

# An Accurate Classification of Rice Diseases Based on ICAI-V4

Nanxin Zeng <sup>1</sup>, Gufeng Gong <sup>1,\*</sup>, Guoxiong Zhou <sup>1,\*</sup> and Can Hu <sup>2</sup>

<sup>1</sup> College of Computer & Information Engineering, Central South University of Forestry and Technology, Changsha 410018, China; m15271698243@163.com

<sup>2</sup> Hunan Polytechnic of Environment and Biology, Hengyang 421005, China; hc@hnebp.edu.cn

\* Correspondence: t20090523@csuft.edu.cn (G.G.); t20060599@csuft.edu.cn (G.Z.)

**Abstract:** Rice is a crucial food crop, but it is frequently affected by diseases during its growth process. Some of the most common diseases include rice blast, flax leaf spot, and bacterial blight. These diseases are widespread, highly infectious, and cause significant damage, posing a major challenge to agricultural development. The main problems in rice disease classification are as follows: (1) The images of rice diseases that were collected contain noise and blurred edges, which can hinder the network's ability to accurately extract features of the diseases. (2) The classification of disease images is a challenging task due to the high intra-class diversity and inter-class similarity of rice leaf diseases. This paper proposes the Candy algorithm, an image enhancement technique that utilizes improved Canny operator filtering (the gravitational edge detection algorithm) to emphasize the edge features of rice images and minimize the noise present in the images. Additionally, a new neural network (ICAI-V4) is designed based on the Inception-V4 backbone structure, with a coordinate attention mechanism added to enhance feature capture and overall model performance. The INCV backbone structure incorporates Inception-iv and Reduction-iv structures, with the addition of involution to enhance the network's feature extraction capabilities from a channel perspective. This enables the network to better classify similar images of rice diseases. To address the issue of neuron death caused by the ReLU activation function and improve model robustness, Leaky ReLU is utilized. Our experiments, conducted using the 10-fold cross-validation method and 10,241 images, show that ICAI-V4 has an average classification accuracy of 95.57%. These results indicate the method's strong performance and feasibility for rice disease classification in real-life scenarios.

**Keywords:** rice disease detection; coordinate attention; ICAI-V4; Candy algorithm; involution



**Citation:** Zeng, N.; Gong, G.; Zhou, G.; Hu, C. An Accurate Classification of Rice Diseases Based on ICAI-V4. *Plants* **2023**, *12*, 2225. <https://doi.org/10.3390/plants12112225>

Academic Editor: Ritesh Krishna

Received: 25 April 2023

Revised: 28 May 2023

Accepted: 2 June 2023

Published: 5 June 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

Rice is a crucial food crop globally, serving as the staple food for approximately 50% of the world's population [1]. It holds significant importance in national agricultural production [2]. The efficient and accurate identification of rice diseases is a pressing issue for rice growers, as rice disease is the primary factor affecting both yield and quality. The loss of yield and quality caused by rice disease has a direct impact on rice production, as well as human and animal safety [3]. Given that the safe production of rice is a strategic concern related to social stability and national security, addressing this issue is of utmost importance. The traditional method of identifying rice diseases involves visual inspection by plant protection experts. However, this method is time consuming and relies heavily on subjective judgment, making it difficult to promptly prevent and control rice diseases [4]. With the rapid improvement in computer hardware processing speed and software technology, the utilization of artificial intelligence, image recognition, big data, and deep learning technology has become increasingly prevalent in the field of agriculture, particularly in crop disease diagnosis and recognition. When rice plants are affected by diseases, their physiological structure and morphological characteristics change, resulting in symptoms such as leaf discoloration, decay, and deformation. To ensure higher yield and protect rice from pests and diseases, it is crucial to develop a new method for detecting rice diseases

that can quickly and accurately identify and classify them in the early stages. This will help to maintain the overall quality and yield of rice.

The agricultural sector is witnessing a shift from traditional intensive farming to emerging technologies such as precision agriculture and intelligent agriculture. Precision agriculture utilizes information technology (IT), satellite technology, geographic information systems (GISs), and remote sensing to enhance all functions and services of the agricultural sector [5]. This technology is crucial in improving crop disease and pest control and management, establishing intensive agricultural production management and operation mode, and enhancing the quality of agricultural products. Precision agriculture has begun to rely on various technologies such as mobile applications [6], smart sensors [7], drones [8], cloud computing [9], artificial intelligence (AI) [10], the Internet of Things (IoT) [11], and blockchain [12]. These technologies enable real-time processing and access to data on soil, crop, and weather conditions. Additionally, they allow for the timely detection and reporting of plant health, and provide guidance to farmers on soil management, crop maturity rotation, optimal planting time, and harvest time. The utilization of emerging technologies, such as the Internet of Things, presents vast potential for the advancement of smart and precision agriculture. These technologies allow for the acquisition of real-time environmental data, facilitating more efficient and effective farming practices. The use of IoT devices such as unmanned aerial vehicles with high spatial and temporal resolution can be beneficial for crop management. However, the adoption of smart agriculture has not been as strong as expected due to various challenges such as data acquisition and image processing methods. Khattab et al. have developed an IoT-based monitoring system specifically designed for precision agriculture applications [13]. This system provides environmental monitoring services to ensure that the crop growing environment is optimal and to predict early conditions that may lead to epidemic outbreaks. While the system is capable of obtaining real-time plant images and making predictions, the accuracy of its classification model for plant diseases remains a challenge. Yuki et al. proposed a straightforward image processing pipeline that adheres to the empirical principle of the simultaneous emergence of leaves and tillers in rice morphogenesis [14]. The team utilized an unmanned aerial vehicle to capture field images at very low flight altitudes. They then calculated the number of long blades using the proposed image processing pipeline, which included binarization, skeletonization, and tip detection. This study highlights the potential of the proposed image-based tiller counting method to aid agronomists in efficient and non-destructive field phenotypic analysis. This paper focuses on addressing the challenges faced by image processing and prediction classification models in the context of mature real-time image acquisition technology. Specifically, the paper proposes a solution to process images and classify rice diseases using acquired image data.

Currently, rice disease classification faces several challenges: (1) The collected rice leaf images often contain noise and insignificant edge details. If these images are directly used for training without proper preprocessing, the network's ability to extract rice disease features is greatly weakened, which ultimately leads to low recognition accuracy. (2) Rice exhibits large intraclass diversity and high interclass similarity [15], evident in diseases such as blast and brown spot. Traditional convolutional neural networks struggle with accurate differentiation, thus requiring improvements in the network model to enhance its feature extraction capabilities [16].

In order to improve the accuracy of rice disease recognition, image processing is performed on collected rice disease pictures to address the issue of unclear edge details and significant noise. This step is crucial in the overall process of rice disease recognition, as it allows for a more precise and accurate identification of the disease. Image processing techniques such as image enhancement, shadow removal (edge detection), noise elimination, and feature extraction are employed in capturing disease images of rice plant leaves. These processed images are then fed into a classifier for classification [17]. In their study, Polesel et al. (2000) utilized an adaptive filter that regulates the sharpening path's contribution. This approach allows for contrast enhancement in high-detail areas while minimizing or

eliminating image sharpening in smooth regions. This is referred to as an unsharp masking technique for improving image contrast [18]. Phadikar et al. (2013) proposed a segmentation method that utilizes Fermi energy to distinguish the infected area of a rice pest image from its background. This method employs new algorithms to extract features such as the color, shape, and location of the infected site to characterize disease symptoms. To minimize information loss and reduce the complexity of the classifier, the researchers utilized rough set theory (RST) to select important features [19]. The Canny edge detection algorithm has become increasingly popular in image processing in recent years. Several studies have suggested using the Canny operator to enhance edge information in images and improve their clarity. The algorithm, proposed by John Canny in the 1980s, is a multi-level edge detection method [20]. John Canny focused on identifying the optimal edge, which required detecting edges that closely resembled the actual edge while minimizing noise interference. However, traditional algorithms face difficulty when filtering low-contrast images, as they cannot adaptively determine the filtering threshold or differentiate the target from the background. To address the aforementioned issues, Xuan et al. (2017) proposed an enhanced Canny algorithm. This algorithm utilizes the amplitude gradient histogram to derive two adaptive thresholds and connect edge points to obtain generalized chains. The algorithm calculates the average value of the generalized chains and removes those that fall below it. The image edge detection results are then obtained using a linear fitting method. Experimental results demonstrate that the improved algorithm is more robust to noise and effectively distinguishes the target from the background [21]. In their 2020 paper, Kalbasi et al. introduced an adaptive parameter selection method. This method uses the estimated noise intensity of the input image and the minimum output performance required for the application to select a value for the Canny parameter from a pre-existing configuration table, rather than calculating it at runtime [22]. This implementation of the Canny algorithm is adaptive, ensuring high edge detection performance and noise robustness in various situations. Despite this, the execution time of our proposed Canny algorithm is still lower than that of the latest cutting-edge implementation. Wang et al. (2019) proposed an improved gravitational edge detection algorithm that utilizes gravitational field intensity instead of image gradient, along with the introduction of the gravitational field intensity operator. The algorithm also includes an adaptive threshold selection method based on the mean and standard deviation of image gradient amplitude for images with varying levels of edge information. The gravitational edge detection algorithm, which is relatively simple to implement, has been shown to retain more relevant edge information and exhibit stronger anti-noise ability through empirical data analysis [23]. This paper proposes a Candy image enhancement algorithm (an improvement upon the Canny algorithm) to address the issues of noise and unclear edge details in rice leaf images. The method involves using an ideal high-pass filter and the gravitational edge detection algorithm to extract the edge mask of the rice disease image. Then, a simple detail-enhanced image is produced by combining smooth filtering and a Laplace operator. The proposed method multiplies the output by a mask and adds it to the input image to enhance edge detail. The experimental analysis demonstrates that this method significantly improves the visual effect of the image compared to the original image and homomorphic filtering. Additionally, the method effectively suppresses noise while maintaining image authenticity.

In order to solve the problem of high diversity and similarity among rice diseases, the accuracy of convolutional neural networks in rice disease image classification needs improvement. Lu et al. (2017) introduced a novel approach for identifying rice diseases using deep convolutional neural network (CNN) technology. The proposed model was evaluated using a 10-fold cross-validation strategy and achieved an impressive accuracy of 95.48%, surpassing that of traditional machine learning models. The feasibility and effectiveness of using deep convolutional neural networks to identify rice diseases and insect pests are demonstrated through simulation results of rice disease recognition. However, the success of this approach is contingent upon the availability of large-scale datasets and high-quality rice disease image samples. Additionally, selecting the optimal parameters

still requires a significant number of experiments [24]. Atole et al. (2018) explored the use of deep convolutional neural networks to classify rice plants according to the health of their leaves. They implemented a three-class classifier using transfer learning from the AlexNet deep network, which was able to distinguish between normal, unhealthy, and serpentine plants. The network achieved an accuracy of 91.23% using stochastic gradient descent. These results suggest that commonly used neural networks could still be optimized for rice diseases and insect pests. However, no improvements have been made to tailor the neural network specifically for rice diseases [25]. Zhou et al. (2019) presented a rapid method for detecting rice diseases using a combination of FCM-KM and Faster-R-CNN fusion techniques. This approach not only enhances the recognition accuracy of the Faster R-CNN algorithm, but also reduces the time required for recognition. The method primarily concentrates on detecting diseases in collected images, which may not be suitable for monitoring large-scale rice cultivation. Furthermore, it also identifies diseases by monitoring the collected images [26]. Rahman et al. (2020) utilized state-of-the-art large-scale architectures, such as VGG16 and InceptionV3, which were fine-tuned to detect and identify rice pests and diseases. While the experimental results demonstrated the efficacy of these models on real datasets, they were found to be inadequate in detecting and classifying rice pests and diseases in heterogeneous backgrounds [27]. Wang et al. (2021) presented the ADSNN-BO, an attention-based deep separable Bayesian optimized neural network, as a means of detecting and classifying rice diseases from rice leaf images. The authors suggest that the use of artificial intelligence in the agricultural field can lead to rapid diagnosis and control of plant diseases. However, it is worth noting that their experiments did not utilize public datasets and as such, it may be difficult to promote their findings [28]. Haque et al. (2022) proposed a method for rice leaf disease classification and detection using YOLOv5 deep learning. The YOLOv5 model was trained and evaluated on a dataset of approximately 1500 photos. The use of a large dataset during training resulted in high accuracy [29].

