agronomy

MDPI

Article

Rice Growth-Stage Recognition Based on Improved YOLOv8 with UAV Imagery

Wenxi Cai [1,†], Kunbiao Lu [1,†], Mengtao Fan [1], Changjiang Liu [1], Wenjie Huang [1], Jiaju Chen [1], Zaoming Wu [1], Chudong Xu [1], Xu Ma [2,3] and Suiyan Tan [1,*]

[1] College of Electronic Engineering, South China Agricultural University, Guangzhou 510642, China; stevenmedition@gmail.com (W.C.); kunbiaolu@stu.scau.edu.cn (K.L.); fan15138377806@163.com (M.F.); liu181141@163.com (C.L.); 18697752762@163.com (W.H.); 15555048740@163.com (J.C.); wzm1117@stu.scau.edu.cn (Z.W.); cd79cd@126.com (C.X.)
[2] College of Engineering, South China Agricultural University, Guangzhou 510642, China; maxu1959@scau.edu.cn
[3] College of Mechanical and Electrical Engineering, Xinjiang Agricultural University, Urumqi 830001, China
* Correspondence: tansuiyan@scau.edu.cn; Tel.: +86-13533522850
† These authors contributed equally to this work.

Abstract: To optimize rice yield and enhance quality through targeted field management at each growth stage, rapid and accurate identification of rice growth stages is crucial. This study presents the Mobilenetv3-YOLOv8 rice growth-stage recognition model, designed for high efficiency and accuracy using Unmanned Aerial Vehicle (UAV) imagery. A UAV captured images of rice fields across five distinct growth stages from two altitudes (3 m and 20 m) across two independent field experiments. These images were processed to create training, validation, and test datasets for model development. Mobilenetv3 was introduced to replace the standard YOLOv8 backbone, providing robust small-scale feature extraction through multi-scale feature fusion. Additionally, the Coordinate Attention (CA) mechanism was integrated into YOLOv8's backbone, outperforming the Convolutional Block Attention Module (CBAM) by enhancing position-sensitive information capture and focusing on crucial pixel areas. Compared to the original YOLOv8, the enhanced Mobilenetv3-YOLOv8 model improved rice growth-stage identification accuracy and reduced the computational load. With an input image size of 400 × 400 pixels and the CA implemented in the second and third backbone layers, the model achieved its best performance, reaching 84.00% mAP and 84.08% recall. The optimized model achieved parameters and Giga Floating Point Operations (GFLOPs) of 6.60M and 0.9, respectively, with precision values for tillering, jointing, booting, heading, and filling stages of 94.88%, 93.36%, 67.85%, 78.31%, and 85.46%, respectively. The experimental results revealed that the optimal Mobilenetv3-YOLOv8 shows excellent performance and has potential for deployment in edge computing devices and practical applications for in-field rice growth-stage recognition in the future.

Keywords: rice growth stages; YOLOv8; Mobilenetv3; attention mechanism; coordinate attention

Citation: Cai, W.; Lu, K.; Fan, M.; Liu, C.; Huang, W.; Chen, J.; Wu, Z.; Xu, C.; Ma, X.; Tan, S. Rice Growth-Stage Recognition Based on Improved YOLOv8 with UAV Imagery. *Agronomy* **2024**, *14*, 2751. https://doi.org/10.3390/agronomy14122751

Academic Editor: Peng Fu

Received: 16 October 2024
Revised: 14 November 2024
Accepted: 19 November 2024
Published: 21 November 2024

1. Introduction

Rice is one of the most important food crops in the world, widely cultivated in 122 countries, with its presence ranging across Asia, Europe, the Americas, Africa, and Oceania. More than 50% of the global population relies on rice as a staple food. So, to ensure the high quality and maximum yield of rice, it is of great importance for growers to timelily and precisely conduct field management at different growth stages, including fertilization, irrigation, and weed and pest control. Therefore, developmental growth monitoring is one of the daily managements in rice cultivation. The tillering, jointing, booting, heading, and filling stages are five key growth stages in the rice growth cycle [1]. Crops exhibit different morphological characteristics at various growth stages, such as size, shape, and

color. In 2001, Karen Moldenhauer et al. [2] studied the growth and development process of rice, including the morphological characteristics and the division of different growth stages. They mentioned that accurately identifying the growth stages of rice is crucial for formulating management strategies: applying nutrients such as nitrogen at key moments during growth stages can stimulate tillering and increase the potential number of panicles, thus affecting the final yield. Specific growth stages, such as the jointing stage, require timely pest and disease monitoring to reduce crop damage. Therefore, rapid and accurate identification of rice growth stages is vital. Currently, traditional methods of identifying rice growth stages mainly rely on manual inspections, which are not only inefficient [3] but also highly subjective, significantly affecting the accuracy of growth-stage identification and hindering the development of scientifically based field management strategies.

Traditional machine learning provides a simple and effective method to identify the growth stages of crops. Tai-Chu Sheng et al. [4] utilized the random forest model in machine learning to study the four stages of rice: the initial stage, the vegetative stage, the generative stage, and the harvest stage. By inputting growth-related factors, the accuracy reached 98.7%, with a macro-F1-score of 98.6%. Ze He et al. [5] used a dual insurance approach of ground detection and satellite images and used Freeman–Durden decomposition parameters from RADARSAT-2 Polarization Synthetic Aperture Radar (PolSAR) data to monitor the growth stages of rice and identify four key phenological stages: transplanting, nutrition, reproduction, and maturity. Yun Shao et al. [6] proposed using multi-temporal RADARSAT data to study the temporal variation of rice backscattering behavior and combined them with satellite images to predict rice yield. Traditional machine learning requires a lot of preprocessing work to extract features manually and has strict requirements for image input. Xiaohang Ma et al. [7] identified up to 28 traits, which is not conducive to rapid recognition work. Although satellite remote sensing is widely used for large-scale estimation of crop growth stages, its low spatial and temporal resolution limits the ability to conduct timely and accurate assessments at the field scale. Moreover, traditional machine learning models often rely on shallow structures, which limits their ability to capture detailed and high-level features in data based on reduced images, thus hindering the improvement of detection accuracy.

Compared with traditional machine learning, deep learning models have more powerful representation learning capabilities through the stacking of multiple layers of neural networks. Many researchers are constantly optimizing and improving algorithm models for various agricultural applications, and some studies have reported their application in detecting important growth stages of crops. For example, Kaixuan Liu et al. [1] proposed a lightweight method for automatically identifying the growth stages of rice called Small-YOLOv5s. The model is based on a dataset captured by a ground camera, replacing the backbone network of YOLOv5s with the feature extraction network MobileNetv3 and replacing standard convolution with lightweight GsConv. Compared with YOLOv5s (5.0) and YOLOv5s-MobileNetv3-Small, this method not only improves accuracy but also significantly reduces model size. This method is used to identify the five stages of greening, tillering, jointing, heading, and milky maturity. However, the recognition accuracy of the improved model in the complex rice field environment still needs to be further improved. Qi Yang et al. [8] proposed a deep learning method based on UAV images to detect the growth stage of rice in real time. Through a convolutional neural network (CNN), the method achieved an accuracy of 83.9% in 82 sample plots and was used to identify six stages of rice leaf development, tillering, stem elongation, inflorescence extraction and flowering, fruit development, and maturity. Jiale Qin et al. [9] chose ResNet-50 as the backbone network for feature extraction to automatically identify multiple developmental stages of rice ears. By optimizing certain learning parameters, the model achieved better performance in recognizing the growth stages of rice, with an accuracy of 87.33%. Sanaz RASTI et al. [10] proposed a method based on close field images, using a convolutional neural network (convnet) to classify the growth stages of wheat and barley. The experimental results showed that the accuracy in the classification tasks for the main growth

stages was 93.5% and 92.2%, respectively. Although the model achieves high accuracy, it has a relatively large demand for training data and computing resources. Yolov5 and YOLOv8 have stronger detection abilities for small objects through fine feature extraction and optimized network structures. YOLOv8 has made various adjustments and optimizations in various aspects based on yolov5 and focuses on improving the reasoning efficiency. It can complete real-time detection with lower computing resources and is suitable for deployment in resource-constrained agricultural environments.

Recent studies are based on close-range shooting (using devices in fixed positions, such as cameras and monitors) and employ deep learning models to detect crop organs. Through the changes in the number of organs detected, the growth stage can be determined. However, crop organs are mostly small and densely distributed and are subject to serious occlusion phenomena. For example, when improving CNN modules, Yuanqin Zhang et al. [11] spent a lot of time discussing how to improve the modules to precisely detect rice panicles and better predict growth stages. On the other hand, UAVs have many advantages. Our research employed a UAV and identified rice growth periods through the overall phenotypic characteristics of the rice field, avoiding errors in the detection of rice panicles. S. K. von Bueren et al. [12] discussed the applicability of four different types of optical UAV sensors (RGB cameras, modified near-infrared cameras, six-band multispectral cameras, and hyperspectral spectrometers) in grassland monitoring in New Zealand. K. von bueren et al. demonstrated the effectiveness of UAVs in precision agriculture applications and proposed that spectral UAVs could overcome the limitations of traditional satellite remote sensing, making data acquisition more efficient and covering a wider range of areas. At the same time, UAVs could collect data without interfering with the growth of vegetation, ensuring the sustainability of the monitoring process, and could provide near-real-time data, which is very suitable for the needs of rice growth-period identification.

Therefore, the use of UAV technology coupled with deep learning techniques has shown promise for automatic crop-growth monitoring. This study aimed to investigate effective and reliable approaches to identify rice at various growth stages in field conditions. To achieve this, comprehensive rice datasets of different growth stages, acquired by a UAV, were built, and then a lightweight and improved YOLOv8 architecture was proposed to identify rice at different growth stages in the complex field environment. After that, the Coordinate Attention (CA) mechanism was integrated into YOLOv8's backbone, and a performance comparison of the Convolutional Block Attention Module (CBAM) was conducted. Finally, the performance of the proposed model was comprehensively evaluated by proper evaluation metrics. In addition, images acquired at different flight heights and cropped to different sizes were evaluated in terms of performance in growth-stage recognition.

2. Materials and Methods

This study summarizes the key processes involved in recognizing rice growth stages, including field image acquisition, image preprocessing, deep learning model development, and performance evaluations (Figure 1). Initially, RGB images of rice at five growth stages were captured using a DJI Phantom4 RTK UAV (DJI Innovations, Shenzhen, China), followed by a series of preprocessing steps to prepare the datasets. Next, an improved YOLOv8 model was proposed to recognize the growth stages of rice after the dataset construction. Finally, the model's performances were assessed and compared with the best-performing model recommended.

