

Article

Automatic Identification of Sea Rice Grains in Complex Field Environment Based on Deep Learning

Ruoling Deng^{1,2}, Weilin Cheng¹, Haitao Liu^{1,2} , Donglin Hou¹, Xiecheng Zhong¹, Zijian Huang¹, Bingfeng Xie¹ and Ningxia Yin^{1,2,*}

¹ School of Mechanical Engineering, Guangdong Ocean University, Zhanjiang 524088, China; dengruoling@gdou.edu.cn (R.D.); chengweilin@stu.gdou.edu.cn (W.C.); gdliuht@126.com (H.L.); 202111413211@stu.gdou.edu.cn (D.H.); shagenggeng1@stu.gdou.edu.cn (X.Z.); 202111413214@stu.gdou.edu.cn (Z.H.); 2021114131291@stu.gdou.edu.cn (B.X.)

² Guangdong Engineering Technology Research Center of Ocean Equipment and Manufacturing, Zhanjiang 524088, China

* Correspondence: yinnx@gdou.edu.cn; Tel.: +13-828-229607

Abstract: The number of grains per sea rice panicle is an important parameter directly related to rice yield, and it is also a very important agronomic trait in research related to sea rice breeding. However, the grain number per sea rice panicle still mainly relies on manual calculation, which has the disadvantages of being time-consuming, error-prone, and labor-intensive. In this study, a novel method was developed for the automatic calculation of the grain number per rice panicle based on a deep convolutional neural network. Firstly, some sea rice panicle images were collected in complex field environment and annotated to establish the sea rice panicle image data set. Then, a sea grain detection model was developed using the Faster R-CNN embedded with a feature pyramid network (FPN) for grain identification and location. Also, ROI Align was used to replace ROI pooling to solve the problem of relatively large deviations in the prediction frame when the model detected small grains. Finally, the mAP (mean Average Precision) and accuracy of the sea grain detection model were 90.1% and 94.9%, demonstrating that the proposed method had high accuracy in identifying and locating sea grains. The sea rice grain detection model can quickly and accurately predict the number of grains per panicle, providing an effective, convenient, and low-cost tool for yield evaluation, crop breeding, and genetic research. It also has great potential in assisting phenotypic research.

Keywords: grain number; rice whole panicle; grain detection; convolutional neural network; plant phenotyping



Citation: Deng, R.; Cheng, W.; Liu, H.; Hou, D.; Zhong, X.; Huang, Z.; Xie, B.; Yin, N. Automatic Identification of Sea Rice Grains in Complex Field Environment Based on Deep Learning. *Agriculture* **2024**, *14*, 1135. <https://doi.org/10.3390/agriculture14071135>

Academic Editors: Panagiotis Tziachris, Christos Karydas and Miltiadis Iatrou

Received: 12 June 2024
Revised: 9 July 2024
Accepted: 10 July 2024
Published: 12 July 2024



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

1. Introduction

Sea rice is commonly known as salt–alkali-tolerant rice, which can grow in tidal flats and saline–alkali land [1,2]. Due to the impact of human activities, land salinization is becoming more and more serious [3]. Among the 230 million hectares of irrigated land in the world, 20% has been affected by salinization, which would greatly affect rice cultivation [4]. Currently, the yield of sea rice is still low, so research about how to increase sea rice yield is receiving more and more attention [5]. Simultaneously, the yield assessment of sea rice has become increasingly important. Among them, the number of grains per panicle of sea rice is a trait directly related to yield, which often needs to be measured in the process of yield evaluation and variety breeding [6]. However, current counting of sea rice panicle grains mainly relies on manual labor, and it is extremely time-consuming for researchers to conduct large-scale field measurements. Therefore, it is of great significance to study a method that can automatically identify grains on the sea rice panicle.

In recent years, many researchers have applied image processing technology to study particle counting methods. For example, visible light and X-ray imaging technologies were used to count grains on the panicle [7]. Also, a rice panicle phenotyping system

that combined X-ray and RGB scanning to comprehensively evaluate spikelet and grain traits was developed [8]. A high-throughput rice phenotyping facility (HRPF), which used multi-angle color images combined with neural network algorithms, was developed to measure the number of grains per panicle in rice plants [9]. Wu et al. used image processing and deep learning algorithms to count grains on the rice panicles and solved the problem of overlapping and dense grains on rice panicles [5]. Li et al. proposed a low-cost method for calculating rapeseed inflorescences using YOLOv5 and a convolutional block attention module (CBAM) based on unmanned aerial vehicle (UAV) RGB images [10]. The above research methods have greatly advanced the detection level of small targets such as rice panicle grains and achieved higher accuracy. However, the rice panicle images of these methods are all taken under conditions with stable light sources and simple backgrounds, and they are not suitable for complex field environments.

Liu et al. aimed at the problem of the detection accuracy of small and medium-sized objects in an SSD (Single Shot Multibox Detector) and proposed to introduce a deconvolution area amplification program and construct a new feature pyramid by extracting features in shallow layers to achieve small-target detection [11]. Wang et al. proposed a multi-scale residual aggregation feature pyramid network called MSRA-FPN, which aggregated features from multiple levels to the top layer through a unidirectional cross-layer residual module to enhance the semantic information of high-level feature maps [12]. The information attenuation during the feature fusion process was alleviated, thereby achieving better target detection performance. Wang et al. proposed a feature pyramid network called an IFPN (Interconnected Feature Pyramid Network), which used an attention mechanism to simultaneously select attention features and had significant improvements in feature enhancement [13]. Ren et al. introduced a region proposal network (RPN) to achieve almost free region proposals, which shared full-image convolutional features with the detection network and could predict object boundaries and object scores at each location, ultimately achieving target detection [14]. The aforementioned methods can well extract the characteristics of small targets, improve the accuracy of small-target detection in target detection tasks, and provide a reliable theoretical basis for research on the detection of small targets such as wheat ears and grains.

