4 network for leaf image analysis. The authors achieved impressive accuracy rates exceeding 80% with a dataset comprising over 7000 images. However, since this study relied solely on a convolutional neural network (CNN) model, there were limitations to its accuracy. Another study [9] delved into deep learning methods for surveillance and the detection of plant foliar diseases. It examined images at various infection levels and proposed a model to assess disease progression over the plant growth cycle. This study also investigated factors influencing image quality, such as dataset size, learning rate, illumination, and more. In a different context, research [10] utilized YOLOv3 to detect *brown spot* and *leaf blast* diseases. Nevertheless, the practicality of this approach was hindered because it exclusively used images with white backgrounds, limiting its flexibility. Meanwhile, studies [11,12] leveraged ResNet and YOLOv3 to detect diseases in tomato leaves. In [13], the authors introduced their CNN model for classifying *leaf blight* diseases. Furthermore, research [14] conducted a comprehensive comparison of CNN models, including DenseNet-121, ResNet-50, ResNeXt-50, SE-ResNet-50, and ResNeSt-50. They even combined these models to improve detection results. Another study [15] employed a CNN model to detect *brown spot* and *leaf blast* diseases in rice, achieving an impressive overall accuracy of 0.91. Vimal K. Shrivastava et al. in [16], utilized AlexNet and SVM to classify three rice leaf diseases, achieving an accuracy rate of 91.37%. Additionally, study [17] performed a survey and comparative analysis of different models for detecting diseases in rice leaves and seedlings. Research [18] used VGG16 and InceptionV3 to classify five distinct rice diseases. Lastly, in research [19], SVM was combined with a DCNN model to identify and classify nine different types of rice diseases, achieving an impressive accuracy rate of 97.5%. Article [20] presents a method for recognizing and classifying paddy leaf diseases, such as *leaf blast*, *bacterial blight*, *sheath rot*, and *brown spot*, using an optimized deep neural network with the Jaya algorithm. The classification of diseases is carried out using an optimized deep neural network with the Jaya optimization algorithm. The experimental results show high accuracy for different types of diseases. In [21], the author used the hue value to extract the non-diseased and diseased parts during the pre-processing process and used CNN for feature extraction to be able to identify the diseases including rice *blast*, *sheath rot*, *bacterial leaf blight*, *brown spot*, *rymv*, and *rice tungro*. Studies [22,23] used YOLOv5 to detect various typical rice leaf diseases.

In this research, we use the YOLOv8 model to detect the three most common rice leaf diseases in Vietnam, namely *leaf blast* (Section 1.1), *leaf folder* (Section 1.2), and *brown spot* (Section 1.3).

1.1. Leaf Blast

The primary culprit behind *leaf blast* disease is the fungus *Magnaporthe oryzae*, which can afflict all components of the rice plant, including the leaves, collar, node, neck, portions of the panicle, and occasionally the leaf sheath. The occurrence of blast disease is contingent upon the presence of blast spores [24]. The *leaf blast* disease is depicted in Figure 1a.



Figure 1. Common rice leaf diseases in Vietnam.

To identify *leaf blast* disease, a thorough examination of the leaf and collar is necessary. Initial signs manifest as pale to grayish-green lesions or spots, characterized by dark green borders. As these lesions mature, they assume an elliptical or spindle-shaped configuration, featuring whitish-to-gray centers surrounded by reddish-to-brownish or necrotic boundaries. Another indicative symptom resembles a diamond shape, broader at the center and tapering toward both ends [24].

1.2. Leaf Folder

The *leaf folder* is formed by *leaf folder* caterpillars, which encircle the rice leaf and secure the leaf edges with threads of silk. They consume the inner part of the folded leaves, leading to the development of whitish and see-through streaks along the surface of the leaf [25]. *Leaf folder* disease is shown in Figure 1b.

Infected leaves typically exhibit vertical and translucent white streaks, along with tubular leaf structures. Occasionally, the leaf tips are affixed to the leaf base. Fields heavily affected by this condition may present an appearance of scorching due to the prevalence of numerous folded leaves [25].

1.3. Brown Spot

Brown spot is a fungal disease that infects cotyledons, leaves, sheaths, panicles, and shoots. Its most noticeable impact is the formation of numerous prominent spots on the leaves, which have the potential to lead to the demise of the entire leaf. In cases of seed infection, it results in unfilled seeds or seeds displaying spots or discoloration [24]. The *brown spot* disease is shown in Figure 1c.

Infected seedlings display small, circular lesions that are yellowish-brown or brown in color and may encircle the cotyledons while distorting the primary and secondary leaves. Starting from the tillering stage, lesions begin to appear on the leaves. Initially, these lesions are small and circular and range from dark brown to purplish-brown. As they fully mature, they adopt a round-to-oval shape with a light brown-to-gray center, encircled by a reddish-brown border produced by *mycotoxins*. On susceptible rice varieties, lesions measuring between 5 and 14 mm in length can result in leaf wilting. Conversely, on resistant varieties, the lesions are brown and approximately the size of a fingertip [24].

The rest of the paper is planned as follows. Section 2 presents the methodology with the system overview, a hardware design, and data preparation. It also demonstrates the YOLOv8 architecture, along with evaluation metrics and a loss calculation. The experimental results and the discussion are given in Section 3. Finally, conclusions, acknowledgments, and references are made in Section 4.

2. Materials and Methods

2.1. System Overview

The procedure can be outlined in the following steps: First, the images of rice leaf diseases are collected and added to dataset. Then, the YOLOv8 model is employed to

swiftly and accurately detect diseases in rice leaves. Finally, the symptoms of unhealthy leaves of rice are used to alert farmers via email and text message. The structure of the system is depicted in Figure 2.