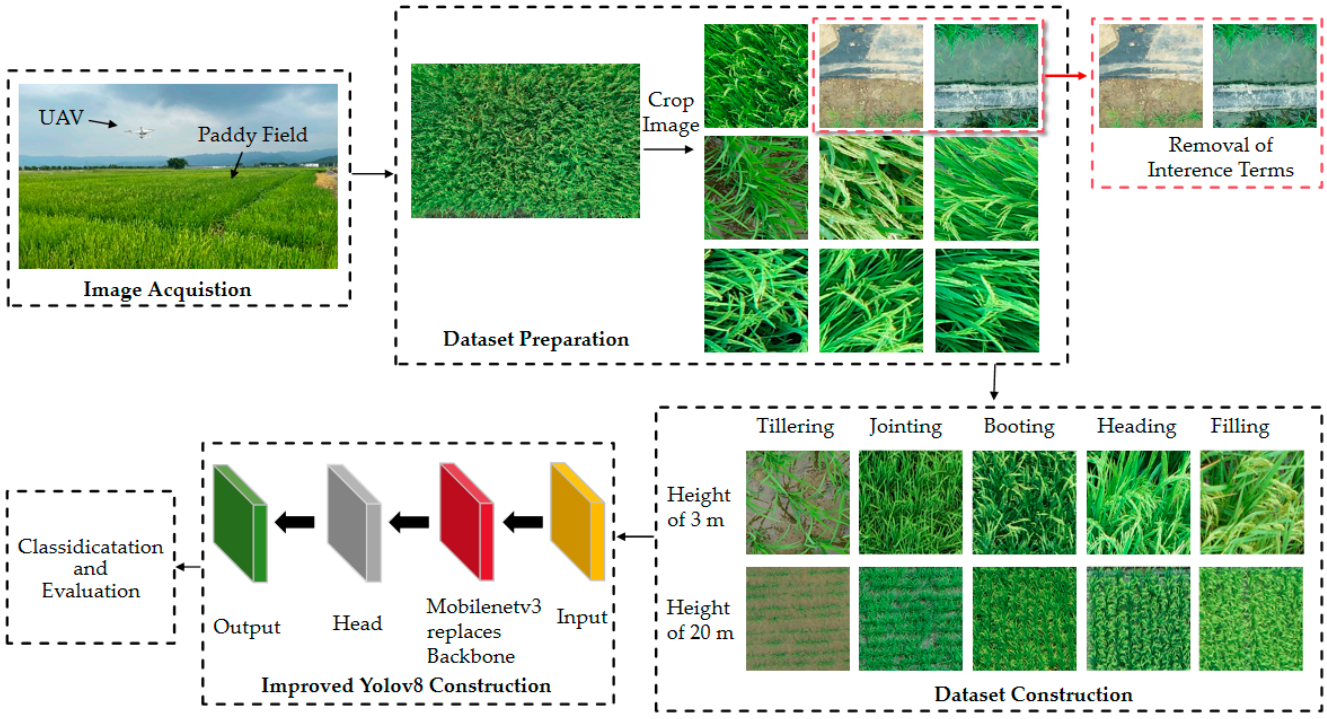


Figure 1. A schematic diagram of the proposed method.

2.1. Data Preprocessing

2.1.1. Field Experiment and Data Acquisition

The field images used in this study were collected from the Shapu Research Center in Zhaoqing, Guangdong Province, China (latitude 23°9′22″ N, longitude 112°39′31″ E), in 2021 and 2022. The planting area is located south of the Tropic of Cancer and belongs to the humid climate zone of the South Asian tropical monsoon. Influenced by the South Asian tropical monsoon, the region has an annual average temperature of 21.93 °C, with an extreme maximum temperature of 37.8 °C and an extreme minimum temperature of 1 °C. The average annual sunshine duration is 1815.72 h, and the average annual rainfall is 1637 mm. The research center annually conducts two extensive field trials on double-cropped rice, including spring-field trials (from March to July 2021 in this study, referred to as EXP.1) and autumn-field trials (from August to November 2021 in this study, referred to as EXP.2). To obtain diverse rice phenotypes at the different growth stages, each plot in the experiment adopted different planting modes, including three rice varieties, five nitrogen fertilizer levels (0, 45, 90, 180, and 270 kg per hectare), and two different planting densities of 30 × 14 cm^2 and 30 × 21 cm^2. Therefore, there were a total of 30 planting modes, with each pattern repeated three times, resulting in a total of 90 planting plots. Each plot was 10.8 × 3.5 square meters. The location of the study and the designs of the double-cropped rice field experiments are shown in Figure 2.

This study employed the DJI Phantom 4 RTK (P4R) UAV (DJI Innovations, Shenzhen, China), produced by DJI Technology Co., Ltd., for image acquisition. The P4R is equipped with an integrated visible-light imaging system featuring a DJI FC6310R camera with a 1-inch, 20-megapixel CMOS sensor (5472 × 3648 pixels) and an 8.8 mm focal length, equivalent to a 24 mm full-frame format. Additionally, the P4R includes a Real-Time Kinematic (RTK) module that provides real-time, centimeter-level positioning data. Considering the impact of lighting conditions and wind on drone photography, we chose to conduct the shooting and sampling on clear and windless days. Rice canopy images were captured at a 3 m height along a horizontal S-shaped trajectory and at a 20 m height with a 45-degree slant. Image acquisition in UAV flights was carried out for five growth stages of rice, namely, the tillering, jointing, booting, heading, and filling stages (Figure 3). The original

image resolution was 5472 × 3648 pixels, and the detailed information of the original datasets is shown in Table 1.

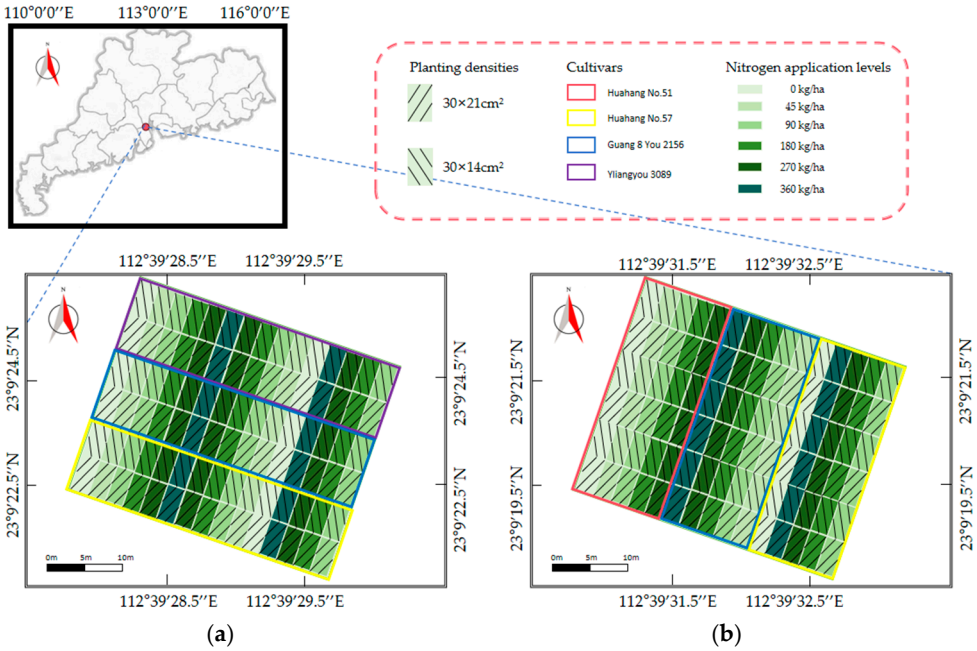


Figure 2. The study site and rice field experiment designs. (**a**) Spring rice field experiment, EXP.1. (**b**) Autumn rice field experiment, EXP.2.

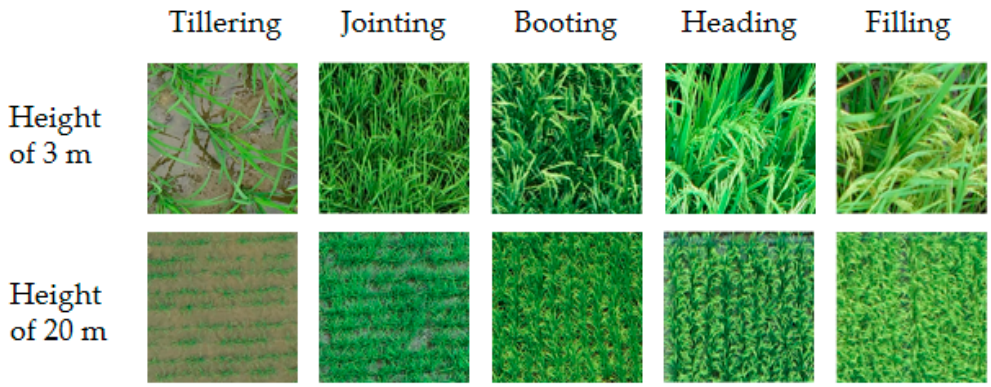


Figure 3. Unmanned Aerial Vehicle photography.

Table 1. Acquisition of multi-spectral images and field data for the two experiments.

Experiment	Acquisition Date (y/m/d)	Growth Stage	Number of Original RGB Images
EXP.1	22 April 2021	Tillering	264
EXP.1	22 May 2021	Jointing	104
EXP.1	9 June 2021	Booting	112
EXP.1	17 June 2021	Heading	295
EXP.1	28 June 2021	Filling	484
EXP.2	7 September 2021	Tillering	570
EXP.2	22 September 2021	Jointing	864
EXP.2	11 October 2021	Booting	513
EXP.2	18 October 2021	Heading	559
EXP.2	26 October 2021	Filling	540

2.1.2. Image Preprocessing

The large size of the original images was not suitable for deep learning model training. Furthermore, some of the original RGB images captured by the UAV included extraneous information, such as roads and fields. Therefore, image preprocessing was essential. The image preprocessing algorithm consisted primarily of three key steps (Figure 1). First, this study cropped the original images into sub-images of equal sizes: 100 × 100 pixels, 200 × 200 pixels, 300 × 300 pixels, 400 × 400 pixels, 500 × 500 pixels, and 600 × 600 pixels. After image cropping, parts of the sub-images contained redundant information; therefore, the Excess Green (ExG) factor was adopted to calculate the proportions of rice plants in the images. Images with less than a threshold of ExG pixels were considered to include redundant information and then were deleted. Ultimately, the remaining images were organized to create the datasets.