With the development of computer science, some researchers have developed methods for the intelligent counting of grains based on the crop plant structure. For example, Saeed Khaki et al. proposed a sliding window-based counting method that could detect and count corn ears under different lighting conditions [15]. Wang et al. proposed a new feature pyramid network called an Adaptive Feature Pyramid Network (AFP), which adopted the design of adaptive feature upsampling and adaptive feature fusion to alleviate the problems caused by medium-scale changes in target detection [16]. Wei et al. trained a wheat grain detection and counting model, which can be used for wheat grain detection and counting at multiple scales and angles in complex backgrounds [17]. Gong et al. proposed an improved Faster R-CNN algorithm to solve problems such as object occlusion, deformation, and small size in object detection [18]. Dandriofosse et al. collected RGB images of wheat from heading to maturity in complex field environments and developed a wheat ear counting and segmentation method [19]. Wang et al. also calculated the number of ears in rice images with different lighting conditions, different backgrounds, and different input sizes in complex field environments and achieved good robustness and accuracy [20]. Also, a novel prototype, dubbed “GN-System”, was developed for the automatic calculation of the grain number per rice panicle based on deep convolutional neural network and panicle structure [21]. The above method obtains more accurate target information by obtaining higher-scale image feature information, and it has strong adaptability when performed under different backgrounds. However, its experimental environment is mainly indoors, and its background complexity is relatively simple. It will reduce the detection robustness of the model. The above research on detection in a field environment requires the artificial addition of other backgrounds for occlusion. Although this method is effective at improving detection robustness, it is very inconvenient.

In summary, the existing methods for detecting rice panicle grains are all performed in indoor environments, using a baffle that can highlight the characteristics of the rice panicles as a background to improve the detection accuracy. Also, in the actual counting of the grain number per rice panicle, the rice panicles need to be sent to a specific environment after harvesting, and then they must be manually unfolded and placed flat in a fixed position for photographing and counting. Alternatively, the rice panicles are threshed and then counted grain by grain. These methods not only increase the workload of researchers but may also cause damage to rice grain. Furthermore, there are few studies on how rice or saline–alkali-tolerant rice can detect and count grains in the complex field environment. Complex environments and unstable light sources will increase the difficulty of grain identification. Simultaneously, the grains of saline–alkali-tolerant rice panicles are denser than those of ordinary rice panicles, and overlapping rice panicle grains also make research more difficult. Therefore, it is meaningful to study a method to directly count grains per sea rice panicle in a field environment. Faster R-CNN, the FPN, and ROI Align have advantages in small-target detection, such as high accuracy, strong multi-scale feature fusion capability, and precise feature extraction. These advantages make the scheme have great application potential and value in the field of small-grain detection. Given the above reasons, this paper proposed an improved Faster RCNN algorithm to detect and count the number of grains per sea rice panicle.

The main objectives of this research were to (a) collect the panicle images of sea rice under complex field environment, (b) establish the grain detection and counting model using a convolutional neural network, and (c) evaluate the detection stability and accuracy of the proposed method.

2. Materials and Methods

The complete panicle images of saline–alkali-tolerant rice under different light conditions were manually collected in the complex field environment, followed by manual labeling. Then, the images were divided via a random method into three sub-data sets: a training set, verification set, and test set. Also, data enhancement operations such as Gaussian blur [22] and flip transformation were performed on the data set to enhance the generalization ability of the model. The transfer learning method was adopted in training the model, which significantly reduces the training time and quickly lowers the loss value.

2.1. Data Set Preparation

2.1.1. Image Acquisition

The panicles of sea rice in the filling stage were blue and white, like reeds. The grains of sea rice had prickles, and the rice kernel inside was red. The height of sea rice was above 1.8–2.3 m, and the root depth was 30–40 cm, while the height of common rice was only 1.2–1.3 m. The rice panicle is an important reproductive organ of rice, and the growth of the rice reproductive organ will directly affect the development of the rice panicle. Therefore, the shape, size, color, texture, and posture of rice panicle region are closely related to the final yield of rice.

The sea water rice panicle images were directly collected in rice fields at the Agricultural Research Institute on Huguang Campus, Guangdong Ocean University, Zhanjiang City, Guangdong Province, China (110.3342° N, 21.26333° E). The advantage of directly collecting panicle images in rice fields is that the rice plants will not be damaged, which is vital for rice research during the growing stage. The collection site in the sea rice panicle image is shown in Figure 1a. The sea rice panicle images were collected from the paddy field during the rice's ripening period in two steps (Figure 1b). Firstly, the ends of sea rice panicles were lifted manually, and the panicle branches were gently spread apart if the grains on the rice panicles were highly dense. This step was to expose the grains on the sea rice panicle as much as possible. Then, the RGB images of sea rice panicles were taken 30 cm parallel to the panicles (the side where more panicle grains can be seen) from the

horizontal direction using a mobile phone. Thus, as many grains as possible of sea rice panicles were captured in the image (Figure 1c).

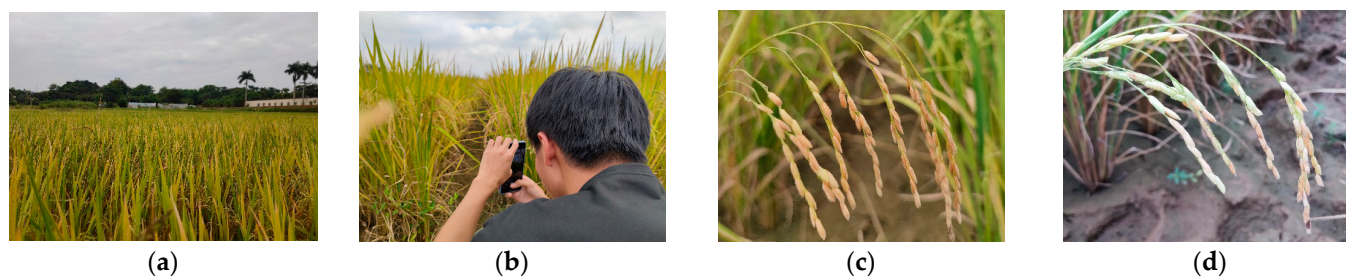


Figure 1. Illustration of sea rice panicle: (a) collection site for sea rice panicle images; (b) sea rice panicle being photographed with a mobile phone; (c) panicle image in the middle part of the rice field; (d) panicle image at the edge of the rice field.