While deep learning has demonstrated success in rice disease recognition, there remains room for improvement in recognition accuracy and the need to minimize model training time. To address these challenges, this paper presents the ICAI-V4 model, which offers the following main contributions:

1. This paper proposes an algorithm for enhancing Candy images. The proposed method involves extracting the edge mask through ideal high-pass filtering and gravitational field intensity operator. The simple detail enhancement image is obtained by combining smooth filtering and a Laplace operator. The resulting image is then multiplied by the mask and added to the input image to obtain a better edge detail enhanced image.
2. This paper proposes the ICAI-V4 model, which incorporates a coordinated attention mechanism into the backbone network. This enhancement is designed to improve the feature capture ability and overall performance of the network model. The INCV backbone structure incorporates both Inception-iv and Reduction-iv structures, while also integrating involution to enhance the network's feature extraction capabilities from a channel perspective. This enables the network to better classify similar images of rice pests and diseases. Leaky ReLU is utilized as an alternative to the ReLU activation function to mitigate the issue of neuron death caused by the latter and enhance the model's resilience.

## 2. Materials and Methods

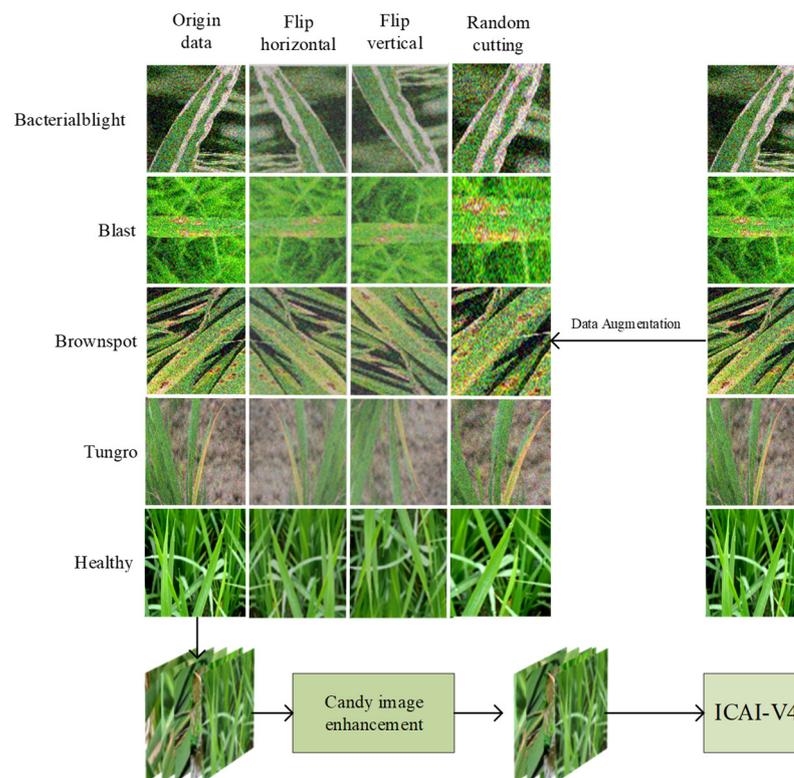
### 2.1. Data Acquisition

The identification and classification of rice diseases heavily rely on data collection. The datasets used in this study were primarily obtained from open-source platforms such as Kaggle (Rice Leaf Diseases Dataset | Kaggle) and Mendeley Data (Rice Leaf Disease Image Samples—Mendeley Data). The Kaggle platform contributed 3355 rice disease images, while the Mendeley Data platform provided 5932 images. To obtain a large number of datasets for network training and address the issue of duplicated images, various techniques such as deletion, horizontal flipping, vertical flipping, and random cropping

were applied after enhancing the collected images. The final result was a dataset of 10,241 images, all saved in jpg format. The distribution and proportion of different types of rice disease images can be found in Table 1. In Figure 1, the process of collecting, transmitting, enhancing, and preprocessing rice images is demonstrated. In order to avoid any noise interference, it is important to enhance the image before inputting it into the recognition network. The dataset included various types of rice diseases such as bacterial blight, blast, brown spot, and tungro disease. The images were evenly distributed and randomly divided into training and test sets at a 9:1 ratio. The training set is utilized to train the model, while the test set is used to evaluate the model's recognition ability. To maintain the objectivity of the results, the test set remains unprocessed. However, the images in the training set are subjected to random rotations, flips, and mirroring to expand the dataset without affecting the results' objectivity. Additionally, all images in this study were normalized to a size of  $299 \times 299$  to expedite the model training process [30].

**Table 1.** Number and proportion of various rice disease images.

Disease Type	Kaggle	Mendeley Data	Expanded	Sum	Percentage (%)
Bacterial blight	1584	935	145	2551	26.70%
Blast	1440	894	289	2531	24.28%
Brown spot	1600	872	569	2845	26.97%
Tungro	1308	654	525	2314	22.05%
Total	5932	3355	1528	10,241	100%



**Figure 1.** Process of acquisition, transmission, enhancement (flip, crop), and preprocessing of four rice disease images.

Our self-built dataset focuses on four primary types of rice diseases: bacterial blight, blast, brown spot, and tungro disease. Each disease is characterized by unique symptoms, which are as follows:

- Rice bacterial blight is caused by the bacterium *Xanthomonas*, which produces water-stained lesions starting from the leaf margin a few centimeters from the tip of the leaf

and spreading to the leaf base. The affected area increases in length and width, and changes color from yellow to light brown due to dryness [19].

- Rice blast disease is characterized by the appearance of green-gray spots on infected leaves, which are surrounded by dark green borders. As the disease progresses, the spots or lesions become oval or spindle-shaped with reddish-brown borders, and some may even become rhombus-shaped. The lesions can expand and merge, ultimately leading to the complete destruction of the leaves.
- Rice brown spot is a fungal disease caused by *Bipolaris oryzae* that primarily damages the aboveground parts of rice plants, particularly the leaves. It is prevalent in all rice-growing regions, especially when there is a shortage of fertilizer and water, leading to poor growth of the rice. It often occurs in combination with rice blight. The disease causes leaf blight, resulting in reduced 1000-grain weight and an increase in empty grains, which adversely affects the yield and quality of rice. Although the harm of disease in rice production is decreasing with improved fertilization and water conservation, certain areas still experience increased incidence of ear blight in late rice due to prolonged seedling age, resulting in significant harvest losses.
- Rice tungro disease causes infected leaves to turn from orange to yellow, with the staining spreading from the tip to the lower part of the leaf. Infected leaves may exhibit a striped appearance and rusty spots, and may contain planthoppers [17].

This section provides a detailed explanation of image enhancement algorithms and rice disease classification models. To aid in comprehension, Table 2 presents a list of variables utilized in the following formula. Table 3 provides an explanation of the abbreviations used in this article. Figure 2 provides a flowchart of the program for this article.

**Table 2.** Variable name and its meaning.

Symbol	Description
$\sigma$	The parameters of the Gaussian filter
$E_x[i, j]$	Approximation of the first partial derivative in the X direction
$E_y[i, j]$	Approximation of the first partial derivative in the Y direction
$G_x$	The image gradient operator in the x direction
$G_y$	The image gradient operator in the y direction
$  M(i, j)  $	Image gradient size
$\theta(i, j)$	The azimuth of the image gradient
$E_{total}$	Intensity of gravitational field
$E_x$	The component of the gradient in the X direction
$E_y$	The component of the gradient in the Y direction
$E$	Gradient magnitude
$\theta$	Gradient azimuth
$E_{ave}$	The average of the magnitude of the gradient
$T_h$	The threshold of a pixel
$H$	Involution kernel height
$W$	Involution kernel width
$K$	kernel size
$\varphi_{i,j}$	An index set in the neighborhood of coordinates $(i, j)$
$X_{\varphi_{i,j}}$	A patch in the feature map contains $X_{i,j}$
$y_c(i, j)$	Coordinate Attention Block output Y
TP	The number of samples that were predicted to be positive
FP	The number of samples that are actually negative, but predicted to be positive
FN	The number of samples that are actually positive, but predicted to be negative
TN	The number of samples that are predicted to be negative

## 2.2. Candy Image Enhancement Algorithm

The quality of rice disease images can be affected by various factors, such as illumination, resulting in issues such as noise and insignificant edge features. These problems can negatively impact the training and testing of the network, making identification and

classification more difficult. To improve the accuracy and reliability of classification after recognition, it is necessary to enhance the image quality. This paper utilizes the Candy image enhancement algorithm to process the images. The algorithm replaces the image gradient used in the traditional Canny operator with gravitational field intensity and implements an adaptive threshold selection method based on standard deviation. The edge enhanced image is obtained by multiplying the initial enhanced image with the mask image, which is obtained through ideal high-pass filtering and gravitational field intensity operator processing. The resulting mask image is then combined with smooth filtering and the Laplacian operator to obtain a simple detail enhanced image. Finally, the edge enhanced image is obtained by adding the mask image multiplied initial enhanced image to the input image. The proposed method results in a visually improved image compared to the original and homomorphic filtering. It enhances edge details while effectively suppressing noise and preserving the authenticity of the image.

Table 3. Explanations for acronyms.

Acronym	Full Term
AI	Artificial Intelligence
CA	Coordinate Attention
FN	False Negative
FP	False Positive
GIS	Geographic Information System
IoT	Internet of Things
IT	Information Technology
TN	True Negative
TP	True Positive

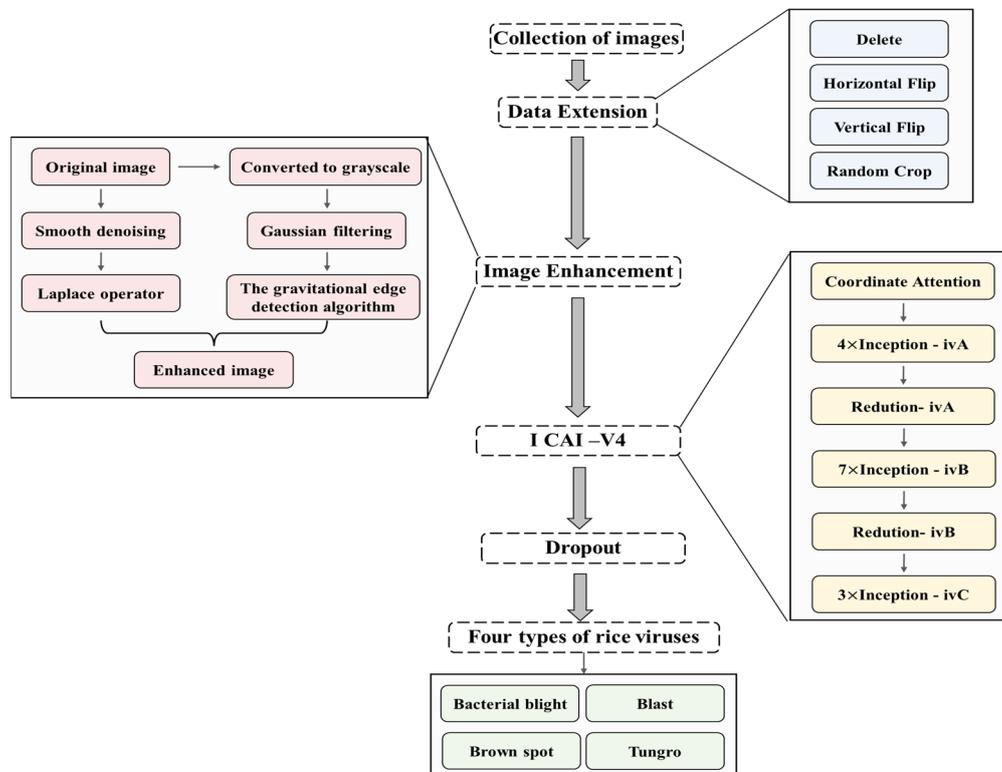


Figure 2. Main flow chart of rice disease classification.