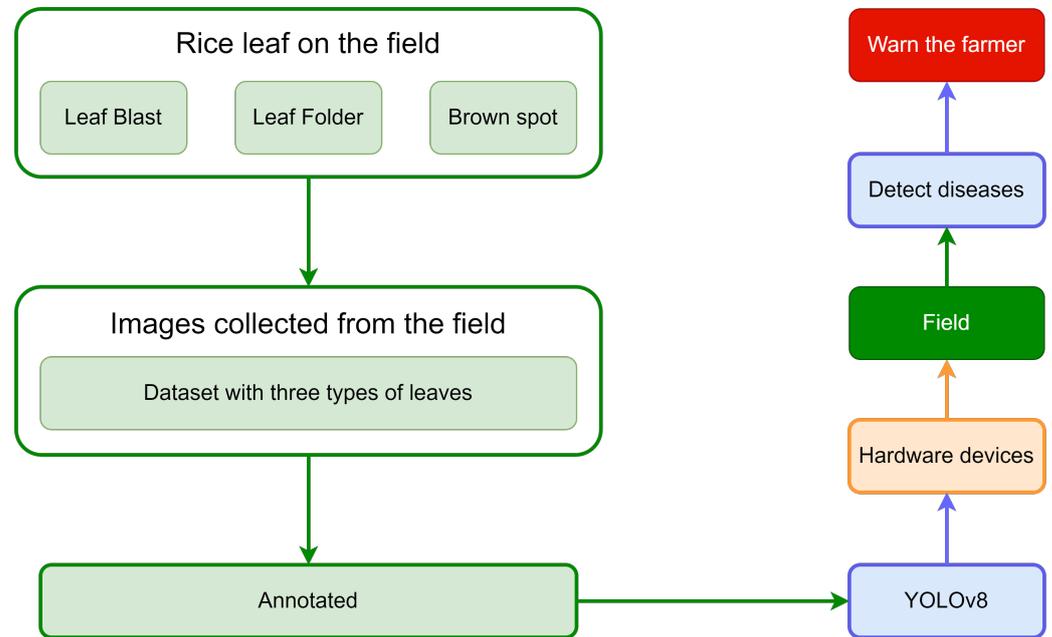


Figure 2. The system of detecting rice leaf diseases.

2.2. System Hardware Design

Figure 3 illustrates the configuration of the hardware devices responsible for the identification and alerting of *leaf folders*, *leaf blasts*, and *brown spots* on rice leaves. The system includes 1 micro-controller, 1 USB camera, and 1 GSM module (Table 1). First, the camera is connected to a micro-controller to capture real-time images of the rice plant on the field. The AI algorithm (implemented in micro-controller) detects and identifies plant diseases. The detection outcomes are then transmitted to the GSM module, which subsequently sends warning messages to the farmers.

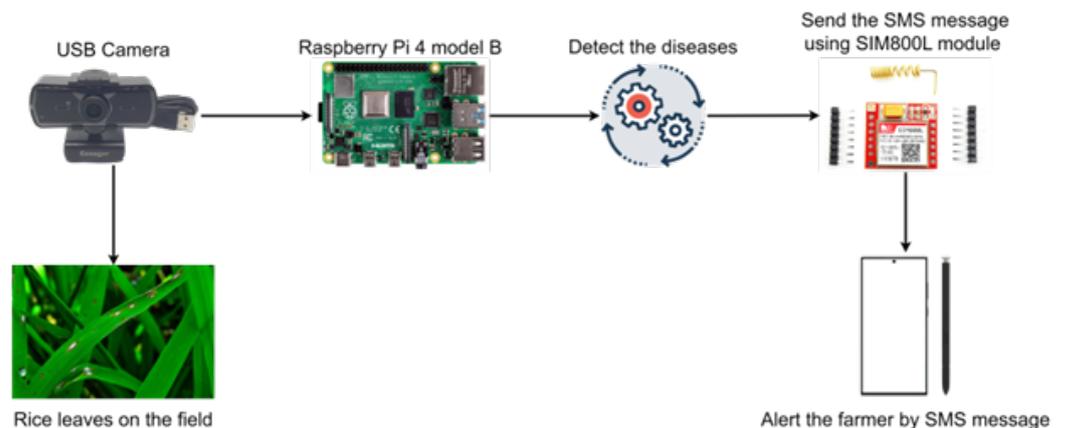


Figure 3. Auto-detecting rice disease system with Raspberry Pi 4 model B.

Table 1. Main features of parts in system

No.	Name of Main Part	Specifications
1	USB Camera Essager C3	<ul style="list-style-type: none"> Resolution: 1920 × 1080; Frame rate: 30 fps; Pixel: 2 mp.
2	Raspberry Pi 4 model B	<ul style="list-style-type: none"> Chipset: Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8 GHz; 8 GB LPDDR4-3200 SDRAM; 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE Gigabit Ethernet; 2 USB 3.0 ports; 2 USB 2.0 ports; 40 pin GPIO header; 5 V DC via USB-C connector (minimum 3 A).
3	SIM 800 L	<ul style="list-style-type: none"> Input voltage: 3.7–4.2 V; GSM antenna connector: IPX and ordinary wire; SIM type: Micro SIM; Allows calls, SMS/MMS sending; Integrated protocols for internet connection (TCP, UDP, HTTP, IP).

2.3. Data Preparation

2.3.1. Image Collection

The experiments are conducted on the rice field at the Vietnam National University of Agriculture (latitude: 21.001792, longitude: −254.068270), Hanoi, Vietnam. A total of 1634 images of rice plants were collected in the experimental rice field with some type of rice variance, such as *DCG66*, *DCG72*, *DH15*, or *Hat ngoc 9*. The dataset contains images of three common diseases in rice plants namely *leaf folder*, *leaf blast*, and *brown spot*. The images were captured under various conditions: different weather conditions (sunny, cloudy, rainy); different environment light. All these images of the rice plant diseases were confirmed manually by agricultural experts.

2.3.2. Dataset Splitting

Our final dataset includes 3175 images and is divided into three parts: a training set, validation set, and test set. The training set includes 2608 images, the validation set includes 326 images, and the test set includes 241 images. The number of samples in each class are 1231 for the leaf folder, 1377 for the blast, and 1237 for the brown spot. One important note is that the images from our test set do not overlap with the images from the validation and training sets; the same is true of the validation set. Our dataset contains small images with the sizes of 150 pixels × 150 pixels and 106 pixels × 200 pixels to large images with the sizes of 3024 pixels × 4032 pixels and 4312 pixels × 5760 pixels.

2.3.3. Data Augmentation

In YOLOv8, different augmentation methods are implemented for the training process; the value of these methods can be adjusted in the default.yaml file. Some of these methods include horizontal and vertical flipping, translation, mosaic, etc. One thing that must be noted is that the augmentation process does not create additional images. Instead, it is applied straightforwardly during each epoch to the dataset. This approach can generate a greater number of unique examples rather than a fixed number of images. Consequently, the model can learn from a significantly larger amount of data. As a result, the augmentation methods can lead to better performance of the model and help the model adapt better to various conditions. Table 2 gives the augmentation hyperparameters in our model.

Table 2. Augmentation hyperparameters table.