The rice panicle sample of sea rice selected in this paper was Haihong 12 cultivated by Guangdong Ocean University. Its length was moderate, with an average panicle length of 23 cm, which was widely popular and typical. The row spacing of sea rice at the sampling points in the rice fields was 20 cm. The data collection time was 12 November 2022, and the weather was sunny with about 10% cloud. At this stage, the sea rice panicles were fully formed, the grain size was full, and the rice panicles had basically reached a mature state. Moreover, the color of the rice panicles was significantly different from the color of rice stems and leaves. Choosing to collect sea rice panicle images at this stage would help the model to fully learn characteristics such as grain color and grain surface texture, which could further improve the grain detection accuracy of the model. The image collection tools were four types of mobile phones. The specifications of the mobile phones are listed in Table 1. The original images were saved in JPG format, and a total of 200 images were taken. Among them, the number of images taken using different mobile phones was 41, 94, 45, and 20 for Xiaomi Mi 11, Redmi K40, Apple iPhone 12, and Apple iPhone 11, respectively.

Table 1. Specifications of mobile phones used in photographing rice panicles.

Phone Model	Brand	Country	Camera Resolution	Focal Length Range
Xiaomi Mi 11	Xiaomi Inc. (Beijing, China)	China	3200 × 1440 pixels	26–50 mm
Redmi K40	Xiaomi Inc. (Beijing, China)	China	2400 × 1080 pixels	25–50 mm
Apple iPhone 12	Apple Inc. (Cupertino, CA, USA)	USA	2532 × 1170 pixels	13–26 mm
Apple iPhone 11	Apple Inc. (Cupertino, CA, USA)	USA	1792 × 828 pixels	13–26 mm

The advantages of using smartphones to collect sample data were small size, portability, easy operation, easy post-processing, etc., and the focusing function could improve image clarity and ensure a high resolution. To avoid experimental errors caused by edge effects (Figure 1d), images of rice panicles at the edge of rice field, whose background was usually soil, were also collected. It can ensure the diversity of rice panicle grain data and help to enhance the identification stability of the grain model.

2.1.2. Image Annotation

The collected data were manually screened to remove blurry and out-of-focus images. The visual image annotation tool LabelImg (version 1.8.1, <https://github.com/tzutalin/labelImg> (accessed on 10 January 2024)) was used to annotate sea rice panicle images (Figure 2). LabelImg supported label output in PASCAL VOC, YOLO, and COCO formats. The label file output by PASCAL VOC in XML format [23] was used. During the image acquisition process, some rice panicles may have been blurred due to being out of focus or obscured by other rice panicles. Therefore, when the area of the panicle grains was blocked

by more than 90% or the panicle grain boundaries were missing due to blur, no labeling was performed.

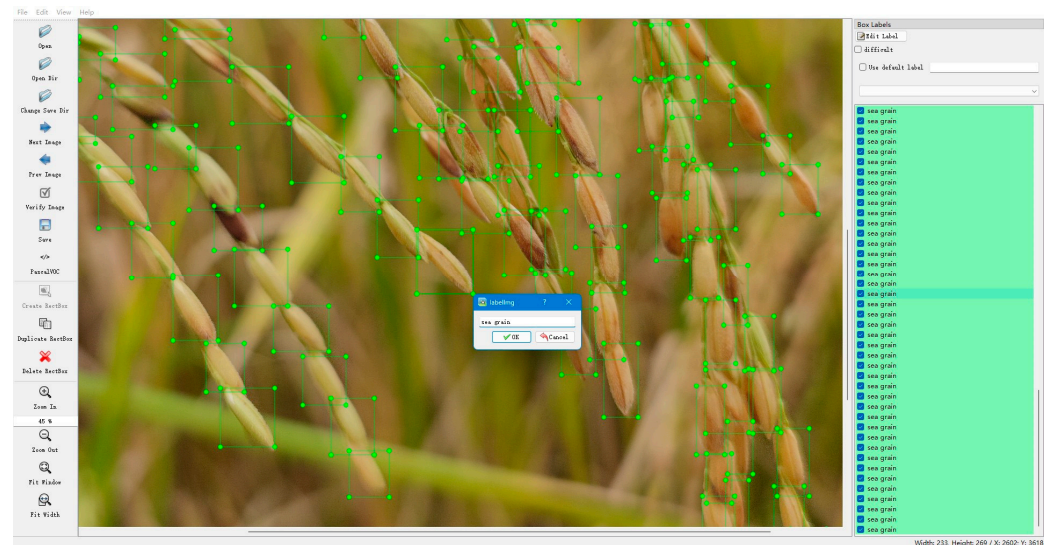


Figure 2. Annotation of sea rice panicle image, where the green squares represent the coordinate positions of sea rice grains in panicle image.

2.1.3. Image Processing

In the training process of our deep learning models, data augmentation was an essential link. The existing deep learning models had many parameters. The number of parameters that a general model could train was tens of thousands to millions. Still, the number of samples in a general training set usually struggled to reach tens of thousands to millions. Moreover, when deep learning was applied to actual engineering projects and finally implemented, many problems would be encountered, such as issues with lighting, occlusion, shadows, etc. The original data collected usually struggled to meet various real environments. In this case, data augmentation was needed to increase the robustness and generalization of the deep learning model. Commonly used data augmentation libraries included torchvision [24], imgaug [25], and albumentations [26]. Since the imgaug third-party library was not only easy to use but also highly customizable and extensible, the imgaug library was ultimately used in this paper. Also, compared with the torchvision official data augmentation library, the imgaug library provided more diverse data augmentation methods and had a faster computing speed.

Compared with the algorithms' third-party data augmentation library, the imgaug library can very conveniently combine multiple methods. For example, the usage ratios of different methods can be used to perform data augmentation on the original images, which can truly improve the generalization ability of the deep model during the subsequent training process. One of the most important reasons for choosing the imgaug library was that it can perform corresponding transformations on keypoints and bounding boxes while augmenting data.