Some of the cropped images contained substantial interference factors (e.g., water, soil, and debris), which severely hindered the extraction of image features. To address this, we employed the ExG factor to extract rice plant pixels while constructing the dataset. This approach enabled us to filter the images, retaining those that contained valid rice plant feature information and discarding those with significant background interference [13].

The ExG factor was obtained through a color-component computation method, where the algorithm automatically traversed the image's color components. With human supervision, the best linear combination coefficients for the RGB color components were determined based on the ExG color-component calculations. The formula for the RGB linear combination is provided in Equation (1).

$$\begin{gathered} T(i,j) = \{rR(i,j) + gG(i,j) + bB(i,j)\} \\ r,g,b \in [-3,3] \end{gathered} \tag{1}$$

where $T(i,j)$ represents the characteristics of the result after the linear combination operation; $R(i,j)$, $G(i,j)$, and $B(i,j)$ are the grayscale values of the red, green, and blue color components of the image at (i,j); r, g, and b are the linear coefficients for the color components $R(i,j)$, $G(i,j)$, and $B(i,j)$, respectively; and (i,j) represents the 2-dimensional array variable of the color component.

If $T(i,j) \leq 0$, $T(i,j) = 0$. And if $T(i,j) \geq 255$, $T(i,j) = 255$. Thus, the rules are all eigenvalues in the range from 0 to 255. According to the component operation, when the coefficients are $r = -1$, $g = 2$, and $b = -1$, the coefficient combination has good robustness [14] with respect to the light and color changes of a specific object and can extract the "green" features of rice-growth-period images well. Therefore, the ExG factor calculated by $2G - R - B$ was used as the image segmentation index in this study.

2.1.3. Dataset Construction

In this study, the data were preprocessed using MATLAB (version 2021a; MathWorks, Inc., Natick, MA, USA). The images of EXP.1 and EXP.2 captured by the UAV (from 2 height altitudes for 5 growth stages) were cropped to 6 different pixel sizes (100 × 100, 200 × 200, 300 × 300, 400 × 400, 500 × 500, and 600 × 600). Simultaneously, using the EXG factor, the cropped images were automatically filtered to exclude those containing extensive balks, water bodies, or channels. This process established training and validation sets using the images collected in EXP.1, and an independent test dataset was constructed using images collected in EXP.2. A portion of the rice UAV image dataset, covering five growth stages, was cropped to a size of 400 × 400 pixels (Figure 3 and Table 2).

Table 2. Detailed information of the datasets.

Image Size	Training and Validation Set in EXP.1					Test Sets in EXP.2				
	Tillering	Jointing	Booting	Heading	Filling	Tillering	Jointing	Booting	Heading	Filling
100 × 100 pixels	4000	4000	4000	4000	4000	852	880	946	960	919
200 × 200 pixels	4000	4000	4000	4000	4000	874	907	876	963	688
300 × 300 pixels	4000	4000	4000	4000	4000	759	795	711	781	701
400 × 400 pixels	4000	4000	4000	4000	4000	898	858	958	936	915
500 × 500 pixels	4000	4000	4000	4000	4000	798	781	881	630	643
600 × 600 pixels	4000	4000	4000	4000	4000	971	852	902	946	947

2.2. YOLOv8

Introduction to YOLOv8

You Only Look Once (YOLO) is a deep learning-based end-to-end convolutional neural network [15]. YOLOv8 is a state-of-the-art model developed by Ultralytics, aimed at modularizing and simplifying code blocks, with a user-friendly Application Programming Interface (API). YOLOv8 consists of four parts: Input, Backbone, Neck, and Head (Figure 4). The Backbone uses the C2f structure, inspired by the residual structure of YOLOv5's C3 module [16] and the Efficient Layer Attention Network (ELAN) idea from YOLOv7 [17]. The C2f module, compared to the C3 module, adds more inter-layer branch connections and introduces additional split operations. It removes convolution operations within the branches, ensuring a lightweight network while enhancing the flow of gradient information. The Neck uses a Path Aggregation Network (PAN) to improve feature fusion across different scales [18]. The Head decouples the classification and detection processes, primarily involving loss computation and object-detection box filtering. The loss computation process utilizes the Task-Aligned Assigner positive-sample assignment strategy and loss calculation [19]. It employs universal loss functions, namely, Binary Cross Entropy (BCE) and Complete Intersection over the Union (CIoU) [20]. Therefore, YOLOv8 has made significant advancements in both detection accuracy and speed.

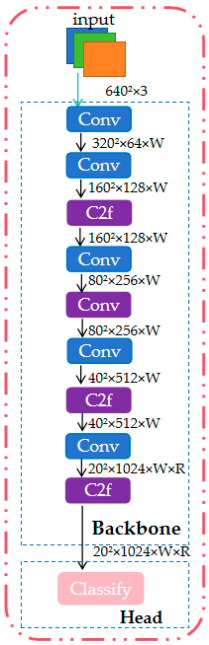


Figure 4. Diagram of YOLOv8 model. Note: The color block in the figure simulates the process of YOLOv8 image input: the image enters the backbone network for feature extraction, passes through the standard convolution and the new C2F convolution structure, and finally enters the image classification function module.

To meet the demands of different usage scenarios, YOLOv8 is divided into five models: YOLOv8n, YOLOv8s, YOLOv8l, YOLOv8m, and YOLOv8x [21], with YOLOv8n having the fewest parameters and the fastest detection speed. Therefore, this study adopted the YOLOv8n model to reduce the size of the model parameters without sacrificing accuracy.

2.3. The Improved YOLOv8 Model

In actual deployment work, the YOLOv8 model is too large and has too many output parameters, which makes it very difficult to deploy the edge end, and it is difficult for small side devices such as UAVs to support the YOLOv8 model. The Mobilenetv3-YOLOv8 model we propose is half the size of the original YOLOv8 model while improving and optimizing its accuracy. First, to reduce the volume of the model and the number of parameters, we replaced the backbone network with Mobilenetv3 to extract information from the images (Figure 5). Secondly, to enhance the features of rice of different growth stages while suppressing irrelevant features, the CA was introduced and compared with the CBAM. Subsequently, the running results of different image pixels and image capture heights were compared. Finally, the attention mechanism was added to the different layers of the backbone network to confirm the optimal model combination.

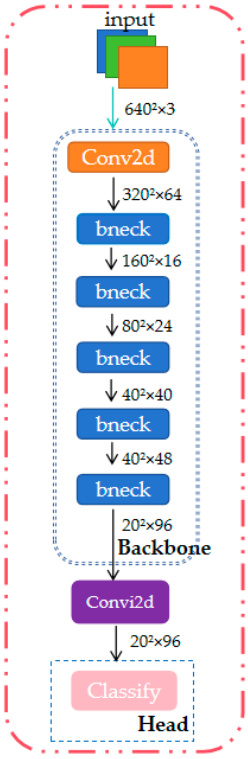


Figure 5. Diagram of Mobilenetv3-YOLOv8 model. Note: Like Figure 4, the backbone part of YOLOv8 in the figure is replaced by Mobilenetv3: Conv2d is a two-dimensional convolution layer. Bneck is a special bottleneck structure of Mobilenetv3.

2.3.1. Mobilenetv3 Model

The MobileNetv3 model retains the depthwise separable convolutions and inverted residual blocks of MobileNetv2 [22] to ensure lightweight characteristics. It improves the bottleneck structure by incorporating SE modules and the hard-swish activation function [23] and optimizes the network architecture to further enhance accuracy. Meanwhile, the MobileNetv3 model enhances the specific information capture capability of each channel through its internal depthwise separable convolution (depthwise convolution combined with pointwise convolution) to blend features across channels [24], thereby improving the characterization ability for rice panicles and leaves. According on the number of resources,

the MobileNetv3 model is divided into the MobileNetv3-large model and the MobileNetv3-small model [25], and the benchmark used in this study was the MobileNetv3-small model. Its structural parameters are shown in Table 3.

Table 3. Mobilenetv3-small model structural parameters.

Input	Operator	Exp Size	#Out	SE	NL	s
$224^2 \times 3$	Conv2d, 3×3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3×3	16	16	√	RE	2
$56^2 \times 16$	bneck, 3×3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3×3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5×5	96	40	√	HS	2
$14^2 \times 40$	bneck, 5×5	240	40	√	HS	1
$14^2 \times 40$	bneck, 5×5	240	40	√	HS	1
$14^2 \times 40$	bneck, 5×5	120	48	√	HS	1
$14^2 \times 48$	bneck, 5×5	144	48	√	HS	1
$14^2 \times 48$	bneck, 5×5	288	96	√	HS	2
$7^2 \times 96$	bneck, 5×5	576	96	√	HS	1
$7^2 \times 96$	bneck, 5×5	576	96	√	HS	1
$7^2 \times 96$	Conv2d, 1×1	-	576	√	HS	1
$7^2 \times 576$	Pool, 7×7	-	-	-	-	1
$1^2 \times 576$	Conv2d, 1×1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	Conv2d, 1×1, NBN	-	k	-	-	1

Note: Exp Size represents the expansion size of the first convolutional layer; #Out represents the number of output feature map channels; Squeeze Excitation(SE) module is an attention mechanism; NL represents the activation function, where HS stands for the hard-swish activation function and RE stands for the ReLU activation function; s represents the stride of the DW convolution; √ indicates the addition of that module; - indicates that the module is not added.

2.3.2. Mobilenetv3-YOLOv8 Model

To address YOLOv8's shortcomings in small-object detection and with large parameter sizes, we integrated Mobilenetv3's feature extraction capabilities with YOLOv8's classification strengths. Employing the MobileNetv3 architecture as an improved backbone network, the Mobilenetv3-YOLOv8 model is proposed. This combination minimizes parameter size without sacrificing accuracy.

2.4. Adding Attention Mechanisms

2.4.1. Convolutional Block Attention Module Mechanism

The Convolutional Block Attention Module (CBAM) consists of two sub-modules: the Channel Attention Module (CAM) and the Spatial Attention Module (SAM), which handle attention operations in the channel and spatial dimensions, respectively [26]. This design not only conserves parameters and computational resources but also guarantees that the CBAM can be incorporated as a plug-and-play module into current network architectures. Figure 6 shows the structure of the CBAM.

The CAM focuses on compressing the spatial dimension while keeping the channel dimension unchanged. First, it uses two parallel max-pooling and average-pooling layers to compress the input feature map from its original size of $C \times H \times W$ to $C \times 1 \times 1$. Then, the feature map is processed by a shared MLP module, which first compresses the number of channels to $1/r$ of the original and then expands back to the original number of channels, followed by a ReLU activation function to obtain two feature maps. Next, these two feature maps are added elementwise and passed through a sigmoid activation function to obtain the output of the channel attention [27]. Finally, this output is multiplied with the original feature map to restore it to the size $C \times H \times W$. The calculation is shown in Equation (2).