No.	Name	Function
1	Hsv_h	0.015
2	Hsv_s	0.7
3	Hsv_v	0.4
4	Translate	0.1
5	Scale	0.5
6	Flipud	0.5
7	Fliplr	0.5
8	Mosaic	1.0

2.4. YOLOv8 Architecture

2.4.1. Overview

YOLOv8 is one of the newest YOLO versions along with YOLO-NAS. Some honorable mentions of earlier versions of YOLO are YOLOv1, YOLO9000v2 [26], YOLOv3 [27], YOLOv4 [28], YOLOv5, YOLOv6 [29], YOLOv7 [30], etc. YOLOv8 itself contains different versions of architecture including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The lightest model is YOLOv8n, and the heaviest model is YOLOv8x. The differences that make one version heavier than another are the convolutional kernel and the feature extraction number. It is important to note that the YOLOv8 version and the YOLOv5 version come from the same developer, Ultralytics. This is the main reason for the similarity of the YOLOv8 architecture and the YOLOv5 architecture. In this project, YOLOv8n is investigated for the rice plant disease detection problem because it has satisfied our requirements for accuracy, inference speed, and lightness. Figure 4 depicts the YOLOv8 architecture diagram.

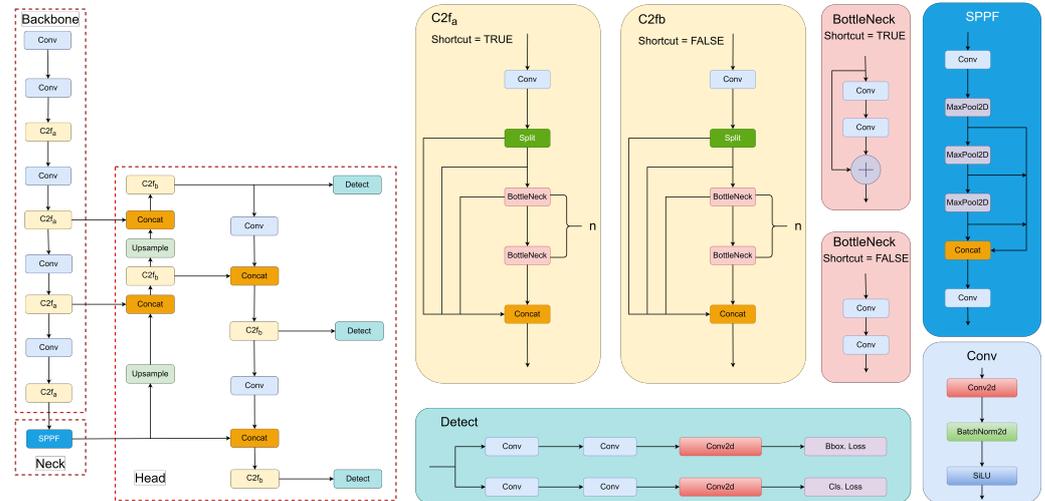


Figure 4. YOLOv8 architecture diagram.

2.4.2. Backbone

YOLOv8 has many similarities to YOLOv5. The C3 module in YOLOv5 is replaced in YOLOv8 with C2f. The C2f idea is the fusion between the C3 module of YOLOv5 and the ELAN model idea. This idea of the ELAN block [31] was influenced by the CSP strategy and VoVNet [32]. The two main blocks of the backbone are C2f and Conv. The C2f block contains a split block, bottleneck, a Conv Block, and a concatenat layer. The number of bottleneck layers is based on the position of the C2f block in the backbone and the depth of the architecture. In the first C2f block, the number of bottleneck layers will be $n = 3 \times d$; for the second and the third, it will be $n = 6 \times d$; and, finally, in the last C2f, the value will be $n = 3 \times d$. One important thing to note is that the feature map of the second and

third C2f blocks will be concatenated with the head part. The Conv Block contains Conv2d, BatchNorm2d, and SiLU.

2.4.3. Neck

Same as in YOLOv5, YOLOv8 uses SPPF in the neck part. The neck part's mission is to perform feature fusion. Feature fusion combines features from the backbone and forwards them into the head part. SPPF is an improved version of SPP. The advantage of SPP is that it allows the model to effectively capture and encode information from objects of various sizes within an image, regardless of their spatial locations. In SPP, max-pooling with different kernel sizes will be applied to each channel in the feature maps of the backbone to create three feature maps with different sizes. This feature map will then be produced into a fixed-length vector. One of the two main differences between SPP and SPPF is the kernel size of the max-pooling layer; while SPP uses different kernel sizes, as mentioned above, SPPF uses the same kernel size for each layer. The second difference is that, instead of applying three max-pooling layers in parallel, SPPF positions these max-pooling layers in series. These differences have made SPPF much faster than SPP.

2.4.4. Head

The main blocks in the head structure are Conv Block, C2f Block, Upsample, and Concat. Note that the C2f of the head part will not contain shortcuts for the bottleneck block. One of the changes in YOLOv8, compared to other versions, is the shift to Anchor-Free instead of Anchor-Based, as seen in the older version. The second change is the use of Decouple-Head. In the older versions of Object Detection, such as Faster R-CNN and YOLO, the Localization and Classification mission is processed on the same branch of the head. This may lead to a problem because Classification needs discriminative features, while Localization needs features containing information on boundary regions. The difference in the features of these tasks is referred to as task conflict. To avoid this, the author of YOLOv8 has separated these tasks into two different branches. However, this led to a task misalignment problem. To solve this, the authors of YOLOv8 have applied TAL in the TOOD paper [33], which is a Label Assignment strategy. YOLOv8 uses TAL to measure the alignment level of Anchor, and it is given in Equation (1):

$$t = s^\alpha \times u^\beta \quad (1)$$

where s and u are the classification score and IoU score, respectively. Alpha and beta are the numbers used to adjust the impact of the two tasks in the Anchor alignment metric, and t is the alignment metric. Depending on the value of t , positive samples and negative samples will be separated for training through the loss function.

2.5. Metrics

In this study, evaluation metrics including Precision (2), Recall (3), mAP (mean Average Precision) (4), and F1-score (5) were used. They are given in Equations (2)–(5), respectively.

$$precision = \frac{TP}{TP + FP} \times 100\% \quad (2)$$

$$recall = \frac{TP}{TP + FN} \times 100\% \quad (3)$$

$$mAP = \frac{\sum_{k=1}^N P(k)R(k)}{C} \quad (4)$$

$$F_{1-score} = 2 \times \frac{precision \times recall}{precision + recall} \quad (5)$$

where TP is the number of exactly identified rice leaves, FP is the number of backgrounds misidentified as the target leaves, FN represents an unspecified number of leaf targets, C represents the number of target categories of rice leaves, N means the number of IOU thresholds, k is the IOU threshold, $P(k)$ is the precision, and $R(k)$ is the recall.