After annotating the original images, the BoundingBoxesOnImage function in imgaug was used to transform the labels and coordinates accordingly. Finally, a data set with 800 images was obtained. This step greatly reduced the time spent labeling the data set. Through statistical analysis, it was known that the total number of sea rice grain labels reached 89,125, which was enough for model training. Figure 3 shows a picture of some of the data after augmentation. After a series of data augmentation steps, each image in the data set corresponded to an XML format annotation file, which contained the label and coordinates of each rice grain. The final data set was divided into a training set, verification set, and test set according to a ratio of 8:1:1 by a random method.

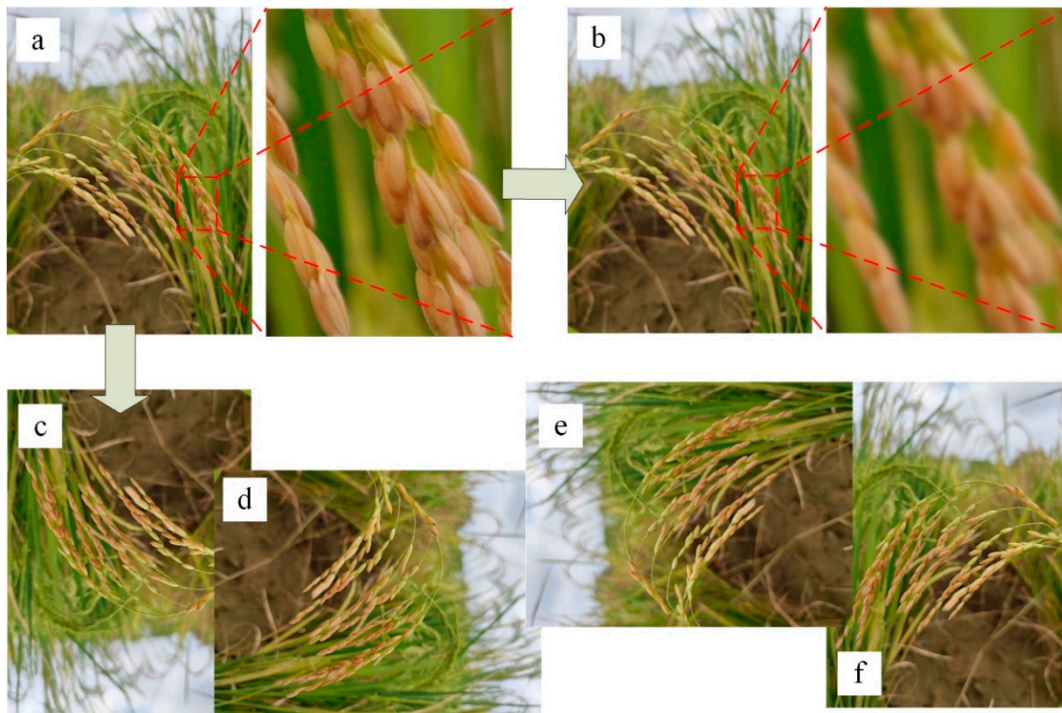


Figure 3. Illustration of panicle images after augmentation: (a) original image; (b) Gaussian blur; (c) 180° flip; (d,e) 90° flip; (f) mirror flip.

2.2. Grain Detection Model

2.2.1. Construction of the Sea Grain Detection Model

The Faster R-CNN model was selected to train the data [14]. It mainly consisted of the RPN (region proposal network) and Fast R-CNN (Figure 4) [27]. Because RPN and Fast R-CNN shared the same backbone network, choosing these models greatly increased the inference speed. There were two reasons for choosing the Faster R-CNN model. First, it was a two-stage target detection network that was more accurate than a one-stage network and could better solve the problem of small targets. Secondly, Faster R-CNN had strong versatility and robustness due to its ability to handle multi-scale and multi-target problems, was easy to use to carry out transfer learning, and could be better applied to different varieties of hybrid rice.

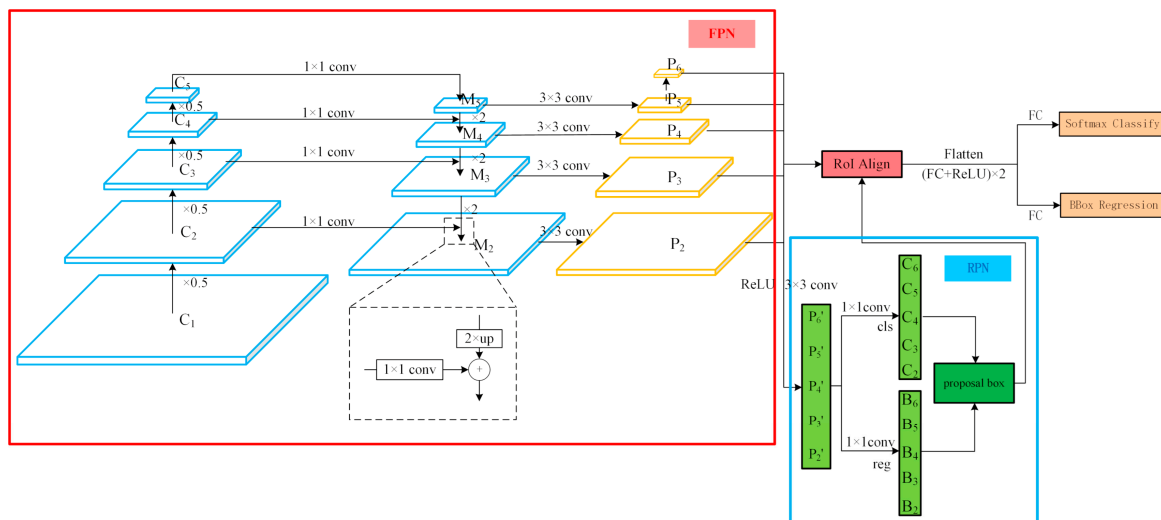


Figure 4. The architecture of the sea grain detection model.