The Canny algorithm has long been used for image edge detection. However, it is known to be sensitive to noise, which often results in the loss of weak edge information while filtering out noise. Additionally, its adaptability to fixed parameters is poor. To

address these issues, Weibin Rong proposed an improved algorithm based on the Canny algorithm [31]. In this paper, the algorithm proposes a new concept, gravitational field strength, to replace the traditional image gradient method. The algorithm then introduces the gravitational field strength operator and an adaptive threshold selection method based on standard deviation for images with rich edge information. By implementing these improvements, the Canny algorithm becomes simpler, more efficient in saving useful edge information, and more robust to noise [21].

The traditional Canny algorithm has the following steps:

1. To remove noise, a Gaussian filter is applied to smooth the image. The filter is selected based on the appropriate Gaussian function to smooth the image according to the rows and columns, which is achieved by convoluting the image matrix. The Canny algorithm typically uses a two-dimensional Gaussian function (Equation (1)) to achieve this, as the convolution operation satisfies the commutative and associative laws.

$$G(x, y) = \frac{e^{-\frac{x^2+y^2}{2\sigma^2}}}{2\pi\sigma^2} \quad (1)$$

2. The parameter  $\sigma$  in the Gaussian filter determines the extent to which the image is smoothed, affecting its expansion.
3. The intensity gradient of an image can be determined by analyzing the change in pixels. When there is a significant difference between adjacent pixels, it can suggest the presence of an edge in that particular area. This edge is visually represented as a transition from black to white. The first-order partial derivative in the  $x$  and  $y$  directions can be approximated using the following formula (Equations (2) and (3)):

$$E_x[i, j] = \frac{I[i+1, j] - I[i, j] + I[i+1, j+1] - I[i, j+1]}{2} \quad (2)$$

$$E_y[i, j] = \frac{I[i, j+1] - I[i, j] + I[i+1, j+1] - I[i+1, j]}{2} \quad (3)$$

The template of the image gradient calculation operator (Equations (4) and (5)) is:

$$G_x = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \quad (4)$$

$$G_y = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \quad (5)$$

The size and direction of the gradient can be calculated. The image gradient size (Equation (6)) is:

$$\|M(i, j)\| = \sqrt{E_x[i, j]^2 + E_y[i, j]^2} \quad (6)$$

The azimuth of the image gradient (Equation (7)) is:

$$\theta(i, j) = \arctan \frac{E_y[i, j]}{E_x[i, j]} \quad (7)$$

4. Non-maximum suppression is a technique used to eliminate false edge detections. It involves suppressing pixels whose gradient is not large enough and retaining only the maximum gradient, resulting in thin edges. This classical thin-edge algorithm is applied after obtaining the gradient magnitude image  $M[i, j]$  to accurately locate the edge. The Canny algorithm utilizes  $3 \times 3$  adjacent regions, consisting of eight directions to interpolate the gradient amplitude along the gradient direction. If the magnitude  $M[i, j]$  is greater than the two interpolation results in the gradient direction, it is identified as a candidate edge point; otherwise, it is marked as a non-edge point. This process generates a candidate edge image.

5. To determine the possible boundary, the double threshold method is applied. Despite non-maximum suppression, there may still be noise points in the image. Therefore, the Canny algorithm employs a technique in which a threshold upper bound and a threshold lower bound are set. The process involves marking pixels with a gradient amplitude higher than the high threshold as edge points and those with a lower gradient amplitude as non-edge points. The remaining pixels are marked as candidate edge points. Candidate edge points connected to edge points are then marked as edge points, reducing the impact of noise on the final edge image.
6. The boundary is tracked using the lag technique. The candidate edge points are re-evaluated and the 8-connected domain pixels of a weak edge point are examined. If there are any strong edge points present, the weak edge point is considered to be retained as part of the edge. However, if there are no strong edge points, the weak edge is suppressed.

The traditional Canny edge detection algorithm has been widely used in practical engineering, but there are still two aspects that require improvement. Firstly, the algorithm uses the first-order finite difference  $2 \times 2$  adjacent region to calculate the image gradient. The lack of deviation in the  $45^\circ$  and  $135^\circ$  directions may cause the loss of genuine edge information. Additionally, setting a fixed value for the double threshold in the traditional Canny algorithm can result in the loss of local feature edge information, especially in images with abundant edge information such as rice pests and diseases. Therefore, the traditional Canny algorithm may not be the most adaptable option for such images. In this paper, the authors propose the use of gravitational field strength as a replacement for image gradient. Additionally, they suggest an adaptive threshold selection method based on standard deviation. These techniques aim to improve the clarity and accuracy of image processing.

The law of universal gravitation is utilized in gravitational edge detection algorithms for image processing. In this method, each pixel is treated as an object with a mass equivalent to its gray value. However, the performance of the gravitational edge detection algorithm varies significantly in bright and dark regions. When pixels are in darker areas with lower quality, their gradients generate less total gravity compared to brighter areas. This causes the gravity method to reduce the correlation of gradient changes in dark regions, leading to a loss of edge points. Therefore, the gravitational field strength is introduced to overcome the difference between the bright and dark regions. The total gravitational field strength generated at a point in the image is a combination of the gravitational field strength generated by the surrounding pixels. In this paper, the obtained gravitational field intensity is understood as an image gradient. If the intensity on a pixel exceeds the threshold, it is classified as an edge point. The formula to calculate the gravitational field strength assigned to the point (Equation (8)) is as follows:

$$E_{total}^{\rightarrow} = \sum_{i=1}^n \frac{Gm_i}{\left\| \frac{\rightarrow}{r_i} \right\|^2} \frac{\rightarrow}{r_i} \quad (8)$$

The gradient component in the X direction (Equation (9)) is:

$$\frac{\rightarrow}{E_x} = \sqrt{2}G(m[i+1, j+1] - m[i, j+1] + m[i+1, j] - m[i, j]) \frac{\rightarrow}{i} \quad (9)$$

The gradient component in the Y direction (Equation (10)) is:

$$\frac{\rightarrow}{E_y} = \sqrt{2}G(m[i+1, j+1] - m[i+1, j] + m[i, j+1] - m[i, j]) \frac{\rightarrow}{j} \quad (10)$$

Therefore, the gradient size (Equation (11)) is:

$$E = \sqrt{E_x^2 + E_y^2} \quad (11)$$

The gradient azimuth (Equation (12)) is:

$$\theta = \arctan \frac{E_y}{E_x} \quad (12)$$

where  $\vec{i}$  and  $\vec{j}$  are the unit vectors in the horizontal and vertical directions, respectively. In order to retain more edge information, the adjacent region  $2 \times 2$  is extended to  $3 \times 3$ .

The gradient component in the X direction (Equation (13)) is:

$$\vec{E}_x [i, j] = G \frac{(m_4 - m_8) + \sqrt{2}(m_5 - m_1 + m_3 - m_7)}{4} \vec{i} \quad (13)$$

The gradient component in the Y direction (Equation (14)) is:

$$\vec{E}_y [i, j] = G \frac{(m_2 - m_6) + \sqrt{2}(m_1 - m_5 + m_3 - m_7)}{4} \vec{j} \quad (14)$$

Therefore, the gradient size (Equation (15)) is:

$$\left\| \vec{E} (i, j) \right\| = \sqrt{\vec{E}_x [i, j]^2 + \vec{E}_y [i, j]^2} \quad (15)$$

The gradient azimuth (Equation (16)) is:

$$\theta = \arctan \frac{\left\| \vec{E}_y [i, j] \right\|}{\left\| \vec{E}_x [i, j] \right\|} \quad (16)$$

The value of the pixel located in the upper left corner of the central pixel  $I[i, j]$  is represented by  $m_1$ . Moving clockwise around the central pixel, the values of the pixels in the adjacent positions are represented by  $m_2, m_3, \dots, m_8$ .

The image of rice diseases and insect pests contains rich edge information and a scattered gradient amplitude distribution. However, due to the inconsistent contrast of each part of the image and the relatively large standard deviation of the image gradient, selecting a double threshold for the entire image is not sufficient for completing edge detection. This is because some edge regions with a small gradient magnitude will have a selected threshold that is too high, resulting in the loss of detailed edges. This study proposes a method for selecting double thresholds for each pixel. The first step involves calculating the average value of the gradient size of the entire image  $E_{ave}$  (Equation (17)).

$$E_{ave} = \sum_{i=1}^m \sum_{j=1}^n \frac{|E[i, j]|}{m \times n} \quad (17)$$

If the gradient size of the pixel  $I[i, j]$  is less than 15–20% of the average gradient amplitude ( $E_{ave}$ ), it will be classified as a non-edge point. This step is crucial in preventing the algorithm from introducing more noise in areas with fewer edges, such as those with small average gradient amplitude and standard deviation, in rice pest images. To determine the threshold of pixel  $I[i, j]$ , the average and standard deviation of the gradient sizes of an  $N \times N$  matrix are calculated.  $N$  is an odd number and is typically greater than 20. The center of the threshold is the pixel  $I[i, j]$ . The pixel threshold (Equation (18)) can then be obtained by:

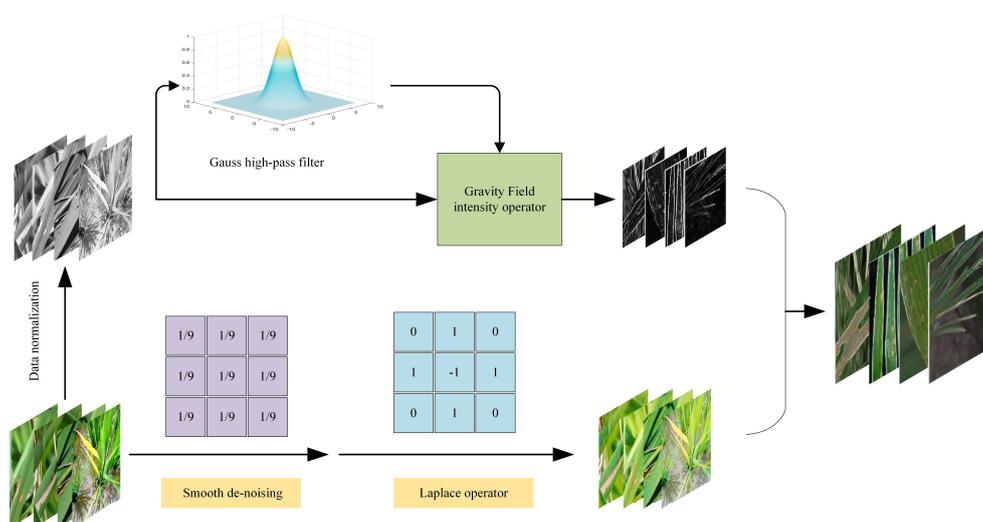
$$T_h = E_{ave} + k\sigma \quad (18)$$

To obtain the threshold for each pixel in the boundary region of an image with a matrix less than  $N \times N$ , the insufficient part is first set to empty. The mean and standard

deviation of the matrix are then calculated. This process results in each pixel having its corresponding double threshold, which can be used to detect and connect the edges of the entire image [32].

The Candy image enhancement algorithm follows a specific process, which is illustrated in Figure 3:

1. The first step in the list number image processing pipeline involves preprocessing the original image. This includes normalization and conversion to grayscale if the original image is in color. The resulting normalized and grayscale image is then used as input for the subsequent steps.
2. The process of extracting the mask and detail enhancement images involves two steps. Firstly, the input image is processed using the improved canny algorithm to extract the edge details. The detected image edge information is then utilized as the mask. In the second step, the input image undergoes a process of smoothing and denoising. To deal with the noise, the Laplace operator is used, which can have a strong response to the noise and cause negative effects. Therefore, denoising processing should be carried out prior to using the Laplacian operator. This allows the denoised image to better highlight small details in the image.
3. A preliminary detail-enhanced image can be obtained by adding the result of the Laplacian operator to the input image. However, since the Laplacian operator is isotropic, it detects isolated points effectively, but may cause loss of the edges of the square. Therefore, further operations are required to preserve the information.
4. To extract the image well, the processed mask image is multiplied with the preliminary enhanced image. The image enhancement algorithm results from adding the edge and detail information to the input image [33].