2.6. Loss

2.6.1. Classification Loss

The loss function of YOLOv5 is divided into 3 subfunctions: class loss (BCE with logits loss), objectness loss (BCE with logits loss), and location loss (CIoU loss). The loss function is expressed in Equation (6).

$$Loss = l_1 L_{cls} + l_2 L_{box} + l_3 L_{dfl} \tag{6}$$

2.6.2. Loss Calculation

With BCE with logits loss, this loss function combines a sigmoid layer and BCE Loss into a single layer. This version is more numerically stable than using sigmoid alone followed by BCE Loss because, by combining operations into one class, the logsum-exp trick for numerical stability is used. The formula of BCE Loss with logits in the multi-label classification case can be demonstrated in Equations (7) and (8).

$$\ell_c(x, y) = L_c = \{l_{1,c}, \dots, l_{N,c}\}^T \tag{7}$$

$$L_{n,c} = -w_{n,c} \left[\begin{array}{l} p_c y_{n,c} \cdot \log \sigma(x_{n,c}) + \\ + (1 - y_{n,c}) \log(1 - \sigma(x_{n,c})) \end{array} \right] \tag{8}$$

where:

- c : class number, in this case, $c > 1$, otherwise $c = 1$;
- p_c : weight of the positive answer for class c ;
- N : the batch size;
- n : number of samples in a batch;
- $y_{n,c}$: positive class relates to that logit, while $1 - y_{n,c}$ represents that for a negative class.

2.6.3. Box Loss

CIoU or Complete IoU loss [34] is a loss function that was created to address the limitations of earlier versions of IoU loss functions, such as GIoU and IoU, and aims to provide a more accurate bounding box regression. CIoU is computed in Equation (9).

$$L_{CIoU} = 1 - IoU + \frac{p^2(b, b^{st})}{c^2} + \alpha v \tag{9}$$

While p indicates the Euclidean distance, α and v are used to calculate the discrepancy in the width-to-height ratio.

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{10}$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{st}}{h^{st}} - \arctan \frac{w}{h} \right)^2 \tag{11}$$

2.6.4. DFL Loss

According to the authors in [35], inside each image, the target object could suffer from various conditions, such as shadow, blur, occlusion, or the boundary being ambiguous (being covered by another object). So, for these reasons, the authors of [35] pointed out that the ground-truth labels are sometimes not trustworthy. To address this problem, DFL was created to force the network to quickly concentrate on learning the probabilities associated

with values in the surrounding continuous regions of the target bounding boxes. The DFL is computed in Equation (12).

$$DFL(S_i, S_{i+1}) = -((y_{i+1} - y) \log(S_i) + (y - y_i) \log(S_{i+1})) \tag{12}$$

where y_{i+1} and y_{i-1} are the two closest values to y or, in other words, the ground-truth bounding box. S_i and S_{i+1} can be described as the probability for y_{i+1} and y_{i-1} .

2.7. Loss

2.7.1. Efficient IoU Loss

In [36], the experimental result has pointed out the improvement in accuracy and convergence speed compared to the other versions. However, the authors of [36] have spotted one big disadvantage of CIoU [34], which is the parameter v , in which:

- v does not represent the relation between w and w^{st} or h and h^{st} ; instead, it represents the difference between their aspect ratios. In cases where $w = kw^{st}$, we will have $v = 0$, which is inconsistent with reality. Because of this, the loss function will only try to increase the similarity of the aspect ratio, rather than decrease the discrepancy between (w, h) and (w^{st}, h^{st}) ;
- The gradient of v with respect to w and h can be demonstrated as:

$$\frac{\partial v}{\partial w} = \frac{8}{\pi^2} \left(\arctan \frac{w^{st}}{h^{st}} - \arctan \frac{w}{h} \right) * \frac{h}{w^2+h^2}, \tag{13}$$

$$\frac{\partial v}{\partial h} = -\frac{8}{\pi^2} \left(\arctan \frac{w^{st}}{h^{st}} - \arctan \frac{w}{h} \right) * \frac{w}{w^2+h^2};$$

$$\frac{\partial v}{\partial w} = -\frac{h}{w} \frac{\partial v}{\partial h} \tag{14}$$

With the second equation, it is easy to see that if w is increased, then h will decrease and vice versa. According to the author, it is unreasonable when $w < w^{st}$ and $h < h^{st}$ or $w > w^{st}$ and $h > h^{st}$. To solve the above problem, the authors have suggested a new version of IoU loss, which is EIoU. The loss function is computed in Equation (15).

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b, b^{st})}{(w^c)^2 + (h^c)^2} + \frac{\rho^2(w, w^{st})}{(w^c)^2} + \frac{\rho^2(h, h^{st})}{(h^c)^2} \tag{15}$$

where w^c and h^c are the width and height of the smallest enclosing box covering two boxes.

2.7.2. Alpha-IoU Loss

Alpha-IoU loss created by [37] is a new family of losses based on the fusion of IoU loss functions and the power parameter α . There are two important properties of alpha-IoU that should be mentioned:

- The loss re-weighting: $w_{lr} = L_{\alpha-IoU} / L_{IoU} = 1 + (IoU - IoU^\alpha) / (1 - IoU)$ with $w_{lr}(IoU = 0) = 1$ and $\lim_{IoU \rightarrow 1} w_{lr} = \alpha$;
- $L_{\alpha-IoU}$ will adaptively down-weight and up-weight the relative loss of all objects according to their IoUs when $0 < \alpha < 1$ and $\alpha > 1$, respectively. When $\alpha > 1$, the reweighting factor increases monotonically with the increase in IoU (w_{lr} decays from 1 to α). In other words, with $\alpha > 1$, the model will focus more on one high-IoU object;
- The loss gradient reweighting: $w_{\nabla r} = |\nabla_{IoU} L_{\alpha-IoU}| / |\nabla_{IoU} L_{IoU}| = \alpha IoU^{\alpha-1}$ with the turning point at $IoU = \alpha^{\frac{1}{1-\alpha}} \in (0, \frac{1}{e})$ when $0 < \alpha < 1$ and $IoU = \alpha^{\frac{1}{1-\alpha}} \in (\frac{1}{e}, 1)$ if $\alpha > 1$. When $\alpha > 1$, the above reweighting factor $w_{\Delta r}$ increases monotonically with the increase in IoU, while decreasing monotonically with the increase in IoU when $0 < \alpha < 1$. In other words, $L_{\alpha-IoU}$ with $\alpha > 1$ helps detectors learn faster on high-IoU objects.