Since the feature maps extracted by Faster R-CNN were all low-level, it was difficult to obtain the semantic information of high-level features, but the bottom-level information could not be lost to obtain accurate target positions. Therefore, the pyramid feature network (FPN) [28] was integrated on this basis. The FPN enabled the features of different sizes to contain rich semantic information, and the computational cost remains manageable. Its structure consisted of three processes: bottom-up, top-down, and lateral connection. It was suitable for multi-scale and small-target detection. Also, the performance was even better.

The backbone network extracted features from the original image through the ResNet101 network (Figure 5) [29] since, in small-target detection, higher-level feature information was needed. Deepening the network layer could obtain richer feature information, and ResNet solved the problem of gradient explosion that occurred as the number of network layers increased. In this process, the layers with constant feature map size were classified into one stage, and the output of each layer was defined as C1, C2, C3, C4, or C5. The output sizes were 2, 4, 6, 8, and 32 times that of the original image. That is to say, the size of the previous layer was twice that of the next layer. This process was the bottom-up stage of the FPN, in which the model network could obtain features of different scales.

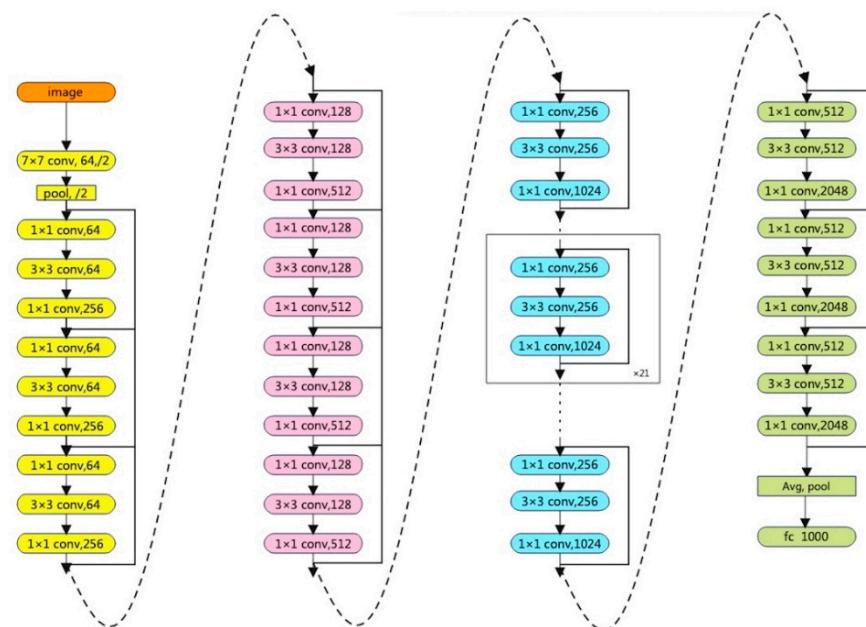


Figure 5. The structure of the Resnet 101 network.

The top-down process upsampled the high-level feature information obtained by the bottom-up process and then passed it downward. High-level features contained rich semantic information, and this feature information could be spread to low-level features through top-down propagation. The upsampling method was nearest-neighbor interpolation, also known as zero-order interpolation. It made equal the transformed gray value to the gray value of the nearest input pixel. This algorithm could upsample the image very quickly. Its job was to double the size of adjacent high-level feature maps and pass them to the next layer. In this process, a lateral connection needed to be performed. In this study, each layer was defined as M5, M4, M3, or M2. Therefore, except for the C1 layer, the feature maps generated by bottom-up convolution would correspond to the feature maps generated by top-down upsampling.

In the process of propagating each top-down layer to the next layer, it would be integrated with the feature map output by each stage in the corresponding backbone network. This process was the lateral connection of the FPN (Figure 6). The feature map C_n output in the backbone network first underwent a 1×1 convolution operation to reduce the dimension, and then it was fused with the feature map M_{n+1} passed by downsampling, which was directly added, and the corresponding M_n layer was obtained. Since in the

bottom-up and top-down processes, the same ratio had been already used to adjust the size of the feature map, the corresponding elements were added directly.

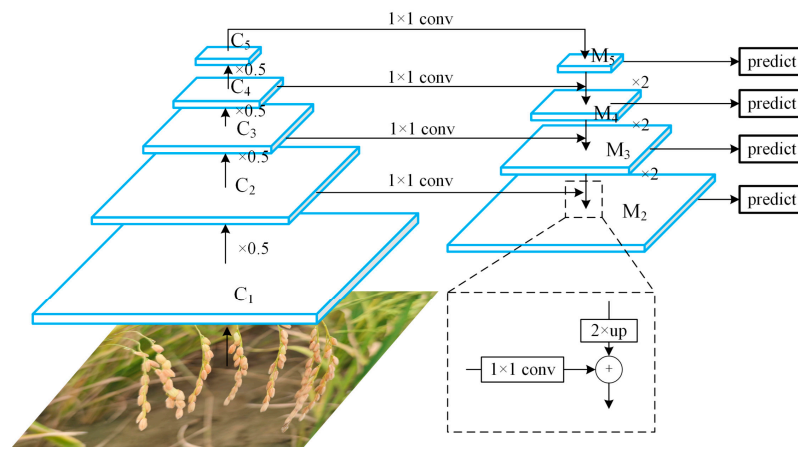


Figure 6. The structure of the FPN.

Finally, a 3×3 convolution on the feature map M_n of each layer was performed to obtain P_n as the output result of this layer. The purpose of 3×3 convolution was to eliminate the aliasing effect caused by upsampling to re-extract features and ensure the stability of their features. The number of output P_n channels of each layer obtained through the above operations was 256. Finally, downsampling with a step size of 2 was performed on P_5 to obtain P_6 , which was introduced to cover a larger Anchor Scale, with a size of 512×512 .

RPN was a fully convolutional network that simultaneously predicted object boundaries and objectness scores for each location (Figure 7). Firstly, the RPN used a 3×3 filter to convolve on the feature map P_n generated by the FPN, which would make the extracted features more robust. Also, the feature map was mapped into multiple proposal boxes (reg layer branch) and semantic information as classification in the bounding box (cls layer branch). Finally, the positive anchors and the corresponding bounding box regression were combined to obtain proposal boxes. Also, the proposal boxes that were too small and exceeded the boundary were eliminated.