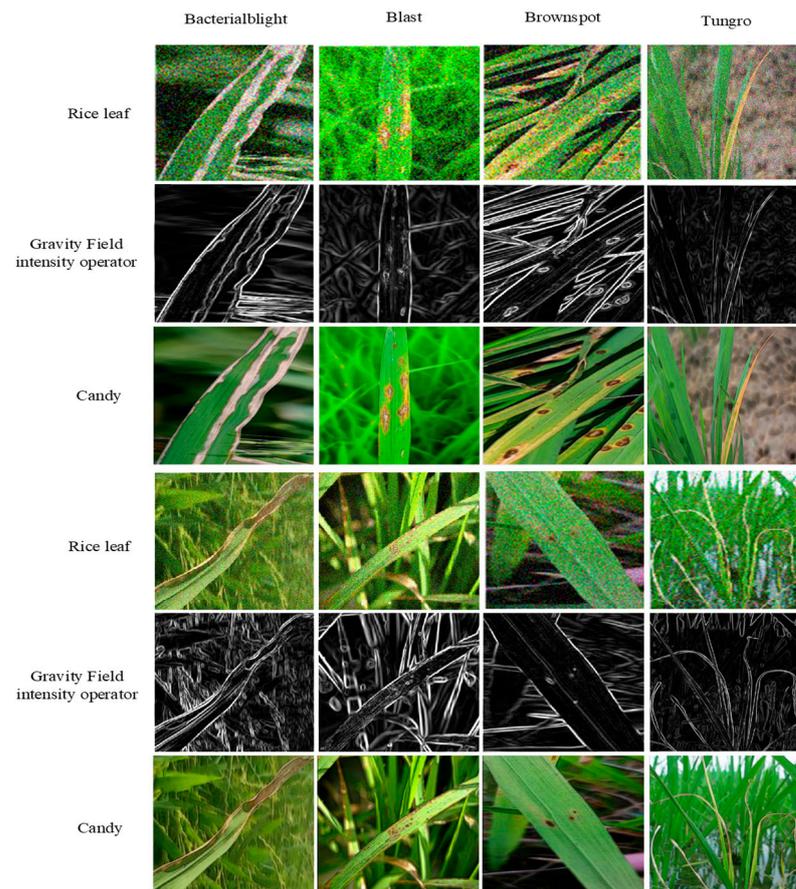
**Figure 3.** Main process of rice leaf image processing with Candy algorithm.

The rice leaf image was processed using the Candy image enhancement algorithm, as depicted in Figure 4.

### 2.3. Identification of Rice Diseases Based on ICAI-V4 Model

In the field of rice pest and disease identification, leaf images exhibit significant intra-class diversity and inter-class similarity. To overcome this challenge, it is crucial to enhance the feature extraction capability of convolutional neural networks. This can be achieved by obtaining more detailed features and reducing the false positive rates. Deep convolutional networks have been instrumental in significantly improving image recognition performance in recent years and are therefore considered the core of this advancement. The Inception-V4 architecture has demonstrated impressive performance while requiring relatively low computational resources. This paper selects the Inception-V4

model as the primary component of the neural network framework. To ensure that the entire model can be stored in memory, the model was divided into six subnetworks and trained accordingly. The Inception architecture offers a high level of tunability, allowing for numerous adjustments in the number of filters per layer without compromising the quality of the fully trained network. Each subnetwork is inspired by traditional image filtering techniques and boasts two key features that contribute to its appeal and widespread use: spatial agnosticity and channel specificity. The convolution kernel's ability to adapt to different visual modes in different spatial locations is limited, which raises questions about its flexibility for different channels.



**Figure 4.** Picture utilizing the gravitational edge detection algorithm and Candy algorithm for processing rice leaf images.

To address the limitations mentioned above, this paper proposes a modification to the Inception blocks by incorporating involution to the subnetwork structure. This operation possesses symmetrical inverse inherent characteristics, which enhance flexibility across different channels and improve the network's learning effect compared to convolution [34]. To address the issue of neuron death caused by the ReLU function, the Leaky ReLU activation function is utilized. Furthermore, coordination concerns are integrated into the backbone network to enhance the overall performance. The proposed method has the ability to capture not only cross-channel information, but also direction-aware and location-sensitive information. This feature helps the model to identify and locate objects of interest more accurately and enhance features by emphasizing information representation.

The ICAI-V4 network structure boasts excellent feature extraction capabilities and robustness. This results in improved accuracy in recognizing and classifying similar disease features, while also reducing the model's running time. The structure comprises a backbone module, a coordination attention module, an INCV structure, an average pooling layer, a discard layer (with a keep probability of 0.8), and a classification block. The network

structure of ICAI-V4 is illustrated in Figure 5. The primary workflow can be summarized as follows:

1. The initial layer of ICAI-V4 is the Stem layer, which primarily serves to rapidly decrease the resolution of the feature map. This reduction in resolution allows for subsequent inceptions to decrease the amount of computation required. Additionally, the activation function of the standard convolution layer has been modified from ReLU to Leaky ReLU, which enhances the network's robustness.
2. The second module in ICAI-V4 is the coordination attention module. This module is capable of capturing cross-channel information, as well as direction-aware and position-sensitive information. This enables the model to more accurately locate and identify objects of interest and enhance features by emphasizing information representation.
3. The third component of ICAI-V4 is the INCV structure, which consists of  $1 \times 1$  convolution,  $3 \times 3$  convolution, pooling layer, and asymmetric convolution decomposition. The involution layer is incorporated into various structures to enhance the flexibility of different channels and improve the overall network learning effect. The network comprises three Inception-iv structures and two Reduction-iv structures used multiple times. Refer to Figure 5 for a detailed illustration of the specific structure.
4. ICAI-V4's fourth component consists of an adaptive pooling layer, data dimensionality reduction, linear layer, and a discard layer. The classification results of input rice pest images are determined by a softmax activation function.

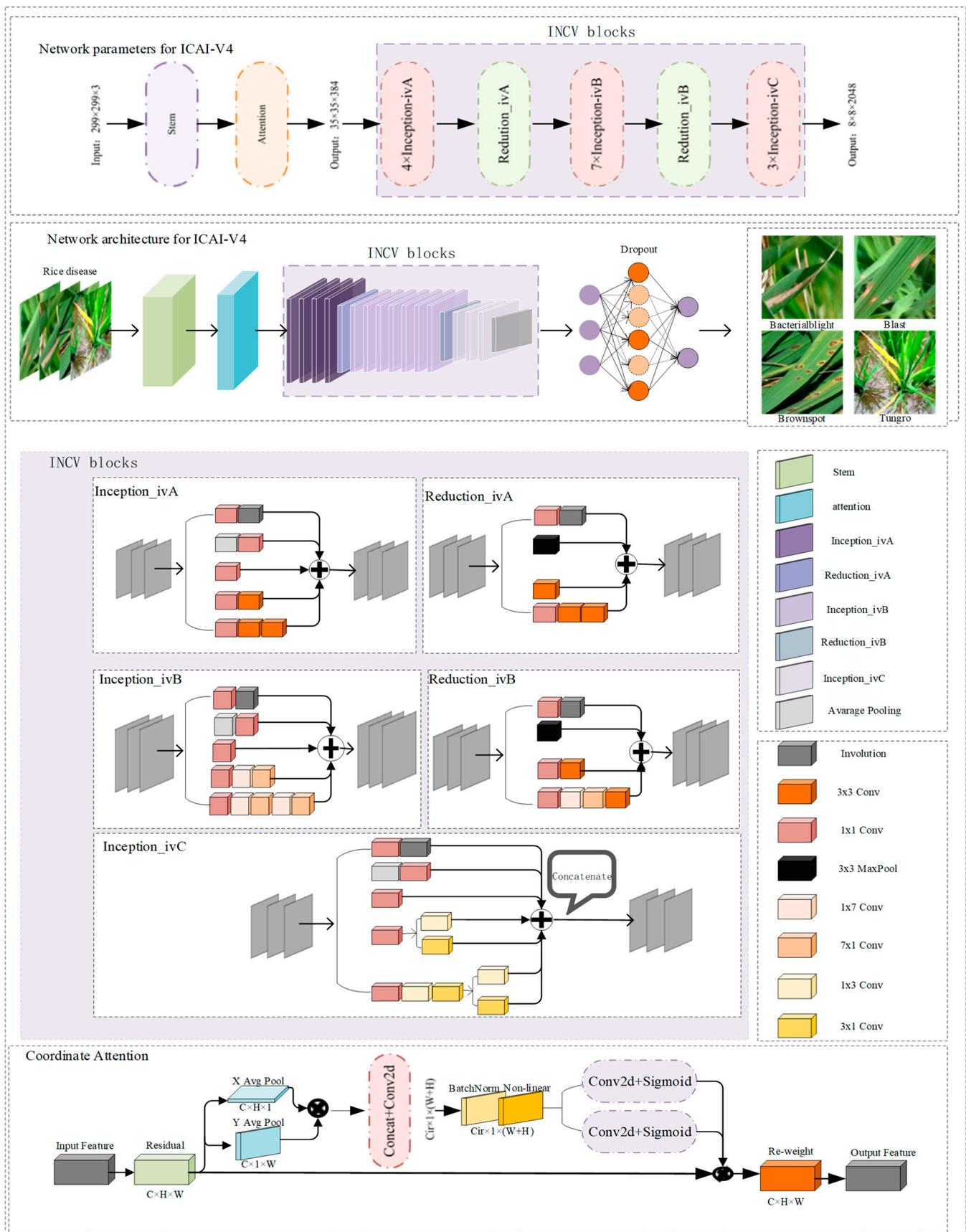
### 2.3.1. Coordinate Attention

In order to enhance the effectiveness of the model's performance, this paper proposes the addition of a coordinated attention mechanism to the backbone network of Inception-V4. The attention mechanism is divided into two modules: channel attention and spatial attention. While channel attention has proven to be effective in improving model performance, it often disregards location information. In their recent study, Hou et al. (2021) introduced a novel approach to mobile network attention mechanisms, termed 'coordinate attention'. This method involves incorporating location data into channel attention, resulting in improved network performance [35].

In coordinating attention using two 1D global pooling operations, input features along the vertical and horizontal directions are aggregated into two separate directional software feature maps. These feature maps, which contain embedded direction-specific information, are then encoded into two attention graphs. Each attention graph captures the long-distance correlation of the input feature map along a spatial direction. The proposed attention method, named coordinate attention, can save location information in the generated attention map [36]. This allows for the application of two attention maps to the input feature map through multiplication, resulting in an emphasized representation of interest. The operation of this method distinguishes spatial directions and generates a coordinate-aware attention map. Coordinated attention not only captures cross-channel information, but also direction-aware and position-sensitive information. This additional information aids the model in accurately identifying and locating objects of interest [37].

A coordinate attention block serves as a computing unit that amplifies feature expression in mobile networks. It takes an intermediate characteristic tensor  $X = [x_1, x_2, \dots, x_c] \in R^{C \times H \times W}$  as input and outputs  $Y = [y_1, y_2, \dots, y_c]$  with the same size and enhanced representation, achieved through transformation.

The global pooling method is often utilized for encoding channel attention and spatial information. However, it can be challenging to retain location information since it compresses global spatial information into channel descriptors. To address this issue, the coordinate attention method decomposes global pooling into one-to-one feature encoding operations (Equation (19)). This allows the attention module to capture remote spatial interactions with accurate location information [38].