2.7.3. Alpha-EIoU Loss

In this paper, we propose a method that replaces the original Box Loss function of YOLOv8 with the combination of the EIoU loss and alpha-IoU loss. The alpha-EIoU loss function is given in Equation (16).

$$L_{\alpha-EIoU} = 1 - IoU^\alpha + \frac{\rho^2(b, b^{st})}{(w^c)^{2\alpha} + (h^c)^{2\alpha}} + \frac{\rho^{2\alpha}(w, w^{st})}{(w^c)^{2\alpha}} + \frac{\rho^{2\alpha}(h, h^{st})}{(h^c)^{2\alpha}} \tag{16}$$

$\alpha = 3$ as recommended by the author in [37].

3. Results and Discussion

3.1. Parameter Setting

The input size of the image is changed to 640 pixels \times 640 pixels, in terms of optimizers, we choose the Stochastic Gradient Descent, the momentum for the Stochastic Gradient Descent is set to 0.937, the initial learning rate and the final learning rate is set by default and equal to 0.01 and 0.0001, respectively. Finally, the dataset is trained within 300 epochs with batch size equal to 16, the figures for the parameter can be seen in the Table 3.

Table 3. Parameter Settings.

Batch size	16
Optimizer	SGD
Epoch	300
Initial learning rate	0.01
Final learning rate	0.0001
Momentum	0.937

3.2. Evaluation of the Proposed Method

The final result of rice disease detection using YOLOv8 with the loss function change is shown in Table 4 and the PR-Curve. This result will be compared with the YOLOv8 model without the change in loss function. This result was achieved in the test set with a total number of 95 leaves containing *folder* disease, 95 leaves containing *blast* disease, and 84 leaves containing *brown spot* disease. This result was achieved by using the nano version of YOLOv8, which is the YOLOv8n model.

The PR-Curve of YOLOv8 is given in Figure 5.

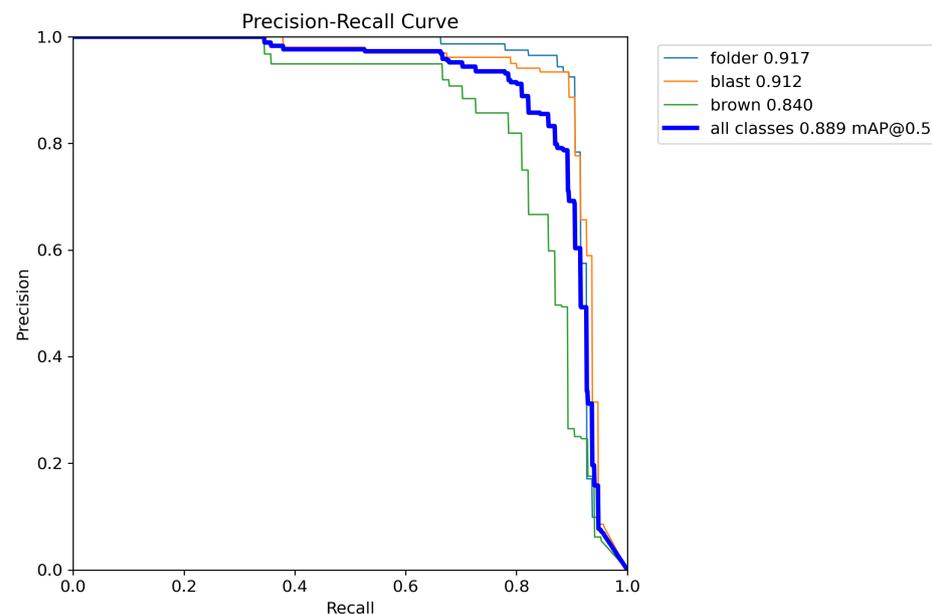


Figure 5. PR-Curve of YOLOv8.

Table 4. The final result of rice disease detection using YOLOv8 with loss function change.

YOLOv8	PRECISION	RECALL	F1	mAP@50
All	89.6	83.5	86.4	88.9
Leaf Folder	93.9	88.4	91.1	91.7
Leaf Blast	86.7	90.5	88.6	91.2
Brown Spot	88.2	71.5	79.0	84.0

The PR-Curve of YOLOv8 with alpha-EIoU loss is given in Figure 6.

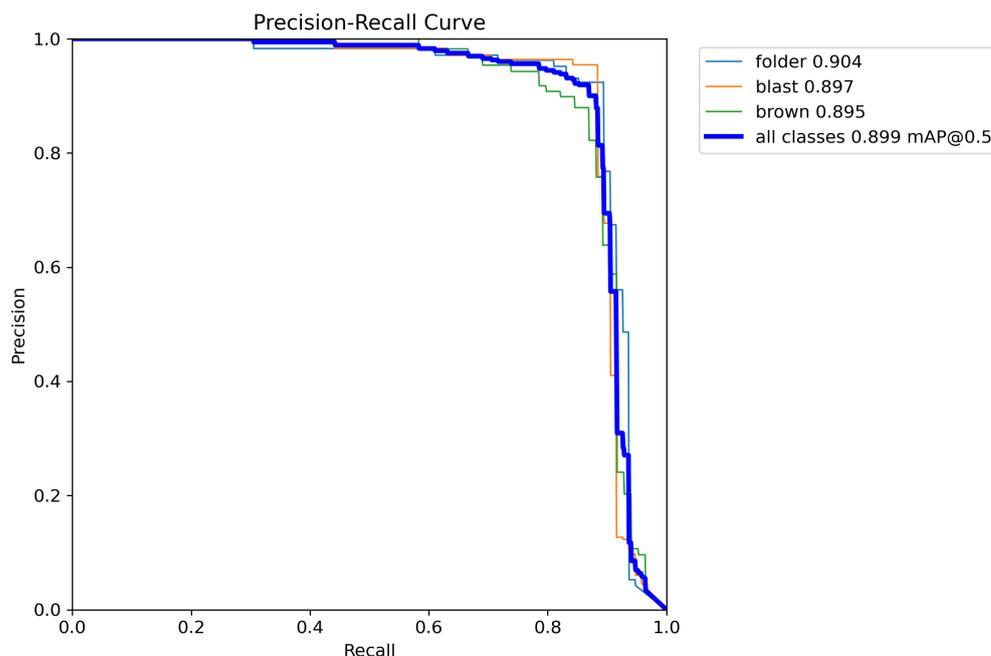


Figure 6. PR-Curve of YOLOv8.