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) = \sigma\left(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1(W_0(F^c_{max}))\right) \quad (2)$$

where M_c is the channel attention feature map, σ is the sigmoid function, MLP is the shared multi-layer perceptron, F^c_{avg} is the average-pooled feature, F^c_{max} is the max-pooling feature.

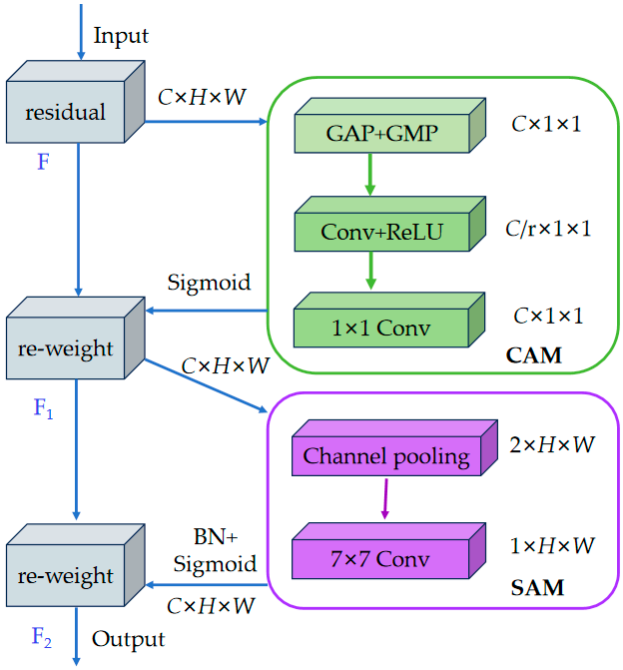


Figure 6. Overview diagram of CBAM mechanism.

SAM focuses on compressing the channel dimension while keeping the spatial dimension unchanged, focusing on the positional information of the target. SAM involves the spatial attention module, which generates two 1 × H × W feature maps through global max-pooling and average-pooling operations on the previously generated feature map F_1. These two feature maps are then fused into a single-channel feature map through a 7 × 7 convolution. After processing by the sigmoid activation function, the spatial attention feature map is obtained [27]. Finally, this feature map is multiplied with the original feature map to generate the attention feature map F_2. The calculation is shown in Equation (3).

$$M_s(F) = \sigma\left(f^{7\times7}([AvgPool(F); MaxPool(F)])\right) = \sigma\left(f^{7\times7}\left(\left[F^s_{avg}; F^s_{max}\right]\right)\right) \quad (3)$$

where M_s is the spatial attention feature map, $f^{7\times7}$ is the 7 × 7 convolution, F^s_{avg} is the average feature of size 1 × H × W, and F^s_{max} is the max-pooling feature of size 1 × H × W.

2.4.2. Coordinate Attention Mechanism

The Coordinate Attention (CA) mechanism, a significant result from CVPR 2021, considers both channel information and direction-related positional information. It is adaptable and lightweight, making it easy to embed into the core modules of lightweight networks, effectively addressing the long-range dependency issues of the CBAM [28].

To capture distant spatial dependencies with accurate positional details, the CA mechanism globally average pooling the input feature map X along the height and width directions to generate feature maps of sizes C × H × 1 and C × 1 × W, as shown in Equation (4).

$$z_c^h(h) = \frac{1}{W}\sum_{0\le i\le W} x_c(h,i), z_c^w(w) = \frac{1}{H}\sum_{0\le j\le H} x_c(j,w) \quad (4)$$

where z_c^h and z_c^w are the outputs of channel c along the height and width directions, respectively, and x_c is the input feature map of channel c.

These feature maps are concatenated and then undergo dimensionality reduction through convolution to generate new feature maps, as shown in Equation (5).

$$f = \delta\left(F_1\left(\left|z^h, z^w\right|\right)\right) \quad (5)$$

where f is the intermediate feature map encoded with spatial information in the horizontal and vertical directions, δ represents the non-linear activation function, and F_1 denotes the 1×1 convolution transformation.

Next, f will be divided along the spatial dimension into f^h and f^w, which are then subjected to dimensionality enhancement using 1×1 convolutions and applying the sigmoid activation function on the spatially divided parts g^h and g^w [29], as shown in Equation (6).

$$g^h = \sigma\left(F_h\left(f^h\right)\right), g^w = \sigma(F_w(f^w)) \quad (6)$$

where g^h and g^w represent the weights in the height and width directions, respectively; σ denotes the sigmoid activation function; and F_h and F_w represent the convolution transformations in the height and width directions, respectively.

The extended attention weights g^h and g^w are multiplied with the input feature map X to perform a weighted operation, yielding the output $Y \in R^{C\times H\times W}$ of the CA module, as shown in Equation (7).

$$y_c(i,j) = x_c(i,j)\cdot g_c^h(i)\cdot g_c^w(j) \quad (7)$$

where y_c represents the output of the c channel (Figure 7).

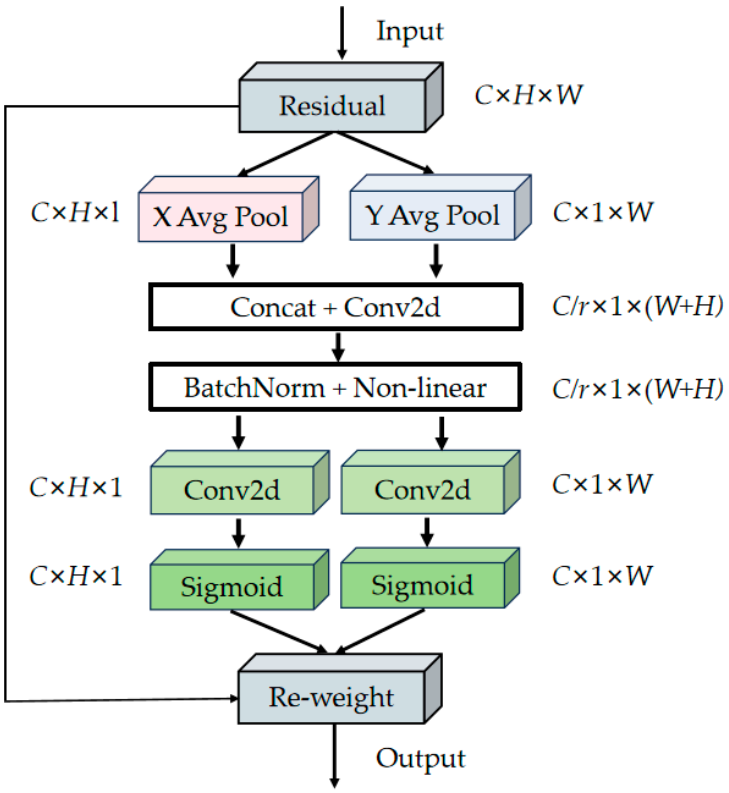


Figure 7. Overview diagram of CA mechanism.

3. Results

3.1. Experimental Platform and Evaluation Metrics

The experiments were conducted on a Windows 11 64-bit operating system, utilizing PyCharm 2022 (JetBrains, Prague, Czech Republic). The hardware setup con-sisted of an NVIDIA GeForce RTX 4090 GPU, a 13th Gen Intel Core i7-13700K 3.40 GHz CPU, and 64 GB of memory (LENOVO, Beijing, China). The open-source deep learning framework PyTorch 2.2.1 + cu118 was used to build the network model.

To evaluate the performance of the proposed rice growth-stage recognition model, Mobilenetv3-YOLOv8, the following evaluation metrics were used: precision (P), recall (R), mean Average Precision (mAP), parameters, and Giga Floating Point Operations (GFLOPs). R reflects the model's ability to find positive samples, P reflects the model's classification ability, mAP reflects the overall performance of the model in detection and classification [30], and the F1-score helps find a balance between precision and recall for good model performance. During evaluation, Average Precision (AP) uses an IoU threshold of 0.5 to assess the overlap between the predicted and actual bounding boxes. mAP is the average of AP values across five growth stages and is used as the final performance metric, as shown in Equations (8)–(12). Parameters refer to the number of trainable parameters in a deep neural network model, measured in millions (M), where 1 M equals 10^6. GFLOPs refers to the number of floating-point operations, used to measure the complexity of a model, and is measured in billions (B), where 1 B equals 10^9 [31].

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{8}$$

$$AP = \int_0^1 P \cdot R dR \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$P = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{12}$$

where TP represents the count of accurately detected positive samples, FP denotes the count of falsely detected positive samples, TN indicates the count of accurately detected negative samples, and N refers to the total number of categories [32].

3.2. Recognition Effect of the Mobilenetv3-YOLOv8 Model on Images of Different Input Size

To verify the recognition capability of the Mobilenetv3-YOLOv8 model on images of different input size, six image sizes (100 × 100, 200 × 200, 300 × 300, 400 × 400, 500 × 500, and 600 × 600) were input into the model for training and validation. Figure 8 indicates that when the input dataset consisted of images of 400 × 400 pixels, the model's precision, recall, and F1-score were significantly higher than for other pixel sizes, with values of 79.20%, 79.30%, and 79.25%, respectively. Therefore, subsequent experiments were conducted using datasets with image sizes of 400 × 400 pixels.

In particular, rice panicles are an important indicator of growth stage. The Ground Sample Distance (GSD) of a UAV flight at a 3 m height is 0.14 cm/pixel. Therefore, image sizes of 100 × 100 pixels, 200 × 200 pixels, 300 × 300 pixels, 400 × 400 pixels, 500 × 500 pixels, and 600 × 600 pixels correspond to field areas of 14 cm^2, 28 cm^2, 42 cm^2, 56 cm^2, 70 cm^2, and 84 cm^2. In the late stages of rice growth, the length of rice panicles is approximately 20–30 cm. It can be inferred that images of 400 × 400 pixels taken at a height of 3 m include appropriate amounts of panicles and are sufficient to identify growth stages. Larger image sizes may encompass more plants and panicles, which may lead to overlooking important local differences and thus reduce the accuracy of rice growth-stage detection.