**Figure 5.** Block diagram of ICAI-V4 model. It includes INCV blocks and coordinate attention network structure.

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_c(i, j) \quad (19)$$

In order to encode each channel in input  $X$ , a pooled kernel of size  $(H, 1)$  or  $(1, W)$  is utilized along the horizontal and vertical coordinates. This results in the output of channel  $c$  with height  $h$  being expressed as (Equation (20)):

$$Z_c^h(h) = \frac{1}{W} \sum_{0 \leq i \leq W} x_c(h, i) \quad (20)$$

Similarly, the output of channel  $c$  with width  $w$  can be written as (Equation (21)):

$$Z_c^w(w) = \frac{1}{H} \sum_{0 \leq j \leq H} x_c(j, w) \quad (21)$$

The above two transformations generate feature maps along two spatial directions, enabling the attention module to capture long-term dependencies along one direction and preserve accurate location information along another. This aids the network in more precise target localization.

After the information embedding transformation, the resulting output is concatenated with another transformation, and then passed through the convolution transformation function (Equations (22)–(24)).

$$f = \delta \left( F_1 \left( \left[ z^h, z^w \right] \right) \right) \quad (22)$$

$$g^h = \sigma \left( F_h \left( f^h \right) \right) \quad (23)$$

$$g^w = \sigma \left( F_w \left( f^w \right) \right) \quad (24)$$

Finally, the output  $Y$  of the coordinate attention block (Equation (25)) can be written as:

$$g_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \quad (25)$$

The specific structure of coordinate attention, which is added to the backbone network in this paper, aims to improve the model's performance and feature grasping ability. This can be seen in Figure 5.

### 2.3.2. INCV Blocks

The high intra-class diversity and inter-class similarity of rice leaf diseases require the network model to effectively extract feature information, thus improving the accuracy and efficiency of the model. Despite the rapid development of neural network architecture, convolution remains the backbone of deep neural network construction. The convolution kernel, inspired by classical image filtering methods, has two significant characteristics that contribute to its popularity and effectiveness: spatial agnosticity and channel specificity. While the concepts of spatial agnosticism and spatial compactness may seem beneficial in terms of increasing efficiency and explaining translation equivalence, they may limit the adaptability of convolution kernels to varying visual modes in different spatial locations. This raises concerns about the ability of convolution kernels to flexibly adjust to different channels.

In their recent paper, Li et al. (2021) introduced involution as a potential solution to address the limitations of convolution [34]. Involution offers symmetrical inverse characteristics that are space specific and channel-unknown. This means that the involution cores are unique to each spatial location, but shared across channels. If the involution kernel is parameterized into a fixed-size matrix, such as a convolution kernel, and updated

using a back-propagation algorithm, it may not be able to transmit between input images with varying resolutions due to its space-specific properties. To address variable feature resolution, the involution kernel for a specific spatial position can only be generated when the corresponding position's incoming feature vector is present. This approach is intuitive and effective. Additionally, Involution reduces core redundancy by sharing involution cores along the channel dimension. The computational complexity of the involution operation increases in a linear fashion as the number of feature channels increase. This means that the dynamic parametric involution kernel is capable of covering a wide range of spatial dimensions. Reverse design allows for two advantages with involution: (i) it can summarize context in a wider spatial arrangement, overcoming the challenge of effectively modeling long-distance interaction; (ii) involution can assign adaptive weights to different locations to prioritize visual elements with the most information in the spatial domain [39].

In contrast to convolution, involution involves sharing the kernel in the channel dimension while utilizing a space-specific kernel for more adaptable modeling in the spatial dimension. The size of the involution kernel (Equation (26)) is:

$$H_I \in R^{H \times W \times K \times K \times G} \quad (26)$$

where  $H$  and  $W$  denote their height and width,  $K$  is kernel size, and  $G < C$  denotes that all channels share  $G$  kernels. In the context of involvements, the fixed weight matrix is not utilized as a learnable parameter, unlike in convolution. Instead, the involvements kernel is generated based on the input feature mapping to ensure the automatic alignment of kernel size and input feature size in the spatial dimension [40]. Unlike the convolution kernel, the shape of the involution kernel  $H_i$  depends on the shape of the input feature map. A natural idea is to generate an involution kernel based on the original input tensor, so that the output kernel and the input are easily aligned. We express the kernel generating function as  $\varnothing$ , and abstract the function mapping at each position  $(i, j)$  (Equation (27)) as [34]:

$$H_{i,j} = \varnothing(X_{\varphi_{i,j}}) \quad (27)$$

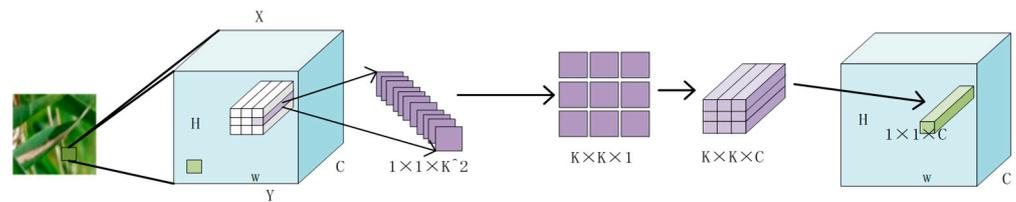
where  $\varphi_{i,j}$  is an index set of the neighborhood of coordinates  $(i, j)$ , and  $X_{\varphi_{i,j}}$  represents a patch containing  $X_{i,j}$  on the feature map.

The output feature map of the involution is obtained by performing a multiplicative addition operation on the input with such an involution kernel (Equation (28)). The definition is as follows:

$$Y_{i,j,k} = \sum_{(u,v) \in \Delta k} H_{i,j,u+\frac{K}{2},v+\frac{K}{2},\frac{kG}{C}} X_{i+u,j+v,k} \quad (28)$$

Figure 6 displays the complete involution diagram, depicting the process of generating the involution convolution kernel and subsequently producing a new feature diagram. In this method, the representation of a pixel of size  $1 \times 1 \times C$  is reduced to  $C/r$  through linear change, where  $r$  represents the reduction ratio. This is then changed to  $K \times K \times G$ , which is involution kernels. When  $G$  is 1, the  $K \times K \times 1$  kernel is multiplied with the pixels in the  $K \times K$  field of the current pixel in the original feature map, and repeated on  $C$  channels to obtain the three-dimensional matrix of  $K \times K \times C$ . The width and height of the three-dimensional matrix are summed in two dimensions, while the channel dimension is retained. This generates a  $1 \times 1 \times C$  vector in the sex feature diagram, where the value of  $C$  channels in a pixel position is generated at once.

The involution kernel  $H_{i,j} \in R_v$  is created using a function  $\varphi$  that takes a single pixel at  $(i, j)$  as a condition and rearranges the channel to space. The multiplication and addition operations of the involution are divided into two steps. The product operation multiplies the tensors of  $C$  channels with the involution kernel  $H$ , respectively, while the addition operation adds the elements within the involution kernel to the involution kernel.



**Figure 6.** Schematic diagram of involution.

Involution has the following advantages over convolution:

1. Sharing kernels on the channel improves performance by allowing us to use large spatial spans while maintaining design efficiency through the channel dimension. This is true even if weights are not shared in different spatial locations, as it does not significantly increase the number of parameters and calculations.
2. While the kernel parameters of each pixel in the space are not directly shared, involution does share meta-weights at a higher level, specifically the parameters of the kernel generating function. This allows us to still share and migrate knowledge across different spatial locations. However, freeing the limitation of the convolution sharing kernel in space and allowing each pixel to learn its corresponding kernel parameters does not necessarily lead to better results, despite addressing the issue of large parameter increase.

The Inception-V4 model is trained in a partitioned manner, consisting of six subnetworks. The Inception architecture is highly flexible, with each subnetwork incorporating multiple branches inspired by classical image filtering methods. These branches exhibit spatial agnosticity and channel specificity, allowing for effective feature extraction. The convolution kernel's ability to adapt to different visual modes in different spatial locations is limited, which questions its flexibility for different channels. Involution, on the other hand, is capable of modeling long-distance interactions and is sensitive to channel information. This paper proposes enhancements to the inception and reduction structures in the backbone network infrastructure of Inception-V4. The subnetworks now include a  $1 \times 1$  convolution and an involution layer, which enhances the network's ability to extract features from channel perspectives. As a result, it can better classify similar images of rice pests and diseases. To better fit the Inception architecture, a  $1 \times 1$  convolution is added to adjust the input and output parameters. After adding this convolution and Involution layer to the structure, the model's accuracy is shown, as in Table 4. The results demonstrate that the proposed INCV structure has higher accuracy and is more suitable for rice pest classification [40].

**Table 4.** Accuracy of rice disease classification after each change of INCV block structure.

Network Structure	Accuracy
Baseline	81.27
Inception_ivA	82.51
Reduction_ivA	82.93
Inception_ivB	85.14
Reduction_ivB	85.94
Inception_ivC	87.74

### 2.3.3. Leaky ReLU

The ReLU function is a commonly used activation function in deep learning. Its expression is (Equation (29)):

$$\text{ReLU} = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (29)$$

The ReLU function has the following characteristics:

1. When the input is positive, there is no gradient saturation problem.

2. The calculation speed is much faster. There is only a linear relationship in the ReLU function, so its calculation speed is faster than the Sigmoid function and the tanh function [41].
3. The Dead ReLU problem arises when the input to the ReLU activation function is negative, rendering it completely invalid. This is not a concern during forward propagation as only certain areas are sensitive to negative input. However, during back propagation, if a negative number is encountered, the gradient will be zero, resulting in non-convergence of calculation results and neuronal death. This prevents weight updates, leading to the problem of non-updating neurons.
4. The output of the ReLU function is zero or positive, which means that the ReLU function is not a zero-centered function.

To address the Dead ReLU problem associated with the ReLU activation function, this paper proposes the use of Leaky ReLU. The proposed activation function is depicted in the accompanying figure. The expression for Leaky ReLU (Equation (30)) is:

$$f(x) = \begin{cases} x, & x \geq 0 \\ ax, & x < 0 \end{cases} \quad (30)$$

The parameter 'a' is randomly generated during model training and eventually converges to a constant value. This process enables one-sided suppression while retaining some negative gradient information to prevent complete loss.

The Leaky ReLU function has the following advantages:

1. Some neurons will not die. The Dead ReLU problem of the ReLU function is solved;
2. The linear and unsaturated form of Leaky ReLU can converge quickly in SGD;
3. The Leaky ReLU function has a faster calculation speed compared to the sigmoid and tanh functions. This is because it has a linear relationship and does not require exponential calculations. Nonlinearity typically requires more computation and thus results in slower execution speeds.

#### 2.4. Evaluating Indicator

This paper examines the classification of rice diseases and evaluates model performance using indicators such as accuracy, precision, recall rate, F1 score, and ROC/AUC.

Assuming that all images categorized under rice blast disease are positive samples, the remaining disease images are considered negative samples and do not belong to the rice blast disease category. The study focused on the category of rice blast and used it as an example. TP, which stands for true positive, refers to rice images that were accurately classified as having rice blast disease and were placed in the correct category. FP (false positive) refers to incorrectly predicting a rice image with other diseases as rice blast disease. TN (true negative) refers to predicting an image of rice with other diseases as belonging to their respective categories. FN (false negative) refers to mistakenly predicting an image of rice with rice blast disease as belonging to another disease category. Table 5 displays the structure of the confusion matrix using a binary task as an example.

**Table 5.** Classification model prediction results table.

Confusion Matrix		Prediction	
		True	False
Practical	True	TP	FN
	False	FP	TN

Accuracy (Equation (31)) is defined as the ratio of correctly classified samples to the total number of samples, expressed as a percentage. It is a commonly used metric to evaluate the performance of a classification model:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (31)$$

The accuracy rate (Equation (32)) is a metric that measures the proportion of correctly predicted positive samples out of all of the samples that are predicted to be positive by the model:

$$Precision = \frac{TP}{TP + FP} \quad (32)$$

Recall rate (Equation (33)), also known as sensitivity or true positive rate, is a measure of the proportion of actual positive samples that are correctly identified as positive by a prediction model. In other words, it is the ratio of true positives to the sum of true positives and false negatives [30]:

$$Recall = \frac{TP}{TP + FN} \quad (33)$$

The F1 score (Equation (34)), also known as the F-score or F-measure, is a measure of a test's accuracy. It is calculated as a weighted average of the precision and recall, which are measures of the test's ability to correctly identify positive results and correctly exclude negative results, respectively. The formula for calculating the F1 score is:

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (34)$$

### 3. Results

#### 3.1. Experimental Environment and Data Preparation

Hardware: Intel (R) Core (TM) i7-10875H CPU @ 2.30GHz; graphics card: NVIDIA GeForce RTX 2060; system memory: 16.0 GB. Software environment: Anaconda Navigator 2.1.2; PyTorch 1.10; Python 3.8.12; MATLAB R2021b. Operating system: Windows 10.