Figure 7 depicts the inference results of our proposed model. Take note that the red, pink, and orange bounding boxes represent the folder, blast, and brown, respectively. Based on the given results of both YOLOv8 and YOLOv8 with the alpha-EIoU loss, the accuracy of the modified loss function has increased the model accuracy up to 1%, which is significant. In terms of other metrics, the proposed method of using alpha-EIoU also yields a better result. In precision and recall, the proposed method achieved 90% and 84.4%, respectively, while YOLOv8 without the modification to the Box Loss function achieved 89.6% and 83.5%. It can be seen that the improvements in these metrics for the proposed method increase from 0.4% to 0.9%. In order to consider both economic factors and the precision factors, the F1-score is known as an important parameter in detecting diseases. Our proposed method achieves a higher F1-score (87.1%) than that of the original method (86.4%). However, if we look at the specific accuracy value of each class, there are still some improvements that need to be made in order to improve our proposed model. The accuracy of the folder disease is 91.7% when predicted by the original method; however, the proposed method only achieved 90.4%. The same can be seen in the blast disease, where the value of mAP@0.5 is 91.2% for the conventional way and only 89.7% for the proposed method. The proposed method greatly increased the accuracy of brown spot disease, in which, the alpha-EIoU method provides an accuracy of 89.5% compared to 84% for the CIoU method, which is more than a 5% increase in accuracy. Compared to the algorithm of YOLOv5 in [22,23], the accuracy of our proposed model on leaf blast has reached 89.7% compared to 69.4% in the study in [22] and 80.3% proposed in [23]. The efficient disease detection of brown leaf spot of our proposed model is also given in Table 5. It is shown that the accuracy of our proposed model on the brown spot has reached 89.5%. This is much better than that of

YOLOv5, used in [23], which gives only a 55.4% accuracy. As depicted in the pictures of bacterial blight in [22], the authors of that study have mistaken the *leaf folder* disease for bacterial blight; therefore, this shows that our model's accuracy on the *leaf folder*, which is 90.4%, is much better than the 65% from [22]. In the comparison of the proposed method with one of the state-of-the-art models, YOLOv7, in the paper [38], written by the same author as that of [18], our method increases the accuracy of the leaf blast class by more than 19%. In terms of leaf folder, which the author mistook for bacterial blight, our model achieves 90.4% compared to only 69.9%. For the cases of precision and recall, our method only achieves 90% and 84.4%, respectively, and does not surpass the values of 1 and 0.92 for YOLOv5 [18] and YOLOv7 [38]. However, our proposed method reaches 87.1% compared to 77% and 76% of the two other methods for the F1-score.

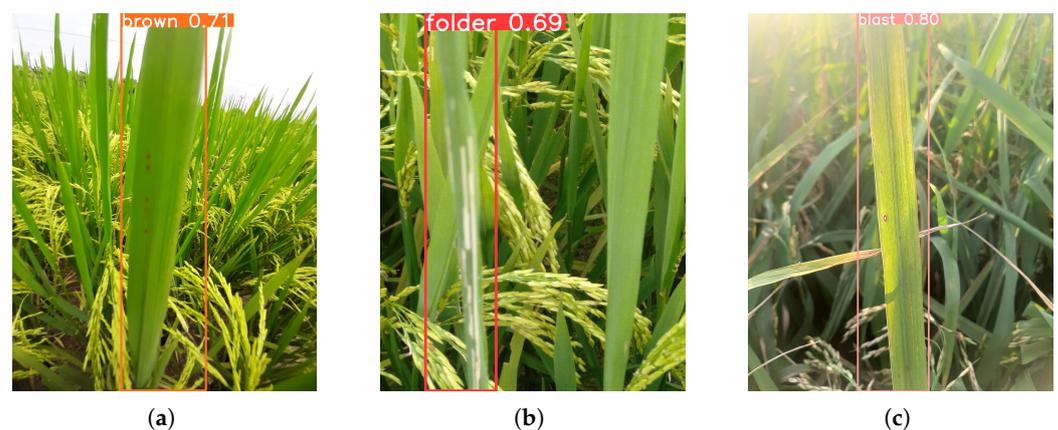


Figure 7. Inference results on rice leaf diseases: (a) inference result on brown spot disease, (b) inference result on leaf folder disease, and (c) inference result on leaf blast disease.

Table 5. YOLOv8 with alpha-EIoU loss results.

YOLOv8	Precision	Recall	F1	mAP@50
All	90.0	84.4	87.1	89.9
Leaf Folder	93.1	85.2	89.0	90.4
Leaf Blast	86.2	88.4	87.3	89.7
Brown Spot	90.6	79.8	84.9	89.5

The complexity and the FPS of our proposed method for YOLOv8n, YOLOv5n, and the YOLOv7-tiny method provided by [39–41] can be seen in Table 6:

Table 6. Computational complexity of YOLOv8n, YOLOv7-tiny, and YOLOv5n.

	Size (pixels)	Params (M)	FLOPs (B)	NVIDIA Jetson AGX Orin (FPS)
YOLOv8n	640	3.2	8.7	383
YOLOv5n	640	1.9	4.5	370
YOLOv7-tiny	640	6.2	13.8	290

As can be seen in the table, YOLOv8n and YOLOv5 are clearly more suitable to implement on low-cost devices. While YOLOv8n's params and flops are higher than those of YOLOv5n, YOLOv8 is still able to yield a better performance. Because of these reasons, our proposed method is a suitable option for low-cost devices. This will further benefit the economic aspect, especially as it requires a large number of devices when being utilized on large-scale rice fields.

For the application of our model, we implement it on Raspberry Pi 4 Model B, as a Single-Board Computer (SBC), supporting a variety of packages and software libraries

across different programming languages. In this study, we use Python packages, such as Serial and Ultralytics. The Serial library was used to enable control of the GPIO pins with the SIM800L module on Raspberry Pi. The Ultralytics library was used to run the YOLOv8 model. Raspberry Pi is a low-cost device, so it took more time to process the AI model. Therefore, in this project, we used YOLOv8n (the lightest version of the YOLOv8 model) so that the Rpi could handle an average processing time of about 1500 ms per frame.

In this research, the prediction results of our approach have been evaluated by the agricultural experts at the Vietnam National University of Agriculture, Hanoi, Vietnam. Despite some mistakes in the prediction process, our proposed model still has great potential to warn farmers of infected regions in their rice fields (Figure 8). The performance of our hardware system, according to experts, is sufficient to apply to real-world scenarios.