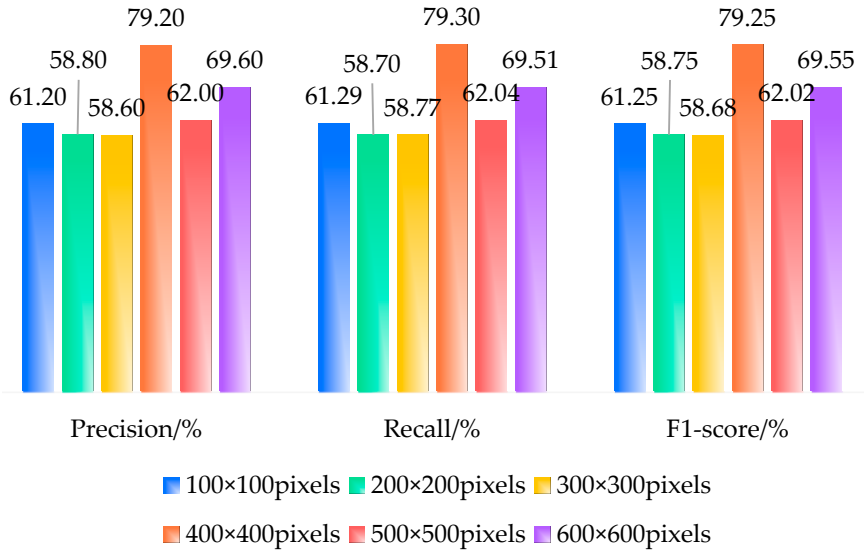


Figure 8. Recognition effect of Mobilenetv3-YOLOv8 model on images of different input sizes.

3.3. Ablation Experiments

Based on the research foundation of the different image parameters mentioned above and maintaining the same quantitative parameters, the study employed 400 × 400-pixel image parameters, which demonstrated the optimal performance of the model, for the following research tasks. To verify the effectiveness of the proposed method, a series of ablation experiments were conducted on the Mobilenetv3-YOLOv8 model. The base network, YOLOv8, was used with its backbone network replaced by the Mobilenetv3 network, and the CA and the CBAM were added to compare their performances. The results are shown in Table 4.

Table 4. Results of ablation experiments.

YOLOv8	Mobilenetv3	Attention Mechanisms	P/%					R/%					mAP/%
			Tillering	Jointing	Booting	Heading	Filling	Tillering	Jointing	Booting	Heading	Filling	
√	-	-	93.62	99.38	50.03	75.83	68.31	60.47	75.06	79.65	77.78	63.61	71.40
√	√	-	97.28	81.53	58.37	88.53	86.98	47.77	97.2	76.41	88.25	86.89	79.20
√	√	CBAM	98.79	88.11	69.71	83.19	68.46	90.98	89.86	82.88	73.96	66.67	81.00
√	√	CA	96.24	87.02	83.96	84.27	66.64	91.2	95.34	72.65	60.68	92.57	82.60

First, using the Mobilenetv1, Mobilenetv2, Mobilenetv3_small, and Mobilenetv3_large networks alone for rice growth-stage recognition resulted in mAP values of 33.34%, 44.39%, 57.07%, and 53.30%, respectively, with parameter sizes of 4.23 M, 3.50 M, 2.55 M, and 5.48 M (Figure 9). The highest mAP of 57.07% and the smallest parameter size of 2.55 M were achieved by the Mobilenetv3_small network. The baseline YOLOv8 model achieved a mAP of 71.4%. Replacing the YOLOv8 backbone network with the Mobilenetv3_small network increased the mAP to 79.2%. Compared to the baseline YOLOv8 model, the performance improved by 7.8 percentage points, indicating that replacing the YOLOv8 backbone network with Mobilenetv3_small allows the model to more accurately identify the growth stages of rice. Subsequently, incorporating the CBAM and the CA resulted in mAP values of 81% and 82.6%, respectively, representing increases of 9.6 and 11.2 percentage points compared to the baseline model. This demonstrates that the attention mechanisms effectively suppress irrelevant features, with the CA outperforming the CBAM. Compared to the Mobilenetv3-YOLOv8 model, the addition of the CA increased the model's parameter count by only 0.00648 M, a negligible amount, and did not alter the model's computation speed, which remained at 0.9 (Figure 10). This indicates that adding the CA can effectively enhance the performance of the Mobilenetv3-YOLOv8 model.

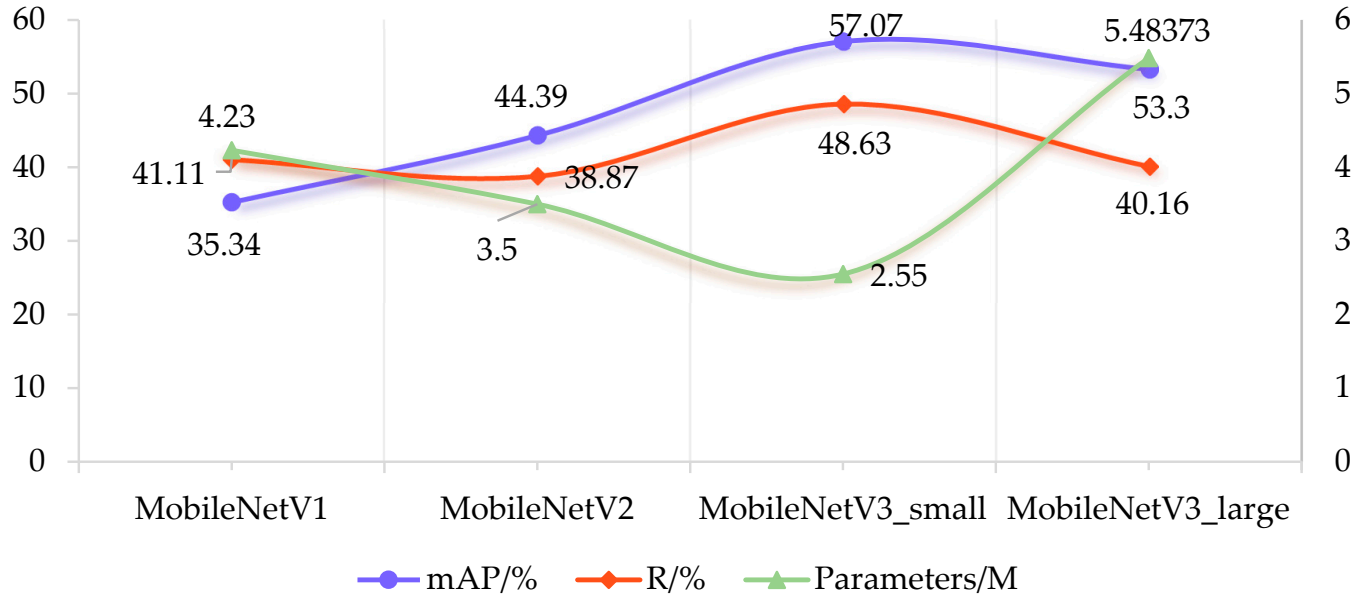


Figure 9. Performance comparison of different Mobilenet networks.

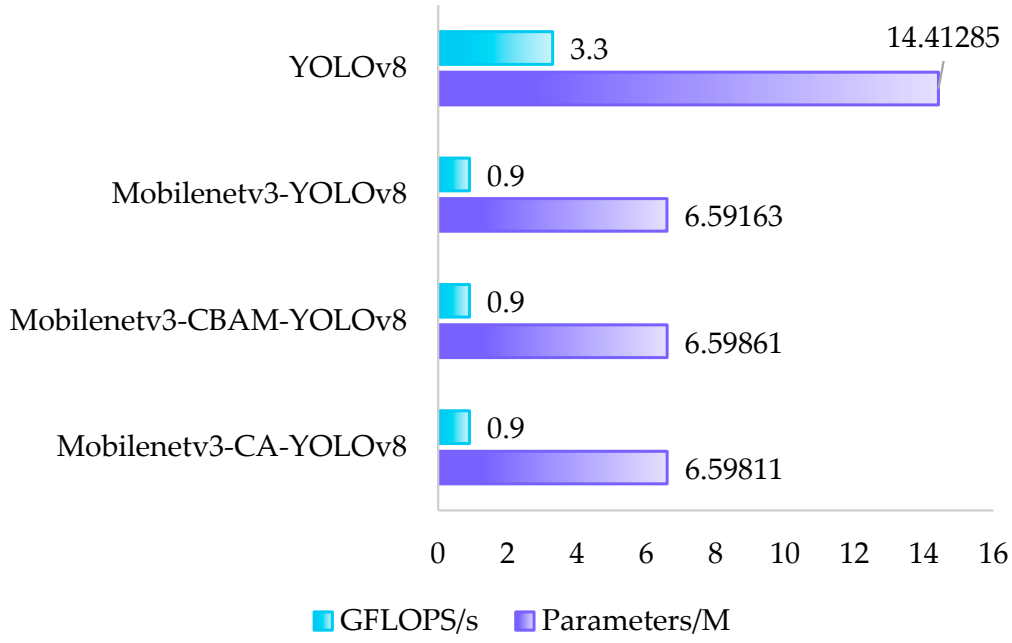


Figure 10. Performance comparison of different models.

3.4. Impact of Attention Mechanisms at Different Layers of the Network

Similarly, the following research content was based on a uniform image parameter quantification at 400 × 400 pixels. After determining the Mobilenetv3-YOLOv8 model, a further comparison was made to evaluate the influence of adding the CA or the CBAM on the model's recognition capability. During the experiments, the number and positions of the CA and the CBAM in the backbone network were varied to compare the recognition performance of the Mobilenetv3-YOLOv8 model for rice growth stages. The layers and positions of the attention mechanisms are shown in Figure 11, using the CA as an example.