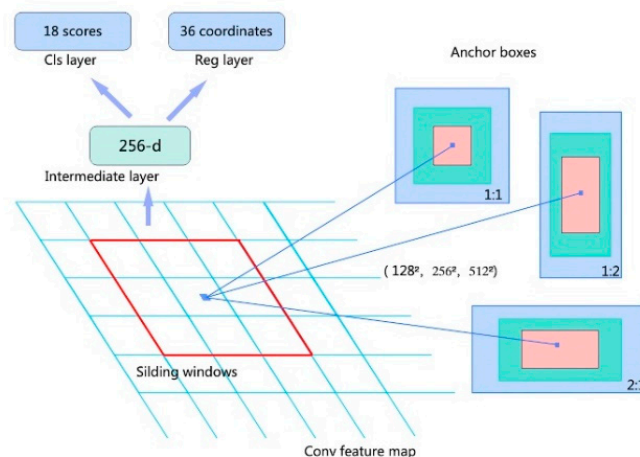


Figure 7. The region proposal network (RPN).

In the process of region of interest (RoI) extraction by Faster R-CNN, directly using RoI Pooling to anchor the bounding box of the target would cause a certain deviation between the candidate box at this time and the position obtained by the initial regression (Figure 8). That is, the feature map was misaligned with the original image. However, for

small targets, such a method would lead to a relatively large deviation in the prediction box. To solve this problem, RoI Align [30] was used for improvement. Compared with RoI Pooling, RoI Align canceled its quantization operation but used the bilinear difference method to calculate the corresponding pixel value. The advantage of this improvement is that RoI Align is more accurate and stable when extracting features, especially when dealing with small grains or grain boundary details. Therefore, when using RoI Align for small-grain detection, the position and size of the grain can be more accurately determined, thereby reducing the error of grain detection. The formulas for bilinear interpolation were as follows:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \tag{1}$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \tag{2}$$

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \tag{3}$$

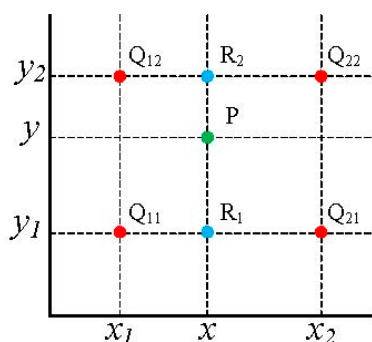


Figure 8. Illustration of RoI Align bilinear difference.

Among them, Q_{11} , Q_{12} , Q_{21} , and Q_{22} represented the pixel positions in the feature map; R_1 and R_2 were the intermediate values; and P was the projection position of the feature frame on the feature map.

RoI Align was performed on the feature map P_n to extract 7×7 features, and then the multi-layer feature map was made one-dimensional through the Flatten layer, and two FC-ReLU were performed in sequence. The last two FC-ReLU inputs to the branch served as regressions for classification and the bounding box.

2.2.2. Training of the Sea Grain Detection Model

The three image sub-sets served as inputs for transfer learning using a pre-trained deep neural network model. The algorithm was implemented in MMDetection, a target detection library based on the Pytorch deep learning framework developed by the MMLab Laboratory of the Chinese University of Hong Kong, and executed on a graphics workstation. The operating environment was the Windows 10 operating system, Pytorch 1.12.1, CUDA 11.6, MMDetection 1.25.1. The training of the model was conducted by a workstation equipped with a GPU (NVIDIA GeForce RTX 3090 with 12 GB Memory, NVIDIA Corporation, California, USA). The parameters of the model were fine-tuned, and their specific values were as follows: Among them, the learning rate was set to 0.001, the positive sample ratio of the convolutional layer was set to 0.75, and the batch size was set to 2. Also, during the entire model training process, the SGD (Stochastic Gradient Descent) was used and momentum was added to speed up the convergence, so that the model had higher accuracy after convergence. This could solve the problem of parameter optimization for large-scale grain data, and it had high computational efficiency and strong adaptability during the training process. The momentum was set to 0.8, while the epoch was set to 120. When

the loss function converged and stabilized, training was stopped and the training model was saved.

2.3. Evaluation Indicators

The RPN was trained end-to-end to generate high-quality region proposals, which were used by Fast R-CNN for detection. When training the RPN, the model would generate a large number of candidate frames based on the feature map, sort them according to their confidence, and evaluate the frames in turn. Therefore, IoU (Intersection over Union) was used as the measurement standard, and, finally, the non-maximum suppression was used to determine which boxes were the expected grains. IoU was the ratio of the overlapping area of the grain's actual area and the estimated area to the area occupied by the two areas as a whole. Generally speaking, when the IoU value was greater than 0.5, it could be considered an acceptable result, and 0.5 was selected as the threshold and used as the standard for evaluating the model [31]. Since the panicle grain covered a small area in the image, it was difficult to achieve a high-precision match between its predicted bounding box position and the position in the label. The appropriate IoU threshold was selected to measure the accuracy of the panicle grain position information.

$$\text{IoU} = \frac{\text{target} \cap \text{prediction}}{\text{target} \cup \text{prediction}} \quad (4)$$

The panicle grain results predicted by the proposed model were compared with the manually labeled panicle grains. The correctly detected panicle grains were called true positive (TP), and the undetected panicle grains were called false positive (FP). False negative was recorded when the background was incorrectly detected as panicle grains. From the three indicators, TP, FP, and FN, some indicators for model evaluation can be derived as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (7)$$

The meaning of precision was the measure of how many of all predicted targets were actually targets that were expected to be predicted, while the meaning of recall was how many of all the targets that were expected to be predicted were correctly predicted by model detection. Precision measured the model's ability to find positive samples in the entire data, while recall measured the model's ability to find positive samples in the entire positive-sample data set. The results of both were approximately as close to 1 as possible.