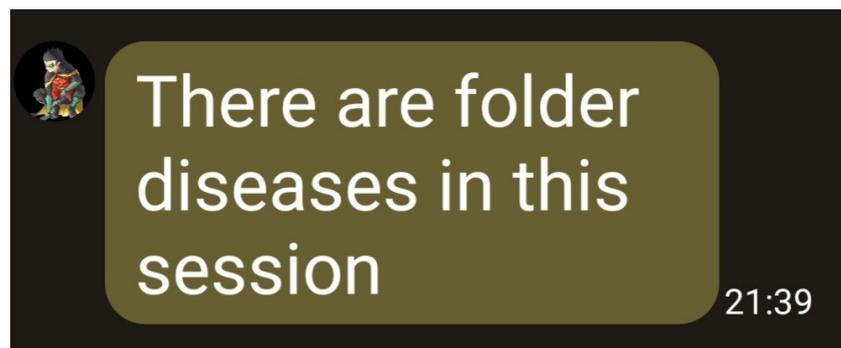


Figure 8. The SMS message sent to farmers.

3.3. Discussion

As mentioned above, while our method can provide better accuracy for all of the combined classes, our method seems to have drawbacks in detecting folder disease and leaf blast disease. In terms of the model complexity, although our method has been applied to the lightest version of YOLOv8, which is YOLOv8 nano, the change in our method did not further reduce the complexity of the original model or increase it. In the case of images captured in different weather conditions, our dataset does not contain images captured in heavily windy conditions, a case that can make the images extremely blurry. Adding images in windy weather can greatly enhance our model performance in real-world scenarios. For the folder class, in extreme lighting conditions, a healthy leaf can be easily mistaken for a diseased leaf, which can lead to a reduced accuracy of the model.

In future research works, we intend to improve our methods in three terms, namely, (1) deep learning techniques, (2) practical application techniques, and (3) the scale of the dataset. For the first term, we would like to test replacing the classification loss function of the original YOLOv8 with the Varifocal Loss Function [42] and train the model on different learning rate schedulers, as inspired by [43], to further increase the accuracy of the proposed model. In the practical aspect, our proposed system may be implemented on a real agricultural robot that can automatically move in the field for early detection and prevention of rice leaf diseases in Vietnam. With high accuracy detection of rice diseases, our proposed method may improve the effectiveness of the rice disease management systems. Our proposed system reduces the human resources and the time needed for rice disease detection; therefore, it can easily diagnose the diseases at an early stage. Moreover, this approach reduces the pesticide residue in the final product of rice and decreases the environmental impact of rice cultivation. Although we only apply the proposed method to three different types of disease, e.g., leaf folder, leaf blast, and brown spot, the number of disease classes and the number of images in our dataset can be further extended. In terms of large datasets, this can easily be performed because our modification to the model does not affect the learning process on either large-scale or small-scale datasets. In addition, in this research, we do not change the optimizer of the original YOLOv8 version, which is the Stochastic Gradient Descent. This is an optimization algorithm that is commonly

used in the case of large-scale datasets. In further extending the number of classes, the new diseases might contain relatively small disease marks. This could be a challenge, however, in this research, our prediction accuracies on the brown spot disease and the leaf blast disease, characterized by the smallest disease marks, are relatively good, which bodes well for future improvements.

4. Conclusions

Our paper introduces a new approach using the YOLOv8 architecture with the YOLOv8n model and the alpha-EIoU loss function to identify the disease of rice plants with an accuracy is up to 89.9%, which is a better result compared to 62% in [23], 80.7% in [22], and 82.8% in [38]. This result was achieved on 241 photos, and this is a relatively good result for further applications in the real world. The experiment system worked well and provided warnings immediately after detection to farmers with knowledge equivalent to an expert; this will help farmers have quick responses to deal with the diseases. With this result, we implement the achieved model into the low-cost hardware that we designed to provide farmers with a solution to detect diseases in a timely manner in the rice field. Our research also provides a completely new rice leaf disease image dataset of the three most common diseases, (*leaf blast*, *leaf folder*, and *brown spot*), in Vietnam. Therefore, it may help researchers reduce the time and money required to obtain such data in future research. It is proved that our proposed modification of the YOLOv8 framework achieved better performance than some other state-of-the-art methods.

Author Contributions: Conceptualization, H.T.N. (Hoc Thai Nguyen) and H.T.N. (Huong Thanh Nguyen); methodology, T.D.B. and H.T.N. (Huong Thanh Nguyen); validation, D.C.T.; formal analysis, H.T.N. (Hoc Thai Nguyen) and T.D.B.; literature review, D.C.T., K.G.D. and A.T.M.; project administration, D.C.T. and T.D.B.; supervision, T.D.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Hanoi University of Science and Technology (HUST) under grant number: T2023-PC-069.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors gratefully acknowledge the support of the Industrial Instrumentation & IoT Laboratory at the School of Electrical and Electronic Engineering, HUST, and other support from experts at the Vietnam National University of Agriculture, Vietnam.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
IoT	Internet of Things
CNN	Convolutional Neural Network
SVM	Support vector machines
GSM	Global system for mobile communications
mAP	Mean Average Precision

References

1. Fairhurst, T.; Dobermann, A. Rice in the global food supply. *World* **2002**, *5*, 454349–511675.
2. Wijayanto, A.K.; Junaedi, A.; Sujaswara, A.A.; Khamid, M.B.; Prasetyo, L.B.; Hongo, C.; Kuze, H. Machine Learning for Precise Rice Variety Classification in Tropical Environments Using UAV-Based Multispectral Sensing. *AgriEngineering* **2023**, *5*, 2000–2019. [[CrossRef](#)]