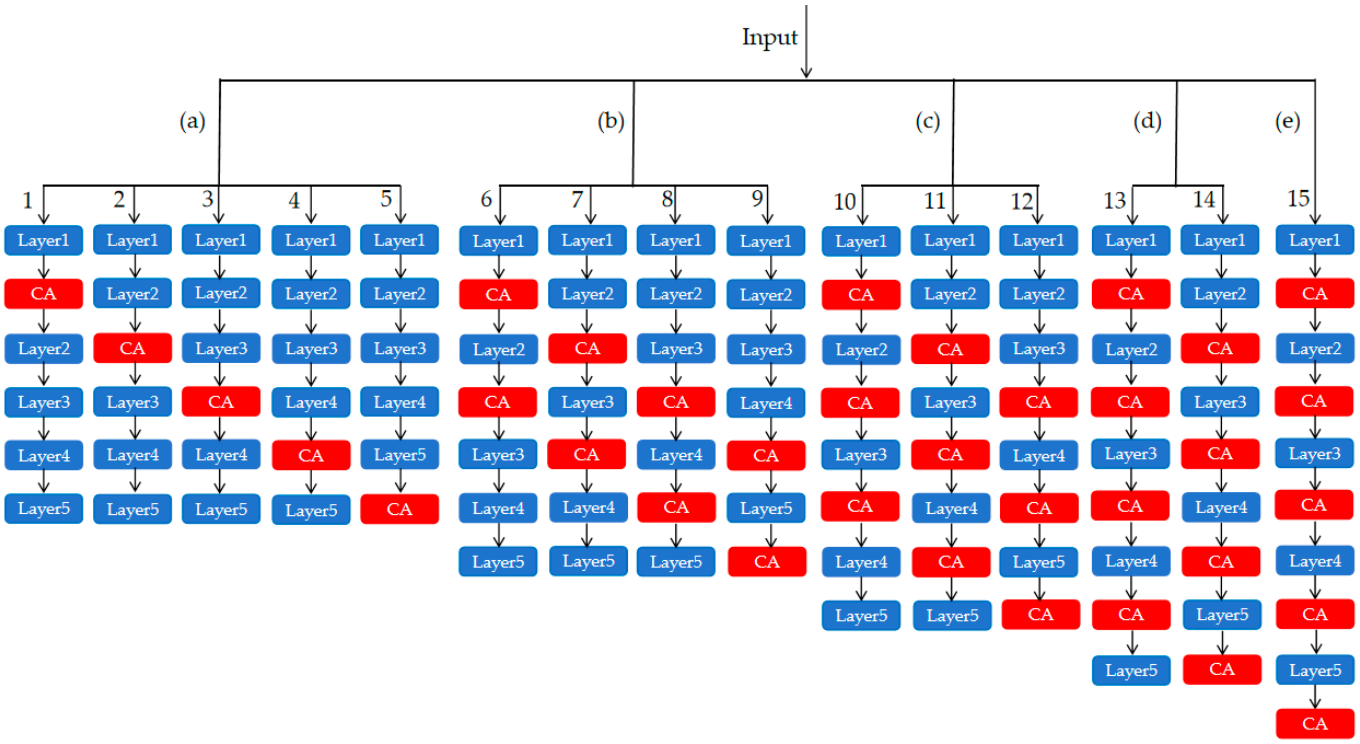


Figure 11. Location map of CA mechanism. Notes: Subfigures (**a**–**e**) are five different discussions on adding one layer of attention mechanism, two layers, three layers, four layers and five layers to the backbone network. The blue color block is the backbone network work layer. The first layer adopts a bottleneck, which includes a 3 × 3 convolution, and the input feature has a spatial dimension of 320 × 320 and consists of 16 channels. The second layer adopts a bottleneck, which includes a 3 × 3 convolution, and the input feature has a spatial dimension of 160 × 160 and consists of 16 channels. The third layer adopts a bottleneck, which includes a 5 × 5 convolution, and the input feature has a spatial dimension of 80 × 80 and consists of 24 channels. The forth layer adopts a bottleneck, which includes a 5 × 5 convolution, and the input feature has a spatial dimension of 40 × 40 and consists of 48 channels. The fifth layer adopts a bottleneck, which includes a 5 × 5 convolution, and the input feature has a spatial dimension of 20 × 20 and consists of 96 channels. The red color block is the added CA attention mechanism layer.

When a single-layer attention mechanism was added, as shown in Figure 11a, the performance of the model was optimal with the CA at position5, achieving a mAP of 82.6% and a recall of 82.49%. For the CBAM, the optimal performance was at position2, with a mAP of 79.40% and a recall of 79.23%.

When adding a two-layer attention mechanism, as shown in Figure 11b, the model's performance was optimal with the CA at position7, achieving a mAP of 84.00% and a recall of 84.08%. For the CBAM, the optimal performance was at position6, with a mAP of 73.60% and a recall of 73.48%.

For a three-layer attention mechanism, as shown in Figure 11c, the optimal performance with the CA was at position12, achieving a mAP of 75.20% and a recall of 75.23%. The CBAM 's optimal performance was at position10, with a mAP of 74.20% and a recall of 74.02%.

When a four-layer attention mechanism was added, as shown in Figure 11d, the CA at position14 yielded the best performance, with a mAP of 79.00% and a recall of 78.83%. The CBAM at position14 achieved a mAP of 74.40% and a recall of 74.41%.

For a five-layer attention mechanism, as shown in Figure 11e, the optimal performance with the CA was at position15, with a mAP of 71.60% and a recall of 71.49%. The CBAM at position15 achieved a mAP of 73.20% and a recall of 73.03%.

It is evident from Table 5 that with a two-layer attention mechanism, the CA at position7 achieved the highest mAP and recall rates of 84.00% and 84.08%, respectively, representing the global optimal performance of the model.

Table 5. Results of attention mechanisms integrated at different layers of the network.

Model	Mechanism in Backbone Network (Index)	CA		CBAM	
		Mean Average Precision (mAP)/%	Recall Rate (R)/%	Mean Average Precision (mAP)/%	Recall Rate (R)/%
YOLOv8n	/	71.40	71.31	71.40	71.31
(a) Mobilenet Backbone Network with One Attention Mechanism Integrated into Layers	1	76.61	76.64	77.20	77.12
	2	73.60	73.63	79.40	79.23
	3	71.00	71.03	73.00	72.97
	4	70.00	69.85	72.60	72.69
	5	82.60	82.49	72.80	72.73
(b) Mobilenet Backbone Network with Two Attention Mechanisms Integrated into Layers	6	84.00	83.97	73.60	73.48
	7	84.00	84.08	66.20	66.17
	8	69.00	68.85	63.40	63.48
	9	62.60	62.54	67.00	66.89
(c) Mobilenet Backbone Network with Three Attention Mechanisms Integrated into Layers	10	73.00	73.27	74.20	74.02
	11	74.20	74.25	67.20	67.26
	12	75.20	75.23	69.00	68.94
(d) Mobilenet Backbone Network with Four Attention Mechanisms Integrated into Layers	13	68.20	68.03	72.80	72.95
	14	79.00	78.83	74.40	74.41
(e) Mobilenet Backbone Network with Five Attention Mechanisms Integrated into Layers	15	71.60	71.49	73.20	73.03

Based on the above data, it was indicated that the mAP and recall rates decrease as the position of the attention mechanism in the backbone network deepens. This analysis suggests that during the multi-layer network processing of the model, image feature information becomes increasingly blurred, and some information may be lost. Therefore, adding attention mechanisms in the early stages of the backbone network can better capture image feature information and improve network precision.

Comparing the fusion of the attention mechanism at position1, -6, -10, and -13, it can be observed that adding the attention mechanism at the first and second fusion positions can enhance the mAP from the original model's 71.40% to 76.61% for position1 and further from 73.60% for position1 to 84.00% for position6 and position7. Adding the attention mechanism at the third and fourth fusion positions results in a decline in mAP from 84.00% for position6 to 73.00% for position10 and from 73.00% for position10 to 68.20% for position13. This analysis suggests that as the network deepens, image feature information is overlaid by feature map information, making it difficult for the attention mechanism to capture useful image information. However, an excessive number of attention mechanisms can overly enhance image information, leading to feature information redundancy and overfitting, which reduces network precision.

Overall, when the CA was fused in position7, Mobilenetv3-YOLOv8 yielded the best performance with mAP and recall values of 84.00% and 84.08%, respectively. Considering the CBAM, the best performance was found in position2, with mAP and recall values of 79.40% and 79.24%, respectively.

3.5. Recognition Ability of the Mobilenetv3-YOLOv8 Model for UAV Imageries Taken at Different Heights

In this study, a UAV was used to capture images of rice growth stages at different heights, namely, 3 m and 20 m. The Mobilenetv3-YOLOv8 model was used to analyze these images, and the results are shown in Table 6. For images taken at a height of 3 m, the model achieved an accuracy of 82.60%, a recall of 84.08%, and a mAP of 84.00%. For images taken at a height of 20 m, the model's accuracy was 67.20%, the recall was 67.08%, and the mAP was 67.20%. Comparing the two, the model's accuracy, recall, and mAP for images taken at 3 m were 16.8, 17.0, and 19.0 percentage points higher, respectively, than for images taken at 20 m. This may have been because rice images collected at lower altitudes contain detailed information about rice phenotypes, resulting in the Mobilenetv3-YOLOv8 model exhibiting higher recognition accuracy.

Table 6. The recognition ability of the model for images taken at different heights.

Period	3 m			20 m		
	P/%	R/%	mAP/%	P/%	R/%	mAP/%
Tillering	99.07	94.88	84.00	99.60	84.64	67.20
Jointing	83.00	93.36		61.12	50.25	
Booting	82.70	67.85		57.49	94.23	
Heading	77.62	78.20		99.80	60.48	
Filling	77.45	85.57		71.99	76.26	

4. Discussion

Many factors affect the performance of the deep learning models in rice growth-stage recognition, for example, the size of input images (Figure 8), the backbone network (Figure 9), the attention mechanism (Table 4), the position at which the attention mechanism is integrated (Table 5), and the flight altitude (Table 6). With comparative performance evaluation of the improved YOLOv8 model, Mobilenetv3-YOLOv8 demonstrated a significant improvement in rice growth-stage recognition and a reduction in the computation of the network after several improvements were implemented. The best performance of Mobilenetv3-YOLOv8 in the recognition of five rice growth stages was found with mAP and recall values of 84.00% and 84.08%, respectively, when the input image size was set at 400 × 400 pixels, Mobilenetv3 was selected as the backbone network, and the CA was integrated in the second and the third layers of the backbone network.

4.1. The Confusion Matrix of Mobilenetv3-YOLOv8

A confusion matrix is a visualization tool for supervised learning that shows how well a model classifies across categories. A confusion matrix shows the percentages of correct and incorrect categorizations for each category and the categories that the model predicts incorrectly. For the optimal Mobilenetv3-YOLOv8 model, a confusion matrix using the test dataset is displayed in Figure 12. Interestingly, most of the errors occur in adjacent classes along the developmental order. These are cases where the two adjacent growth stages exhibit similar characteristics in rice phenotypes and the human eye may experience uncertainty in determining the precise growth stage when switching from one class to another. However, there are errors in non-adjacent classes along the developmental order. There are two main reasons for this. First, rice is planted in plots, and a guard row of ridges is constructed along the boundary between two adjacent plots and a plastic cover is placed over the ridges to prevent the diffusion of fertilizers among plots. Furthermore, a planting area measuring 1 m in width was created around the experimental site. Therefore, the images collected by the UAV inevitably contained a large amount of redundant information, including ridges and roads. Although image prepossessing was performed to exclude images with redundant information, some images that contained ridges were left and were prone to being wrongly classified. An image of the booting stage was falsely recognized as

the tillering stage and an image of the filling stage was falsely recognized as the jointing stage because there was road and ridge information in the field images (Figure 13a,b). Another reason for misidentification of growth stages is that some images were collected under strong sunlight reflection and the rice panicles were overexposed, making it hard to identify them (Figure 14), resulting in one instance in an image of the booting stage being falsely recognized as the tillering stage. Different weather conditions and climatic factors can impact the accuracy of the model's detection. The dataset in this study included only RGB images captured by a spectral drone under clear, windless conditions and lacked image data for varying lighting, temperature, wind speed, humidity, and cloudy weather. Consequently, the model's adaptability to diverse weather conditions remains limited.