3. Results

3.1. Training of the Sea Grain Detection Model

The loss function and accuracy curves were used to evaluate the effect of sea rice model training. Then, the hyperparameters in model training were adjusted to obtain the optimal parameter configuration. Using the transfer learning method could greatly shorten the training time, so a weighted summation of the losses of the RPN and Fast R-CNN for joint training was used. The loss of Fast R-CNN was similar to that of the RPN. Also, the loss of the box was only calculated when the proposal box was a positive sample. Figure 9a showed the changes in the model loss function, where the abscissa represented the number of iterations, and the ordinate represented the loss value of the model loss function. It showed that the classification loss (cls loss) rapidly decreased in the first 20,000 iterations of training, then leveled off, and finally the model gradually converged. The regression loss (bbox regression loss) gradually decreased in the first 40,000 iterations

of training, then leveled off, and finally the model gradually converged. Figure 9b shows that the accuracy increased rapidly in the first 20,000 iterations and then continued to increase slowly between 20,000 and 40,000 iterations. After 40,000 iterations, the model gradually converged.

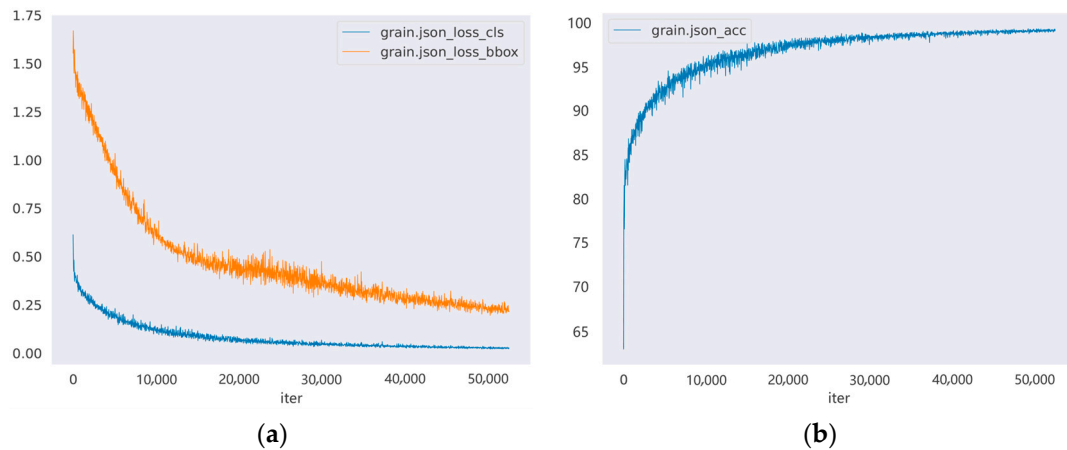


Figure 9. Illustration of training results for the sea grain detection model: (a) cls and bbox regression loss; (b) accuracy.

3.2. Detection Results of the Sea Grain Detection Model

Sea rice panicle images collected using different pixel-level cameras under different lighting conditions were used to test the detection performance of the model. Figure 10a–c are images of rice panicles collected at the center of the rice field. The backgrounds of these images are mainly rice panicle leaves and other rice panicles. The color of the target rice grain to be detected is relatively similar to the background, and there are also stacks between the panicle grains. It can be seen from the detection results that the proposed model can well identify most of the grains of rice panicles, and it does not identify the grains of other rice panicles in the image as positive samples. Figure 10d–f are images collected at the edge of the rice field. In the background of these images, in addition to rice leaves and other rice panicles, there are also weeds, soil, etc. It can be seen that the proposed model also had good accuracy for detecting grains in these cases. The above testing results showed that the proposed model had strong robustness in detecting sea rice grains in complex rice fields.

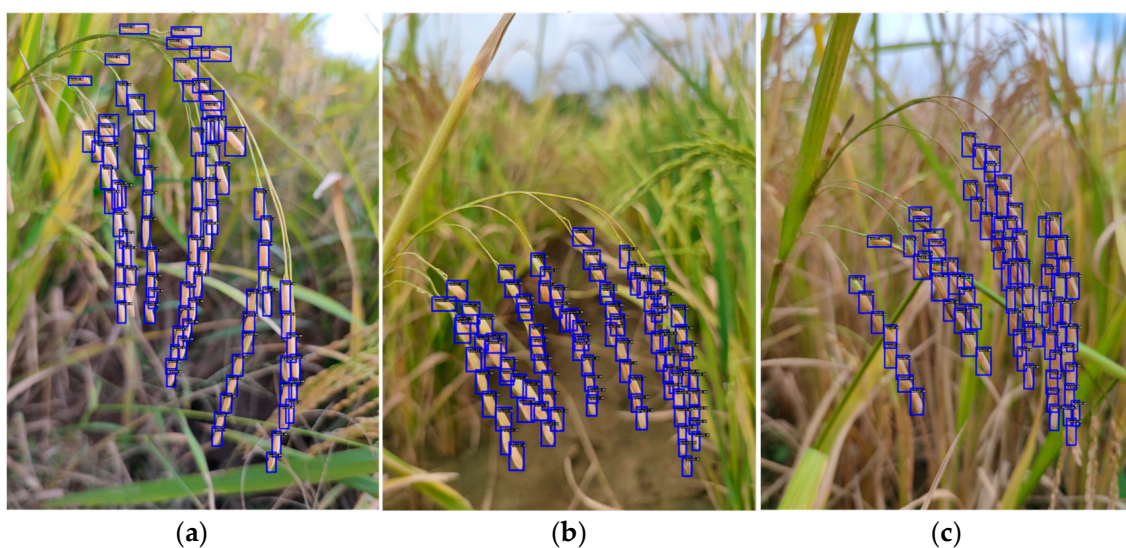


Figure 10. Cont.