The training of machine learning models for detecting rice diseases requires a large number of disease samples. However, obtaining sufficient disease images is a challenging task. To overcome this challenge, it is necessary to expand the dataset to increase the number of samples before training the model. This approach can help prevent the overfitting of the network and improve the model's performance by generating equivalent values for limited data. To increase the quantity of images, geometric transformations such as horizontal flipping, vertical flipping, and random cropping can be applied. The direction of the image does not affect the classification of rice diseases. Therefore, simulating the shooting of rice images from different angles through geometric transformations can help restore the real shooting situation.

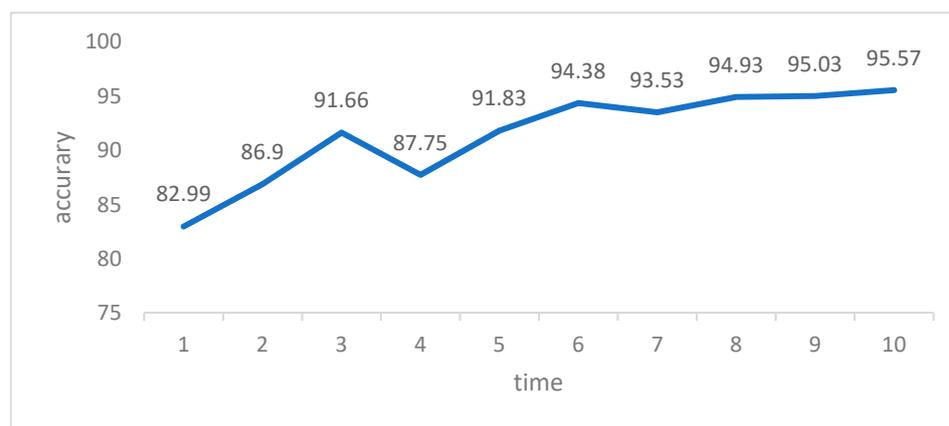
#### 3.2. Experimental Design and Results

The experimental dataset in this paper includes four types of rice diseases: bacterial blight, blast, brown spot, and tungro. To ensure consistency, the input image size was standardized to  $299 \times 299 \times 3$ . Following image enhancement and data expansion, a total of 10,241 rice disease images were obtained. To ensure the reliability and accuracy of the model, K-fold cross-validation was employed to train and test the model in all subsequent verification experiments. Cross-validation is a technique used to assess the performance of a model. It involves dividing the original data into groups, where most of the samples are used as the training set to train the model, and the remaining samples are used as the test set to evaluate the model's performance. The test error of the small samples is determined and their sum of squares is recorded. This process is repeated until all samples have been predicted and tested once. This article employs a K value of 10 for 10-fold cross validation. The rice disease dataset was split into 10 groups, with 9 groups being used for training and 1 group for testing. The accuracy of the model was measured by averaging the results from the 10 tests. To enhance the diversity of the dataset partitioning and improve the model's reliability and accuracy, we employed 10-fold cross-validation. The dataset was

randomly divided 10 times, and each model was trained and tested accordingly. The overall classification accuracy of the network model is determined by retaining the best results of each 10-fold cross-validation and using the average of these 10 results. During the process of network training, it is crucial to adjust the hyperparameters appropriately. One commonly used method for hyperparameter adjustment is grid search, which helps in selecting the optimal set of hyperparameters. Grid search involves optimizing the model by traversing different hyperparameter combinations. When training a network model, it is important to pay attention to and adjust the following hyperparameters: learning rate, batch size, epoch, and momentum parameter. The paper describes the experimental setup for a machine learning model, including the values used for the learning rate (0.1, 0.01, 0.001), batch size (16, 32, 64), epoch (10, 20, 30), and momentum parameter (0.2, 0.4, 0.9). To ensure reliable results in the grid search process, a three-fold cross validation approach is employed. The study determined that the optimal hyper-parameter combination is achieved with a learning rate of 0.001, a batch size of 32, an epoch of 20, and a momentum parameter of 0.9. Further grid search was conducted on the Adam, RMSprop, and SGDM optimizers, and the RMSprop optimizer was found to perform the best under the aforementioned hyperparameter combination. The final hyperparameter combination chosen is presented in Table 6. To ensure accuracy and reliability, all experiments were conducted in the same hardware and software environment, and all models were trained using identical hyperparameter settings. The training results of the 10-fold cross-validation are depicted in Figure 7.

**Table 6.** Hyperparameter settings.

Hyperparameters	Value
Learning rate	0.001
Epoch	20
Momentum	0.9
Batch size	32
Optimizer	RMSprop



**Figure 7.** Accuracy of 10-fold cross-validation training results.

### 3.3. Verifying the Effectiveness of Data Preprocessing

To ensure the reliability of the rice disease datasets, similar data extensions were performed on unenhanced datasets processed by the Candy image enhancement algorithm after image preprocessing, including image enhancement and data extension. This involved utilizing methods such as horizontal and vertical flips, as well as random clipping. After applying the Candy operator filtering, four distinct datasets were obtained: the original dataset without any enhancements, the initial dataset with improved Candy operator filtering, the extended dataset without any enhancements, and the extended dataset with

improved Candy operator filtering. After inputting the datasets into the proposed ICAI-V4 model for training and testing, the results in Table 7 show that data expansion can improve accuracy regardless of whether image enhancement is performed. This is due to the smaller sample size of the original dataset, and data expansion provides more samples for the model, increasing the diversity of samples and improving the model's generalization ability.

**Table 7.** Classification accuracy of different networks before and after image enhancement and dataset enhancement.

Image Enhancement Methods	Original Dataset (%)	Extended Dataset (%)
No image enhancement	81.27	89.26
Image enhancement	86.48	95.57

### 3.4. Ablation Experiment

To assess the effectiveness of the method, an ablation experiment was conducted. The ablation experiments use the same experimental setting and a 10-fold cross-validation method using augmented datasets after adding noise to some samples. This paper utilized InceptionV4, AlexNet, and ResNeXt as general frameworks for the network, and conducted ablation experiments using one or more of the methods proposed in the paper. This study aims to validate the efficacy of Coordinate Attention, INCV, and Leaky ReLU through experimentation. The results of the experiment are presented in Table 8.

**Table 8.** Ablation experimental results.

Model	InceptionV4	AlexNet	ResNeXt
Baseline	81.27	89.68	89.07
CA	89.06	91.25	91.21
INCV	87.74	92.14	90.13
Leaky ReLU	84.14	90.23	89.91
CA+INCV	94.25	94.35	91.25
CA+Leaky ReLU	91.24	93.13	90.63
INCV+Leaky ReLU	90.25	92.53	91.25
CA+INCV+Leaky ReLU	95.57	93.23	92.14

Table 8 demonstrates that Coordinate Attention, Involution, and Leaky ReLU all contribute to enhancing the classification accuracy of rice. Notably, ICAI-V4 exhibits the highest average classification accuracy of 95.57%. In order to enhance the efficiency of the model, the coordinated attention mechanism is utilized to transform the internal location information into channel attention. This enhances the network's ability to extract detailed features of rice diseases. To increase the flexibility of the convolution kernel and optimize performance across different channels, we propose an INCV architecture that employs dynamic weight assignment to prioritize the most informative visual elements in the spatial domain. Our results demonstrate that the addition of both INCV and Coordinate Attention to the basic neural network leads to higher accuracy, as shown in the table below. When utilizing both Coordinate Attention and Leaky ReLU in the AlexNet network, the accuracy reaches its peak at 94.13%. The integration of Coordinate Attention, INCV, and Leaky ReLU led to improved accuracy in all three models. Through analyzing the ablation experimental results, it has been found that ICAI-V4 can achieve higher classification accuracy and exhibit relatively strong classification performance and practical value.

### 3.5. Comparison with Other Networks

In the realm of digital photography, the images captured are often subject to noise interference due to the internal characteristics of the equipment and the imaging environment. This interference can manifest as composite noise, which is a combination of Gaussian noise and impulse noise. All models listed in Table 9 underwent retesting to simulate

compound noise. The standard deviation of Gaussian noise was set to 15–60, while the ratio of impulse noise was set to 0.02–0.08. To test the model’s anti-interference ability, synthetic noise can be added to the sample and then removed to restore the image to a more realistic quality. In order to validate the proposed rice disease classification model, multiple models, including the ICAI-V4 model as well as classical and recent models, were tested such as MobileNetV3 (Tarek et al., 2022) [42], EfficientNet (Chen et al., 2021) [43], DenseNet121 (Huang et al., 2017) [44], and EfficientNetV2 (Sunil et al., 2022) [45]. This approach allowed us to establish the reliability and accuracy of our model in comparison to others. AlexNet is an early deep convolutional neural network, while ResNet50 proposes a residual block that allows for cross-layer connections, enabling features of different layers to transmit to each other and alleviating the problem of gradient disappearance. ResNeXt combines the advantages of ResNet and Inception by using group convolution, residual structure, and feature layer connections. MobileNetV3 is a lightweight network that combines the benefits of MobileNetV1 and MobileNetV2. On the other hand, efficientNetV1 uses neural architecture search technology and introduces composite coefficients to expand the network from three dimensions of network depth, width, and image resolution. EfficiencyNetV2 introduces an enhanced progressive learning approach built on the foundation of EfficiencyNetV1. This updated model boasts faster training speeds and fewer parameters. In summary, the structure of this model is unique and it offers distinct advantages in image classification tasks. As such, it is essential to compare our model with others in order to demonstrate the validity and reliability of our proposed approach.

**Table 9.** Performance of ICAI-V4 and other network models before and after adding noise to the dataset.

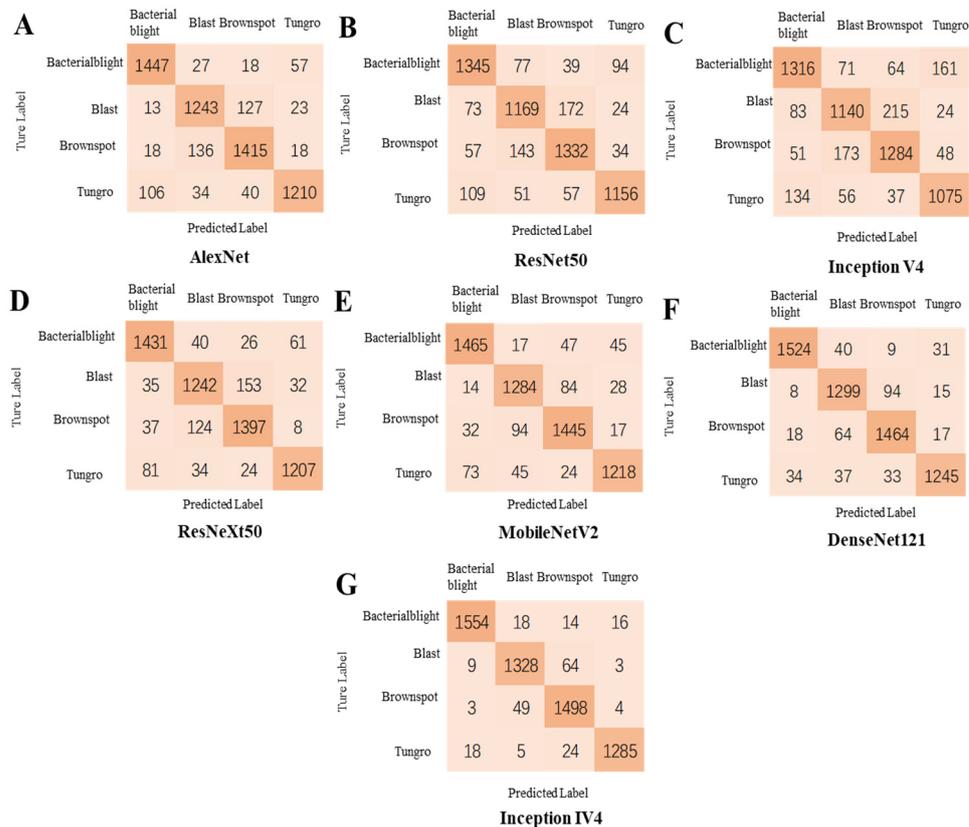
Methods	Accuracy (Test Set without Noise) (%)	Accuracy (Test Set Added Noise) (%)
AlexNet	89.68	84.72
ResNet50	84.48	80.67
Inception V4	81.27	79.31
ResNeXt50	89.07	87.45
MobileNetV2	91.31	89.14
DenseNet121	93.31	93.23
ICAI-V4	95.57	95.25

According to Table 9, Inception V4 has the lowest average classification accuracy for rice diseases compared to all other models. However, our proposed ICAI-V4 model has a good classification effect with an accuracy of 95.57%. Additionally, when random compound noise of Gaussian noise and impulse noise were added to the dataset, the average classification accuracy of ICAI-V4 only decreased by 0.32%. After adding noise, the average accuracy reached its highest in the test set.