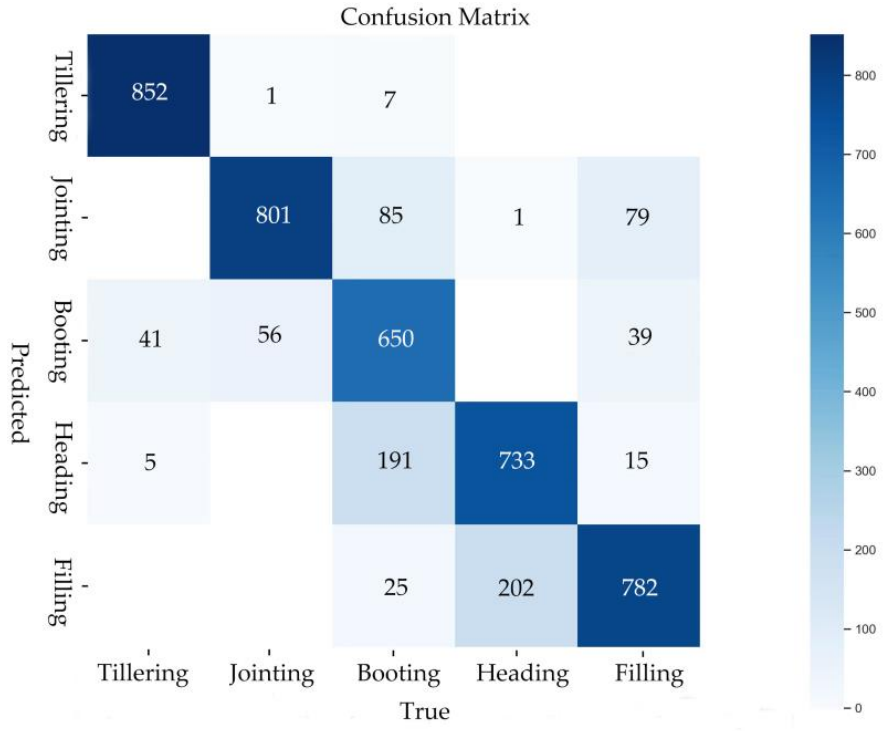
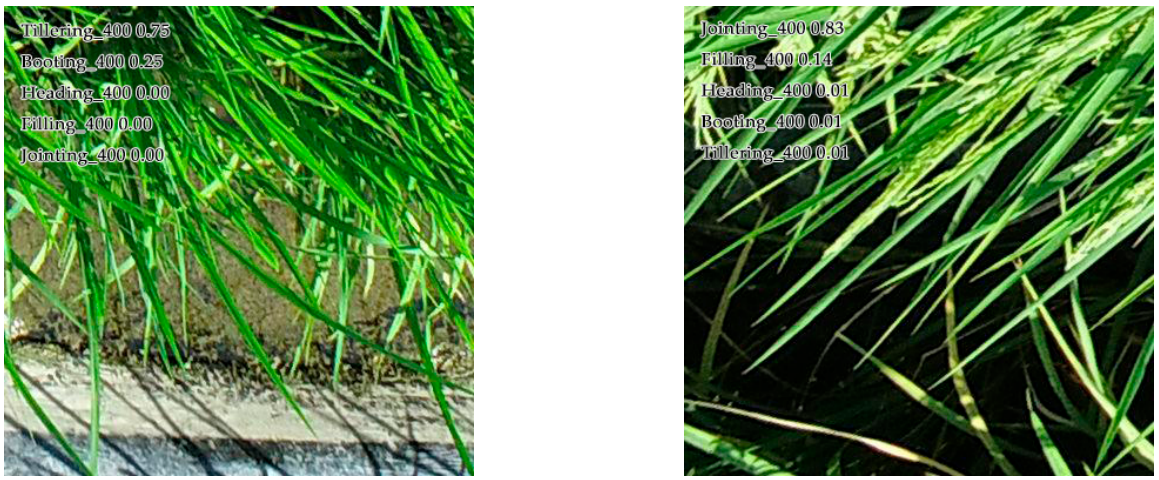


Figure 12. Confusion matrix for Mobilenetv3-YOLOv8 evaluated on the test dataset.



(**a**) (**b**)

Figure 13. False-positive detection with (**a**) booting stage being falsely recognized as tillering stage and (**b**) filling stage being falsely recognized as jointing stage.

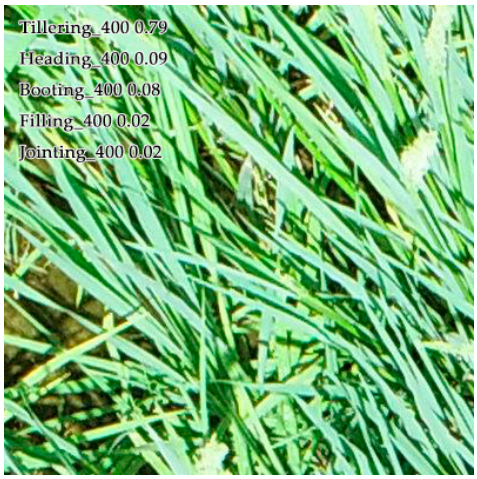


Figure 14. False-positive detection with booting stage being falsely recognized as tillering stage.

In summary, the experimental results demonstrated that the optimal Mobilenetv3-YOLOv8 model shows excellent performance and has potential for deployment in edge computing devices and practical applications in in-field rice growth-stage recognition. Figures 10 and 12 reveal that the parameters and GFLOPs of the optimal Mobilenetv3-YOLOv8 were reduced to 6.60M and 0.9, respectively, with precision values for tillering, jointing, booting, heading, and filling stages of 94.88%, 93.36%, 67.85%, 78.31%, and 85.46%, respectively. In the future, more advanced deep learning models and different types of field crops need to be adopted to further validate and improve the generality and robustness of our algorithm.

4.2. Comparison with the Existing Literature

Deep learning-based detection of rice growth stages surpasses traditional machine learning techniques that rely on manual design and feature extraction. Table 7 summarizes recent applications of deep learning in rice growth-stage detection, with the values of the model proposed in this study highlighted in bold. Kaixuan Liu et al. [1] developed Small-YOLOv5 to automatically identify rice growth stages. Although Small-YOLOv5 achieved excellent accuracy, the Mobilenetv3-YOLOv8 model offers a more extensive dataset and faster GFLOPs, making it more suitable for widespread agricultural applications. Qi Yang et al. [8] improved convolutional neural networks (CNNs), reaching an accuracy of 83.9%; however, the simpler dataset limits its adaptability for practical use. Jiale Qin et al. [9] utilized YOLO for dataset filtering and ResNet-50 to automatically recognize multiple developmental stages of rice panicles, but the large parameter size increases hardware demands for agricultural equipment. Therefore, without compromising accuracy, Mobilenetv3-YOLOv8 shows greater potential for deployment in practical devices.

Table 7. Comparison with the existing literature.

Algorithms	Species	Scale of Dataset	mAP/%	Parameter/M	GFLOPs	Reference
Small-YOLOv5	5	4844	94.20	1.24	2.3	[1]
Improved CNN	4	3160	83.9	-	-	[8]
YOLO+ResNet50	6	25,385	87.33	23.5	-	[9]
Mobilenetv3-YOLOv8	**5**	**147,492**	**84.00**	**6.60**	**0.9**	**This study**

4.3. Limitations and Future Work

This study still has certain limitations and areas for improvement.

(1) In the construction of the dataset, the impact of climatic factors during shooting was not considered. This includes issues such as insufficient lighting leading to darker

colors in the input images, high wind speeds causing drone instability and resulting in blurry images, and the effects of different geographical regions on rice growth. In the future, the dataset could be expanded to include images of rice growth stages under more diverse geographical regions and climatic conditions, aiming to counteract the negative feedback caused by environmental factors.

(2) The experimental results show that model misidentifications primarily occur between two adjacent periods, possibly due to similar phenotypic traits. In addition, only RGB images were used in this study. In the future, more feature descriptors captured by drones equipped with different sensors, such as spectral and infrared sensors and LiDAR, can be introduced and fused with RGB information to reduce such confusion.

(3) Although Mobilenetv3-YOLOv8 reduces computational load, its deployment in edge computing devices still needs to consider the limitation of computing resources. Therefore, further optimizing the model to maintain better performance under lower computing resources is a necessary direction.

Moreover, the development space of this study is very broad: the model can be combined with drone and real-time data processing technologies to develop a system for real-time monitoring of rice growth stages, thereby enhancing agricultural management efficiency. On the basis of this study, it is also possible to achieve cross-crop growth-stage identification, extending the model approach to the identification of growth stages in other crops and exploring its applicability to different crops.

5. Conclusions

Traditional manual rice growth-stage recognition is characterized by low efficiency and subjectivity, significantly hindering the development of scientific and reasonable field management strategies. The current challenges motivate the development of more precise and efficient methods for detecting crop growth stages in large-scale fields. Therefore, this study aimed to propose a lightweight and improved YOLOv8 to identify five key growth stages of rice. The main contributions are summarized in the following points:

(1) **Dataset Construction**. Rice images of five growth stages were obtained from two comprehensive field plot experiments, namely, EXP.1 and EXP.2. The original UAV images were cropped into sub-images of equal sizes, and image prepossessing was performed to exclude images containing background information. Finally, images collected in EXP.1 were established as training and validation sets, while images collected in EXP.2 were used as an independent test dataset. In addition, we established six datasets with different image sizes and completed the anchor box work in advance in the form of classification discussions of the dataset. This allowed the model to skip the anchor box confirmation part after inputting the images, which helped to accelerate the model's running speed while also lightweighting the model.

(2) **Construction of Mobilenetv3-YOLOv8, the Novel Network**. First, we replaced the backbone network of YOLOv8n with Mobilenetv3, which minimizes parameter size without sacrificing accuracy. Second, to enhance the features of rice of different growth stages while suppressing irrelevant features, the CA was introduced and compared with the CBAM. In addition, the attention mechanisms were integrated into different layers of the backbone network.