3. de Oliveira Carneiro, L.; Coradi, P.C.; Rodrigues, D.M.; Lima, R.E.; Teodoro, L.P.R.; de Moraes, R.S.; Teodoro, P.E.; Nunes, M.T.; Leal, M.M.; Lopes, L.R.; et al. Characterizing and Predicting the Quality of Milled Rice Grains Using Machine Learning Models. *AgriEngineering* **2023**, *5*, 1196–1215. [[CrossRef](#)]
4. Paidipati, K.K.; Chesneau, C.; Nayana, B.M.; Kumar, K.R.; Polisetty, K.; Kurangi, C. Prediction of rice cultivation in India—Support vector regression approach with various kernels for non-linear patterns. *AgriEngineering* **2021**, *3*, 182–198. [[CrossRef](#)]
5. Rahman, H.; Sharifee, N.H.; Sultana, N.; Islam, M.I.; Habib, S.A.; Ahammad, T. Integrated Application of Remote Sensing and GIS in Crop Information System—A Case Study on Aman Rice Production Forecasting Using MODIS-NDVI in Bangladesh. *AgriEngineering* **2020**, *2*, 264–279. [[CrossRef](#)]
6. Krichen, M. Convolutional Neural Networks: A Survey. *Computers* **2023**, *12*, 151. [[CrossRef](#)]
7. Naved, M.; Devi, V.A.; Gaur, L.; Elngar, A.A. (Eds.) E-Learning Modeling Technique and Convolution Neural Networks in Online Education. In *IoT-Enabled Convolutional Neural Networks: Techniques and Applications*; River Publishers: Aalborg, Denmark, 2023.
8. Uoc, N.Q.; Duong, N.T.; Thanh, B.D. A novel automatic detecting system for cucumber disease based on the convolution neural network algorithm. *GMSARN Int. J.* **2022**, *16*, 295–302.
9. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant disease detection and classification by deep learning. *Plants* **2002**, *8*, 468. [[CrossRef](#)] [[PubMed](#)]
10. Agbulos, M.K.; Sarmiento, Y.; Villaverde, J. Identification of leaf blast and brown spot diseases on rice leaf with yolo algorithm. In Proceedings of the IEEE 7th International Conference on Control Science and Systems Engineering (ICCSSE), Qingdao, China, 30 July–1 August 2021; pp. 307–312. [[CrossRef](#)]
11. Zhang, K.; Wu, Q.; Liu, A.; Meng, X. Can deep learning identify tomato leaf disease? *Adv. Multimed.* **2018**, *8*, 6710865. [[CrossRef](#)]
12. Liu, J.; Wang, X. Tomato diseases and pests detection based on improved yolo v3 convolutional neural network. *Front. Plant Sci.* **2020**, *11*, 898. [[CrossRef](#)]
13. Sharma, R.; Kukreja, V.; Kaushal, R.K.; Bansal, A.; Kaur, A. Rice leaf blight disease detection using multi-classification deep learning model. In Proceedings of the 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 13–14 October 2022; pp. 1–5. [[CrossRef](#)]
14. Deng, R.; Tao, M.; Xing, H.; Yang, X.; Liu, C.; Liao, K.; Qi, L. Automatic diagnosis of rice diseases using deep learning. *Front. Plant Sci.* **2021**, *12*, 701038. [[CrossRef](#)]
15. Altinbilek, H.F.; Kizil, U. Identification of paddy rice diseases using deep convolutional neural networks. *Yuz. Yil Univ. J. Agric. Sci.* **2021**, *32*, 705–713. [[CrossRef](#)]
16. Shrivastava, V.K.; Pradhan, M.K.; Minz, S.; Thakur, M.P. Rice plant disease classification using transfer learning of deep convolution neural network. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 631–635. [[CrossRef](#)]
17. Chaudhari, D.J.; Malathi, K. A survey on rice leaf and seedlings disease detection system. *Indian J. Comput. Sci. Eng. (IJCSE)* **2021**, *12*, 561–568. [[CrossRef](#)]
18. 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](#)]
19. Hasan, M.J.; Mahbub, S.; Alom, M.S.; Nasim, M.A. Rice disease identification and classification by integrating support vector machine with deep convolutional neural network. In Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology, Dhaka, Bangladesh, 3–5 May 2019; pp. 1–6. [[CrossRef](#)]
20. Ramesh, S.; Vydeki, D. Recognition and Classification of Paddy Leaf Diseases Using Optimized Deep Neural Network with Jaya Algorithm. *Inf. Process. Agric.* **2020**, *7*, 249–260. [[CrossRef](#)]
21. Maheswaran, S.; Sathesh, S.; Rithika, P.; Shafiq, I.M.; Nandita, S.; Gomathi, R.D. Detection and Classification of Paddy Leaf Diseases Using Deep Learning (CNN). In *Computer, Communication, and Signal Processing. ICCSP 2022. IFIP Advances in Information and Communication Technology*; Neuhold, E.J., Fernando, X., Lu, J., Piramuthu, S., Chandrabose, A., Eds.; Springer: Cham, Switzerland, 2022; Volume 651. [[CrossRef](#)]
22. Haque, M.E.; Rahman, A.; Junaid, I.; Hoque, S.U.; Paul, M. Rice Leaf Disease Classification and Detection Using YOLOv5. *arXiv* **2022**, arXiv:2209.01579.
23. Jhatial, M.J.; Shaikh, R.A.; Shaikh, N.A.; Rajper, S.; Arain, R.H.; Chandio, G.H.; Bhangwar, A.Q.; Shaikh, H.; Shaikh, K.H. Deep learning-based rice leaf diseases detection using yolov5. *Sukkur IBA J. Comput. Math. Sci.* **2022**, *6*, 49–61. [[CrossRef](#)]
24. Sparks, N.C.A.; Cruz, C.V. Brown Spot. Available online: <http://www.knowledgebank.irri.org/training/fact-sheets/pest-management/diseases/item/brown-spot> (accessed on 5 September 2023).
25. Catindig, J. Rice Leaf Folder. Available online: <http://www.knowledgebank.irri.org/training/fact-sheets/pest-management/insects/item/rice-leaffolder> (accessed on 5 September 2023).
26. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
27. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
28. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
29. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.

30. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
31. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing network design strategies through gradient path analysis. *arXiv* **2022**, arXiv:2211.04800.
32. Lee, Y.; Hwang, J.W.; Lee, S.; Bae, Y.; Park, J. An energy and GPU-computation efficient backbone network for real time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
33. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE Computer Society: Washington, DC, USA, 2021; pp. 3490–3499. [[CrossRef](#)]
34. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [[CrossRef](#)] [[PubMed](#)]
35. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
36. Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [[CrossRef](#)]
37. He, J.; Erfani, S.; Ma, X.; Bailey, J.; Chi, Y.; Hua, X.S. Alpha-IoU: A family of power intersection over union losses for bounding box regression. *Adv. Neural Inf. Process. Syst.* **2022**, *34*, 20230–20242.
38. Haque, M.E.; Paul, M.; Rahman, A.; Tohidi, F.; Islam, M.J. Rice leaf disease detection and classification using lightly trained Yolov7 active deep learning approach. In Proceedings of the Digital Image Computing: Techniques and Applications (DICTA), Port Macquarie, NSW, Australia, 28 November–1 December 2023.
39. Available online: <https://docs.ultralytics.com/models/yolov8/> (accessed on 19 January 2024).
40. Available online: <https://docs.ultralytics.com/models/yolov7/> (accessed on 19 January 2024).
41. Available online: <https://www.stereolabs.com/blog/performance-of-yolo-v5-v7-and-v8> (accessed on 19 January 2024).
42. Zhang, H.; Wang, Y.; Dayoub, F.; Sunderhauf, N. VarifocalNet: An IoU-aware Dense Object Detector. *arXiv* **2020**, arXiv:2008.13367.
43. Loshchilov, I.; Frank, H. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv* **2016**, arXiv:1608.03983.

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.