### 3.6. Classification Performance Evaluation of ICAI-V4 and Other Networks

In this study, a larger dataset with added noise was utilized to demonstrate the effectiveness of Inception IV4 in accurately classifying rice diseases. The results show the validity of using Inception IV4 for this task. The evaluation of the model is carried out through the use of a confusion matrix, as illustrated in Figure 8, which provides a clear representation of the classification results for each rice disease by various models. In Table 10, the ICAI-V4 model’s classification performance evaluation is presented. The paper evaluated the average accuracy, recall rate, and *F1* scores of four rice diseases. The results showed that both ICAI-V4 and DenseNet121, proposed in this paper, achieved impressive results in classifying these diseases. The average *F1* score was approximately 95%. The classification effectiveness of blast and brown spot in most models is generally low due to the similarities in their disease characteristics. This similarity makes it challenging to train the models and places higher demands on their ability to extract feature information. Compared to other models, the ICAI-V4 proposed in this paper exhibits exceptional classi-

fication performance. Specifically, the F1 average scores for bacterial blight, blast, brown spot, and tungro are 97.54%, 93.38%, 94.98%, and 97.34%, respectively. Furthermore, our model demonstrated a slight improvement in the classification accuracy of blast and brown spot. The addition of a coordinated attention mechanism to the backbone structure of ICAI-V4 allows for the utilization of feature information between channels, resulting in an improved ability to extract image information. This enhancement is particularly beneficial when dealing with the classification of similar diseases.



**Figure 8.** Classification confusion matrix of citrus diseases by different models (the models corresponding to the (A–G) confusion matrix are: AlexNet, ResNet50, Inceptionv4, ResNeXt, MobileNetv3, DenseNet121, and ICAI-V4).

**Table 10.** Classification performance evaluation of ICAI-V4 and other networks.

Methods	Evaluation Indicators	Rice Disease			
		Bacterial Blight	Blast	Brown Spot	Tungro
AlexNet	Precision (%)	93.41%	88.40%	89.16%	87.05%
	Recall (%)	91.35%	86.31%	88.43%	92.50%
	F1-score (%)	92.36%	87.34%	88.79%	89.69%
ResNet50	Precision (%)	86.49%	81.29%	85.05%	84.19%
	Recall (%)	84.91%	81.18%	83.25%	88.37%
	F1-score (%)	85.69%	81.23%	84.14%	86.22%
Inception V4	Precision (%)	81.63%	77.97%	82.51%	82.56%
	Recall (%)	83.08%	79.16%	80.25%	82.18%
	F1-score (%)	82.34%	78.56%	81.36%	82.36%
ResNeXt50	Precision (%)	91.84%	84.95%	89.20%	89.67%
	Recall (%)	90.34%	86.25%	87.31%	92.27%
	F1-score (%)	91.08%	85.59%	88.24%	90.95%
MobileNetV2	Precision (%)	93.07%	91.06%	90.99%	89.55%
	Recall (%)	92.48%	89.16%	90.31%	93.11%
	F1-score (%)	92.77%	90.09%	90.64%	91.29%

Table 10. Cont.

Methods	Evaluation Indicators	Rice Disease			
		Bacterial Blight	Blast	Brown Spot	Tungro
DenseNet121	Precision (%)	95.01%	91.73%	91.50%	95.18%
	Recall (%)	96.21%	90.20%	91.50%	95.18%
	F1-score (%)	95.06%	90.95%	91.50%	95.18%
ICAI-V4	Precision (%)	97.00%	94.58%	96.39%	96.47%
	Recall (%)	98.10%	92.22%	93.62%	98.24%
	F1-score (%)	97.54%	93.38%	94.98%	97.34%

### 3.7. Formatting of Mathematical Components

To assess the effectiveness of ICAI-V4, it was retrained and tested on three different datasets: PlantVillage, Stanford Automobile, and ImageNetDogs. The specifics of each dataset, such as the number of classes, total number of samples, and source, are detailed in Table 11. The proposed model was evaluated using these public datasets, and the results of the tests are presented in Table 12. Our proposed model demonstrated strong performance on three publicly available datasets, suggesting that it possesses a high degree of generalizability to other datasets.

Table 11. Detailed information about the dataset.

Dataset	Category	Total	Available
PlantVillage	38	55,400	<a href="https://www.kaggle.com/datasets/hiyash99/plantvillage">https://www.kaggle.com/datasets/hiyash99/plantvillage</a> (accessed on 24 March 2023)
Stanford cars	196	16,185	<a href="https://www.kaggle.com/datasets/jutrera/stanford-car-dataset-by-classes-folder">https://www.kaggle.com/datasets/jutrera/stanford-car-dataset-by-classes-folder</a> (accessed on 24 March 2023)
ImageNetDogs	120	20,580	<a href="http://vision.stanford.edu/aditya86/ImageNetDogs/">http://vision.stanford.edu/aditya86/ImageNetDogs/</a> (accessed on 24 March 2023)

Table 12. Public dataset test results.

Dataset	Accuracy (%)
PlantVillage	96.25
Stanford cars	94.67
ImageNetDogs	97.24

## 4. Discussion

The effectiveness of ICAI-V4 in identifying rice leaf diseases was assessed and compared to other well-known networks including AlexNet [46], ResNet-50 [47], ResNeXt-50 [48], Inception-V4 [49], MobileNet-V2 [50], and DenseNet-121 [44]. The results presented in Table 10 demonstrate that ICAI-V4 outperformed the traditional mainstream networks in terms of recognition rate for the four rice diseases in the dataset. AlexNet employs dropout regularization to mitigate overfitting in its fully connected layers. To preserve the most salient features, maximum pooling is preferred over average pooling to avoid blurring effects. Furthermore, AlexNet suggests using a stride length smaller than the pooling kernel size to achieve overlap and coverage between the pooling layer outputs, thereby enhancing feature richness and minimizing information loss. The calculation process is straightforward and the convergence speed is rapid. However, the recognition accuracy is limited to 93.41%, 88.4%, 89.16%, and 87.05%. The ResNet-50 deep residual network addresses the issues of low learning efficiency and ineffective accuracy improvement caused by increasing network depth to some extent. While using a deep network layer can be beneficial, it can also result in the selective discarding of some layers, leading to information blocking and decreased recognition accuracy. In this particular case, the

recognition accuracy was measured at 86.49%, 81.29%, 85.05%, and 84.19%. ResNeXt-50, a modification of ResNet-50, widened the network and improved its robustness and identification accuracy (91.84%, 84.95%, 89.20%, 89.67%). The network achieved better identification results for four rice leaf diseases compared to ResNet-50 (+5.35%, +3.66%, +4.15%, +5.48%). However, this improvement came at the cost of a significant increase in parameters and a decline in recognition efficiency. Inception-V4 is designed with a sparse network architecture, allowing it to perform well even with limited memory and computational resources. Its performance is impressive, achieving up to 81.63%, 77.97%, 82.51%, and 82.56%. On the other hand, MobileNet-V2 is a network that utilizes deeply separable convolution and shortcuts. The model utilizes Inverted Residuals and Linear Bottlenecks, resulting in a fast training process and a low number of parameters. While the network layer is not as deep, its recognition accuracy is lower (93.07%, 91.06%, 90.99%, 89.55%). On the other hand, DenseNet-121 employs a dense connection structure that imports the output of each layer into all subsequent layers, effectively reducing the number of parameters in the network. The use of a dense connection structure can effectively address the issues of gradient disappearance and parameter sparsity, resulting in improved generalization ability and model accuracy. In fact, the model achieved an accuracy of (95.01%, 91.73%, 91.50%, 95.18%). The study found that while MobileNet-v2 and Densenet121 had higher recognition accuracy for rice diseases, the ICAI-v4 proposed in the paper achieved even higher accuracy rates of (97.54%, 93.38%, 94.98%, 97.34%) in rice disease classification. The paper provides a summary of the reasons why ICAI-V4 is superior to other mainstream networks.

1. This study utilized the Candy algorithm to enhance images prior to input classification. The results, as shown in Figure 3, demonstrate that this method significantly improves the visual quality of images compared to traditional filtering methods. The algorithm enhances edge details while suppressing noise, which maintains the authenticity of the image. The results of Table 7 indicate that the classification accuracy of the model for rice diseases improved by 5.21% when using this method compared to no image enhancement.
2. This academic paper introduces a coordinated attention mechanism into the backbone network to enhance the feature capture capability and overall performance of the network model. The experiment in Table 8 shows that the classification accuracy of Inception-V4, AlexNet, and ResNeXt for rice diseases improved after the addition of the coordinated attention mechanism (+7.79%, +1.57%, +2.14%).
3. The INCV trunk structure is a combination of Inception-iv and Reduction-iv structures, with the addition of involution to enable more flexible modeling in spatial dimensions. This approach also incorporates a channel dimension shared kernel with a space-specific kernel to enhance the feature extraction capability of the entire network. As a result, the classification of similar rice disease images is improved. This paper incorporates involution into Inception-V4 and conducts experiments by adding involution to each part. The results, presented in Table 4, demonstrate that this matching enhances the network's feature extraction ability and improves its classification accuracy (81.27%, 82.51%, 82.93%, 85.14%, 85.94%, 87.74%).
4. The use of Leaky ReLU as a replacement for the ReLU activation function has been shown to alleviate the problem of neuron death caused by the latter, while also enhancing the model's elasticity. In fact, Table 8 demonstrates that changing the activation function to Leaky ReLU resulted in a 2.87% improvement in the classification accuracy of rice diseases.
5. According to our research, this method has shown superior classification accuracy compared to other deep neural network models. The average accuracy is 95.57%, with a recall rate of 95.54% and an F1-score of 95.81%.

Although existing methods for detecting rice disease are effective, they may not be as accurate when the diseased leaves are in the early stages and their characteristics have not fully developed. Additionally, the automatic detection of rice disease has not been

widely implemented and real-time dynamic detection has not been thoroughly researched. Current methods mainly focus on detecting disease in collected images. Further research is required to determine how this method can be applied to the large-scale detection of rice planting and disease dynamics. For the future, in the process of promoting this method, it is necessary to combine intelligent Internet facilities such as agricultural Internet of Things and mobile terminal processors to realize the real-time monitoring of grain storage and pest identification, which is conducive to promoting the modernization and intelligence of the agricultural industry.