(3) **Validation Experiments.** Comparative experiments were conducted to evaluate the performance of the improved YOLOv8 in the recognition of five key rice growth stages, and other factors that affect the performance of Mobilenetv3-YOLOv8 were also evaluated. First, the performance of YOLOv8 with different image-size inputs was compared, and then ablation experiments were performed to analyze the individual impact of the different version of the Mobilenet network and the CA and CBAM. Afterward, the performance of the Mobilenetv3-YOLOv8, which integrated attention mechanisms into different position layers, was evaluated. Last, the impact of image collection at two flight altitudes on model performance was also considered.

(4) **New Results.** With a comparative performance evaluation of the improved YOLOv8 model, Mobilenetv3-YOLOv8 demonstrated a significant improvement in rice growth-stage recognition and a reduction in the computation of the network. The best performance of Mobilenetv3-YOLOv8 in the recognition of five rice growth stages was found with mAP and recall values of 84.00% and 84.08%, respectively, when the input image size was set to 400 × 400 pixels, Mobilenetv3 was selected as the backbone network, and the CA was integrated in the second and the third layers of the backbone network. The parameters and GFLOPs of the optimal Mobilenetv3-YOLOv8 were reduced to 6.60M and 0.9, respectively, with precision values for tillering, jointing, booting, heading, and filling stages of 94.88%, 93.36%, 67.85%, 78.31%, and 85.46%, respectively. The experimental results revealed that the optimal Mobilenetv3-YOLOv8 model shows excellent performance and has potential for deployment in edge computing devices and practical applications in in-field rice growth-stage recognition.

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

Funding: This work was supported by the Guangzhou Science and Technology Project (No. 2024B03J1310), the Innovation Projects of the Ordinary University in Guangdong Province (No. 2024KTSCX099), and the College Students' Innovative Entrepreneurial Training Project (No. 202310564007).

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

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

References

1. Liu, K.; Wang, J.; Zhang, K.; Chen, M.; Zhao, H.; Liao, J. A Lightweight Recognition Method for Rice Growth Period Based on Improved YOLOv5s. *Sensors* **2023**, *23*, 6738. [CrossRef]
2. Moldenhauer, K.; Slaton, N. Rice Growth and Development. In *Rice Production Handbook*; University of Arkansas: Fayetteville, AR, USA, 2001; Volume 192, pp. 7–14.
3. Bai, X.; Cao, Z.; Zhao, L.; Zhang, J.; Lv, C.; Li, C.; Xie, J. Rice Heading Stage Automatic Observation by Multi-Classifier Cascade Based Rice Spike Detection Method. *Agric. For. Meteorol.* **2018**, *259*, 260–270. [CrossRef]
4. Sheng, R.T.-C.; Huang, Y.-H.; Chan, P.-C.; Bhat, S.A.; Wu, Y.-C.; Huang, N.-F. Rice Growth Stage Classification via RF-Based Machine Learning and Image Processing. *Agriculture* **2022**, *12*, 2137. [CrossRef]
5. He, Z.; Li, S.; Wang, Y.; Dai, L.; Lin, S. Monitoring Rice Phenology Based on Backscattering Characteristics of Multi-Temporal RADARSAT-2 Datasets. *Remote Sens.* **2018**, *10*, 340. [CrossRef]
6. Shao, Y.; Fan, X.; Liu, H.; Xiao, J.; Ross, S.; Brisco, B.; Brown, R.; Staples, G. Rice Monitoring and Production Estimation Using Multitemporal RADARSAT. *Remote Sens. Environ.* **2001**, *76*, 310–325. [CrossRef]
7. Ma, X.; Wu, Y.; Shen, J.; Duan, L.; Liu, Y. ML-LME: A Plant Growth Situation Analysis Model Using the Hierarchical Effect of Fractal Dimension. *Mathematics* **2021**, *9*, 1322. [CrossRef]
8. Yang, Q.; Shi, L.; Han, J.; Yu, J.; Huang, K. A near Real-Time Deep Learning Approach for Detecting Rice Phenology Based on UAV Images. *Agric. For. Meteorol.* **2020**, *287*, 107938. [CrossRef]
9. Qin, J.; Hu, T.; Yuan, J.; Liu, Q.; Wang, W.; Liu, J.; Guo, L.; Song, G. Deep-Learning-Based Rice Phenological Stage Recognition. *Remote Sens.* **2023**, *15*, 2891. [CrossRef]
10. Rasti, S.; Bleakley, C.J.; Silvestre, G.C.M.; O'Hare, G.M.P.; Langton, D. Assessment of Deep Learning Methods for Classification of Cereal Crop Growth Stage Pre and Post Canopy Closure. *J. Electron. Imaging* **2023**, *32*, 033014-1–033014-21. [CrossRef]
11. Zhang, Y.; Xiao, D.; Liu, Y.; Wu, H. An Algorithm for Automatic Identification of Multiple Developmental Stages of Rice Spikes Based on Improved Faster R-CNN. *Crop J.* **2022**, *10*, 1323–1333. [CrossRef]
12. von Bueren, S.K.; Burkart, A.; Hueni, A.; Rascher, U.; Tuohy, M.P.; Yule, I.J. Deploying Four Optical UAV-Based Sensors over Grassland: Challenges and Limitations. *Biogeosciences* **2015**, *12*, 163–175. [CrossRef]
13. Zhang, Y.; Jiang, Y.; Xu, B.; Yang, G.; Feng, H.; Yang, X.; Yang, H.; Liu, C.; Cheng, Z.; Feng, Z. Study on the Estimation of Leaf Area Index in Rice Based on UAV RGB and Multispectral Data. *Remote Sens.* **2024**, *16*, 3049. [CrossRef]

14. Meyer, G.E.; Neto, J.C. Verification of Color Vegetation Indices for Automated Crop Imaging Applications. *Comput. Electron. Agric.* **2008**, *63*, 282–293. [CrossRef]
15. Hu, J.; Shi, C.J.R.; Zhang, J. Saliency-Based YOLO for Single Target Detection. *Knowl. Inf. Syst.* **2021**, *63*, 717–732. [CrossRef]
16. Folarin, A.; Munin-Doce, A.; Ferreno-Gonzalez, S.; Ciriano-Palacios, J.M.; Diaz-Casas, V. Real Time Vessel Detection Model Using Deep Learning Algorithms for Controlling a Barrier System. *J. Mar. Sci. Eng.* **2024**, *12*, 1363. [CrossRef]
17. Samma, H.; Al-Azani, S.; Luqman, H.; Alfarraj, M. Contrastive-Based YOLOv7 for Personal Protective Equipment Detection. *Neural Comput. Appl.* **2024**, *36*, 2445–2457. [CrossRef]
18. Li, C.; Chen, C.; Hei, Y.; Mou, J.; Li, W. An Efficient Advanced-YOLOv8 Framework for THz Object Detection. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 5019211. [CrossRef]
19. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics* **2023**, *12*, 2323. [CrossRef]
20. Hu, D.; Yu, M.; Wu, X.; Hu, J.; Sheng, Y.; Jiang, Y.; Huang, C.; Zheng, Y. DGW-YOLOv8: A Small Insulator Target Detection Algorithm Based on Deformable Attention Backbone and WIoU Loss Function. *IET Image Process.* **2024**, *18*, 1096–1108. [CrossRef]
21. Yue, M.; Zhang, L.; Huang, J.; Zhang, H. Lightweight and Efficient Tiny-Object Detection Based on Improved YOLOv8n for UAV Aerial Images. *Drones* **2024**, *8*, 276. [CrossRef]
22. Huang, L.; Xiang, Z.; Yun, J.; Sun, Y.; Liu, Y.; Jiang, D.; Ma, H.; Yu, H. Target Detection Based on Two-Stream Convolution Neural Network with Self-Powered Sensors Information. *IEEE Sens. J.* **2023**, *23*, 20681–20690. [CrossRef]
23. Wang, S.; Wang, Y.; Zheng, B.; Cheng, J.; Su, Y.; Dai, Y. Intrusion Detection System for Vehicular Networks Based on MobileNetV3. *IEEE Access* **2024**, *12*, 106285–106302. [CrossRef]
24. Jang, J.-G.; Quan, C.; Lee, H.D.; Kang, U. Falcon: Lightweight and Accurate Convolution Based on Depthwise Separable Convolution. *Knowl. Inf. Syst.* **2023**, *65*, 2225–2249. [CrossRef]
25. Quan, L.; Feng, H.; Li, Y.; Wang, Q.; Zhang, C.; Liu, J.; Yuan, Z. Maize Seedling Detection under Different Growth Stages and Complex Field Environments Based on an Improved Faster R-CNN. *Biosyst. Eng.* **2019**, *184*, 1–23. [CrossRef]
26. Zeng, W.; He, M. Rice Disease Segmentation Method Based on CBAM-CARAFE-DeepLabv3+. *Crop Prot.* **2024**, *180*, 106665. [CrossRef]
27. Zhang, P.; Li, D. CBAM plus ASFF-YOLOXs: An Improved YOLOXs for Guiding Agronomic Operation Based on the Identification of Key Growth Stages of Lettuce. *Comput. Electron. Agric.* **2022**, *203*, 107491. [CrossRef]
28. Jia, L.; Wang, T.; Chen, Y.; Zang, Y.; Li, X.; Shi, H.; Gao, L. MobileNet-CA-YOLO: An Improved YOLOv7 Based on the MobileNetV3 and Attention Mechanism for Rice Pests and Diseases Detection. *Agriculture* **2023**, *13*, 1285. [CrossRef]
29. Xie, C.; Zhu, H.; Fei, Y. Deep Coordinate Attention Network for Single Image Super-Resolution. *IET Image Process.* **2022**, *16*, 273–284. [CrossRef]
30. Xu, L.; Zhang, H.; Wang, C.; Wei, S.; Zhang, B.; Wu, F.; Tang, Y. Paddy Rice Mapping in Thailand Using Time-Series Sentinel-1 Data and Deep Learning Model. *Remote Sens.* **2021**, *13*, 3994. [CrossRef]
31. Nan, Y.; Zhang, H.; Zeng, Y.; Zheng, J.; Ge, Y. Faster and Accurate Green Pepper Detection Using NSGA-II-Based Pruned YOLOv5l in the Field Environment. *Comput. Electron. Agric.* **2023**, *205*, 107563. [CrossRef]
32. Quan, L.; Xu, L.; Li, L.; Wang, H.; Huang, X. Solar Active Region Detection Using Deep Learning. *Electronics* **2021**, *10*, 2284. [CrossRef]