## 5. Conclusions

This paper presents a method for recognizing and classifying rice diseases using an improved Canny algorithm and ICAI-V4 model. The proposed method first utilizes an image enhancement algorithm based on the improved Canny algorithm to highlight edge details and reduce image noise. The enhanced image is preprocessed using various techniques such as horizontal flipping, vertical flipping, and random cropping to increase the number of datasets. This extended dataset is then fed into the ICAI-V4 network model. The network model is improved by introducing a coordinated attention mechanism and designing an INCV structure, which enhances the ability of the network to extract feature information from rice disease images. The study found that the implementation of the 10-fold cross-validation method resulted in an average classification accuracy of 95.57% for ICAI-V4 when dealing with datasets containing random composite noise. These results demonstrated that the image recognition and classification method for rice disease, which is based on the improved Canny algorithm and ICAI-V4 model, has a strong ability to resist interference and is feasible for real-world applications.

The identification and classification of plant diseases using image recognition and classification technology remains a crucial research area for improving the agricultural economy. Despite the remarkable achievements made by deep learning, there is still a need for further improvement in this field. The number of rice disease samples collected in this study is insufficient. To improve the model's generalization ability and better serve the needs of agricultural development, it is necessary to expand the dataset and increase the testing scale.

**Author Contributions:** N.Z.: conceptualization, data curation, writing—original draft, formal analysis and resources. G.Z.: funding acquisition, investigation, and software. G.G.: methodology, project administration, supervision, and validation. C.H.: project administration. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Scientific Research Project of Education Department of Hunan Province (Grant No. 21A0179); in part by Changsha Municipal Natural Science Foundation (Grant No. kq2014160); in part by the Natural Science Foundation of Hunan Province (Grant No. 2021JJ41087); in part by the Natural Science Foundation of China (Grant No. 61902436); and in part by Hunan Key Laboratory of Intelligent Logistics Technology (2019TP1015).

**Data Availability Statement:** All data are presented in the article.

**Acknowledgments:** We are grateful to all members of the Food Academy of Central South University of Forestry and Technology for their advice and assistance in the course of this research.

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

## References

1. Yang, X.; Pan, Y.; Qing, D.; Xia, X.; Zhou, W.; Chen, W.; Li, J.; Liang, H.; Yang, X.; Li, D.; et al. Dissecting the Molecular Basis for Genetic Improvement of Rice, 22 June 2022, PREPRINT (Version 1) Available at Research Square. Available online: <https://doi.org/10.21203/rs.3.rs-1749870/v1> (accessed on 27 January 2023).
2. Li, P.; Chen, Y.H.; Lu, J.; Zhang, C.Q.; Liu, Q.Q.; Li, Q.F. Genes and their molecular functions determining seed structure, components, and quality of rice. *Rice* **2022**, *15*, 18. [[CrossRef](#)] [[PubMed](#)]
3. Yang, X.; Xu, H.; Zang, H.; Feng, X.; Pan, R.; Gu, C.; Gao, T. Current situation and countermeasures of rice disease prevention and control. *Mod. Pestic.* **2022**, *21*, 1–5. [[CrossRef](#)]

4. Guo, D. Research on Identification Method of Rice Sheath Blight Based on Deep Learning. Ph.D. Thesis, Heilongjiang Bayi Agricultural University, Mishan, China, 2020. [\[CrossRef\]](#)
5. Khanal, S.; Fulton, J.; Shearer, S. An overview of current and potential applications of thermal remote sensing in precision agriculture. *Comput. Electron. Agric.* **2017**, *139*, 22–32. [\[CrossRef\]](#)
6. Jagyasi, B.; Mohite, J.; Pappula, S. Applications of mobile sensing technologies in precision agriculture. *CSI Commun.* **2013**, *18*, 21–23.
7. Sartori, D.; Brunelli, D. A smart sensor for precision agriculture powered by microbial fuel cells. In Proceedings of the 2016 IEEE Sensors Applications Symposium (SAS), Catania, Italy, 20–22 April 2016; pp. 1–6. [\[CrossRef\]](#)
8. Puri, V.; Nayyar, A.; Raja, L. Agriculture drones: A modern breakthrough in precision agriculture. *J. Stat. Manag. Syst.* **2017**, *20*, 507–518. [\[CrossRef\]](#)
9. Mekala, M.S.; Viswanathan, P. A Survey: Smart agriculture IoT with cloud computing. In Proceedings of the 2017 International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore, India, 10–12 August 2017; pp. 1–7. [\[CrossRef\]](#)
10. Jha, K.; Doshi, A.; Patel, P.; Shah, M. A comprehensive review on automation in agriculture using artificial intelligence. *Artif. Intell. Agric.* **2019**, *2*, 1–12. [\[CrossRef\]](#)
11. Ahmed, N.; De, D.; Hussain, I. Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas. *IEEE Internet Things J.* **2018**, *5*, 4890–4899. [\[CrossRef\]](#)
12. Huynh-The, T.; Gadekallu, T. R.; Wang, W.; Yenduri, G.; Ranaweera, P.; Pham, Q.V.; da Costa, D.B.; Liyanage, M. Blockchain for the metaverse: A Review. *Future Gener. Comput. Syst.* **2023**, *143*, 401–419. [\[CrossRef\]](#)
13. Khattab, A.; Habib, S.E.; Ismail, H.; Zayan, S.; Fahmy, Y.; Khairy, M.M. An IoT-based cognitive monitoring system for early plant disease forecast. *Comput. Electron. Agric.* **2019**, *1*, 105028. [\[CrossRef\]](#)
14. Yamagishi, Y.; Kato, Y.; Ninomiya, S.; Guo, W. Image-Based Phenotyping for Non-Destructive In Situ Rice (*Oryza sativa* L.) Tiller Counting Using Proximal Sensing. *Sensors* **2022**, *22*, 5547. [\[CrossRef\]](#)
15. Lv, L.; Peng, M.; Wang, X.; Wu, Y. Multi-scale information fusion network with label smoothing strategy for corneal ulcer classification in slit lamp images. *Front. Neurosci.* **2022**, *16*, 993234. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Liu, X.; Hu, Y.; Zhou, G.; Cai, W.; He, M.; Zhan, J.; Hu, Y.; Li, L. DS-MENet for the classification of citrus disease. *Front. Plant Sci.* **2022**, *13*, 88446. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Bera, T.; Das, A.; Sil, J.; Das, A.K. A Survey on Rice Plant Disease Identification Using Image Processing and Data Mining Techniques. *Emerg. Technol. Data Min. Inf. Secur.* **2018**, *3*, 365–376. [\[CrossRef\]](#)
18. Polesel, A.; Ramponi, G.; Mathews, V.J. Image enhancement via adaptive unsharp masking. *IEEE Trans. Image Process.* **2000**, *9*, 505–510. [\[CrossRef\]](#)
19. Phadikar, S.; Sil, J.; Das, A.K. Rice diseases classification using feature selection and rule generation techniques. *Comput. Electron. Agric.* **2013**, *90*, 76–85. [\[CrossRef\]](#)
20. Kumawat, A.; Panda, S. A robust edge detection algorithm based on feature-based image registration (FBIR) using improved canny with fuzzy logic (ICWFL). *Vis. Comput.* **2022**, *38*, 3681–3702. [\[CrossRef\]](#)
21. Xuan, L.; Hong, Z. An improved canny edge detection algorithm. In Proceedings of the 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 24–26 November 2017; pp. 275–278. [\[CrossRef\]](#)
22. Kalbasi, M.; Nikmehr, H. Noise-robust, reconfigurable canny edge detection and its hardware realization. *IEEE Access* **2020**, *8*, 39934–39945. [\[CrossRef\]](#)
23. Wang, Y.Y.; Zhou, Z.G.; Luo, L.K. Image enhancement algorithm based on sobel operator filtering. *Comput. Appl. Softw.* **2019**, *36*, 184–188. [\[CrossRef\]](#)
24. Lu, Y.; Yi, S.; Zeng, N.; Liu, Y.; Zhang, Y. Identification of rice diseases using deep convolutional neural networks. *Neurocomputing* **2017**, *267*, 378–384. [\[CrossRef\]](#)
25. Atole, R.R.; Park, D. A multiclass deep convolutional neural network classifier for detection of common rice plant anomalies. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 1.
26. Zhou, G.; Zhang, W.; Chen, A.; He, M.; Ma, X. Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion. *IEEE Access* **2019**, *7*, 143190–143206. [\[CrossRef\]](#)
27. 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. [\[CrossRef\]](#)
28. 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. [\[CrossRef\]](#)
29. Haque, M.E.; Rahman, A.; Junaeid, I.; Hoque, S.U.; Paul, M. Rice leaf disease classification and detection using yolov5. *arXiv* **2022**, arXiv:2209.01579. [\[CrossRef\]](#)
30. Venkatesan, R.; Er, M.J.; Dave, M.; Pratama, M.; Wu, S. A novel online multi-label classifier for high-speed streaming data applications. *Evol. Syst.* **2017**, *8*, 303–315. [\[CrossRef\]](#)
31. Rong, W.; Li, Z.; Zhang, W.; Sun, L. An improved Canny edge detection algorithm. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014; pp. 577–582. [\[CrossRef\]](#)
32. Li, R.; Song, B.; Liu, X.; Pan, X.; Han, L.; Xu, Y. Smoking Behavior Detection Based on TF-YOLOv5. In Proceedings of the 2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML), Chengdu, China, 15–17 July 2022; pp. 14–19. [\[CrossRef\]](#)

33. Qaddoori, A.S.; Saud, J.H. Comparison of edge discrimination of microbubbles between Sobel and improved mathematics of Canny edge detection. *Int. J. Nonlinear Anal. Appl.* **2022**, *13*, 817–829. [[CrossRef](#)]
34. Li, D.; Hu, J.; Wang, C.; Li, X.; She, Q.; Zhu, L.; Zhang, T.; Chen, Q. Involution: Inverting the inherence of convolution for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12321–12330. [[CrossRef](#)]
35. Pan, C.; Fei, S. Violence detection based on attention mechanism. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 6438–6443. [[CrossRef](#)]
36. Cheng, S.; Cheng, H.; Yang, R.; Zhou, J.; Li, Z.; Shi, B.; Lee, M.; Ma, Q. A High Performance Wheat Disease Detection Based on Position Information. *Plants* **2023**, *12*, 1191. [[CrossRef](#)]
37. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722. [[CrossRef](#)]
38. Guo, G.; Zhang, Z. Road damage detection algorithm for improved YOLOv5. *Sci. Rep.* **2022**, *12*, 15523. [[CrossRef](#)]
39. Li, M. Dual Channel with Involution for Long-Tailed Visual Recognition. *Open J. Appl. Sci.* **2022**, *12*, 421–433. [[CrossRef](#)]
40. Lei, X.; Lin, X.; Zhong, Y.; Chen, Q. Optimization of neural network structure using involution operator based on particle swarm optimization for image classification. In Proceedings of the 2021 11th International Conference on Information Technology in Medicine and Education (ITME), Wuyishan, China, 19–21 November 2021; pp. 168–173. [[CrossRef](#)]
41. Liu, J.; Yin, L.; Hu, Y.; Lv, S.; Sun, L. A Novel Intrusion Detection Algorithm for Industrial Control Systems Based on CNN and Process State Transition. In Proceedings of the 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), Orlando, FL, USA, 17–19 November 2018; pp. 1–8. [[CrossRef](#)]
42. Tarek, H.; Aly, H.; Eisa, S.; Abul-Soud, M. Optimized deep learning algorithms for tomato leaf disease detection with hardware deployment. *Electronics* **2022**, *11*, 140. [[CrossRef](#)]
43. Chen, C.-C.; Ba, J.Y.; Li, T.J.; Ki Chan, C.C.; Wang, K.C.; Liu, Z. EfficientNet: A low-bandwidth iot image sensor framework for cassava leaf disease classification. *Sens. Mater.* **2021**, *33*, 4031–4044. [[CrossRef](#)]
44. Huang, G.; Liu, Z.; Laurens, V.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017. [[CrossRef](#)]
45. Sunil, C.K.; Jaidhar, C.D.; Patil, N. Cardamom plant disease detection approach using EfficientNetV2. *IEEE Access* **2022**, *10*, 789–804. [[CrossRef](#)]
46. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [[CrossRef](#)]
48. Xie, S.; Girshick, R.; Dollar, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [[CrossRef](#)]
49. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017. [[CrossRef](#)]
50. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [[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.