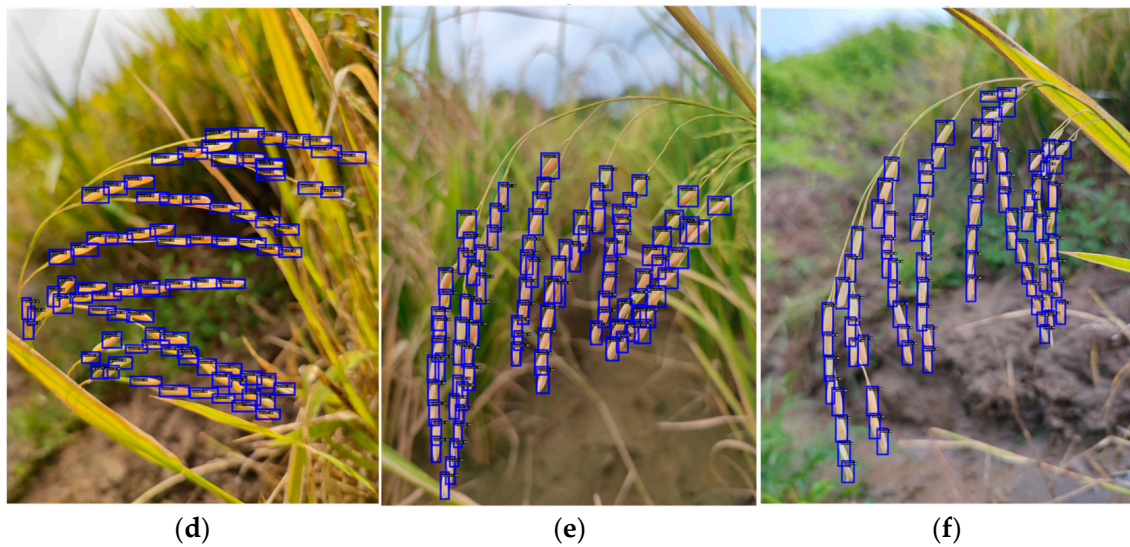


Figure 10. Detection results of sea grain detection model: (a–c) detection results of panicle images in middle part of rice field; (d–f) detection results of panicle images at the edge of rice field; blue box represents the location of sea rice grains detected in panicle image.

3.3. Comparison with Other Detection Models

During the training process, the precision (P) was taken as the vertical axis, and the recall rate (R) was taken as the horizontal axis; then, the precision–recall (P-R) array was drawn and connected into a curve to identify the corresponding P-R curve. The area enclosed by the P-R curve under different thresholds for each category was the Average Precision (AP). The average AP of different categories was called the mAP (mean Average Precision). Figure 11 shows the P-R curves of the sea grain detection model, YOLOv3, and Grid R-CNN. When the IoU threshold was 0.5, the mAP of the sea grain detection model reached above 0.9. For a detection model, the larger the area it encompassed, the better the performance. As can be seen in Figure 11, as the recall rate increased, the overall precision of the model showed a downward trend. When the recall rate was between 0% and 85%, the downward trend was slow, and when the recall rate was between 85% and 95%, it decreased rapidly. Overall, the area under the P-R curve of the sea grain detection model was the largest, indicating that our proposed model had better recognition results.

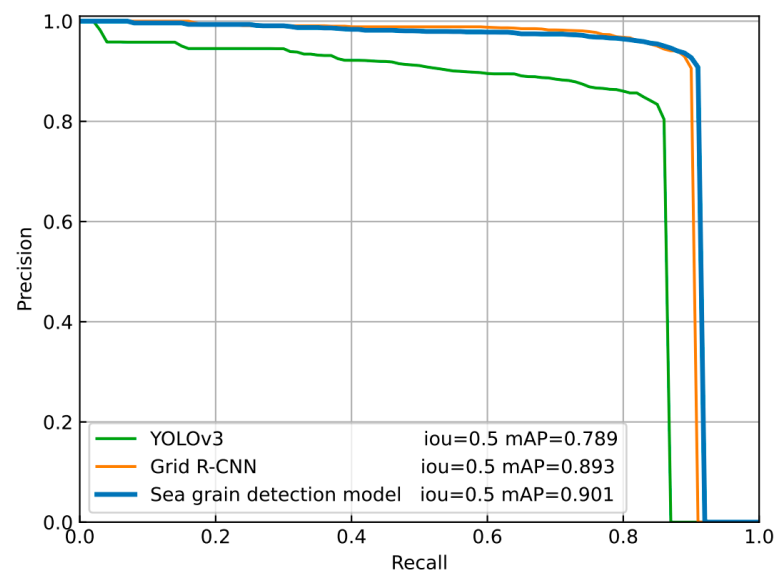


Figure 11. The P-R curve of the sea grain detection model.

The accuracy of the sea grain detection model and the Grid R-CNN model was much better than that of the YOLOv3 model (Table 2). Although the speed (epoch/s) of the proposed method was not much different from that of the YOLOv3 model, it was better than that of the Grid R-CNN model. Table 1 shows that the precision indicators of the sea grain detection model and Grid R-CNN models could be stable above 0.96. The sea grain detection model and Grid R-CNN model both had relatively strong abilities to discover the correlation of positive samples, and the recall index can also reach above 0.89. The precision indicators and recall indicators of YOLOv3 were much lower than that of the sea grain detection model. Overall, the proposed method could quickly and accurately detect sea rice grains in complex field environments.

Table 2. A comparison of the accuracy of the sea rice grain detection model with other models.

Index	Sea Grain Detection Model	YOLOv3	Grid R-CNN
Precision	0.97 ± 0.01	0.86 ± 0.01	0.96 ± 0.01
Recall	0.91 ± 0.03	0.83 ± 0.03	0.90 ± 0.03
mAP (%), IoU:0.5	90.1 ± 0.2	78.9 ± 0.2	89.3 ± 0.2
Time (epoch/s)	105	90	300

3.4. Counting Accuracy of the Sea Grain Detection Model

To evaluate the counting accuracy of the sea grain detection model, another 10 images of sea rice panicle samples were randomly selected to carry out the counting experiments. The counting results for these samples are summarized in Table 3. The average counting accuracy for the sea grain detection model was 94.9%, demonstrating that the proposed method performed well in counting the grain number per sea rice panicle.

Table 3. The counting accuracy of the sea grain detection model.

No.	Actual Number	Counting Result	Accuracy
1	69	67	97.1%
2	75	64	85.3%
3	103	97	94.2%
4	87	81	93.1%
5	88	86	97.7%
6	97	93	95.9%
7	117	100	85.5%
8	82	82	100.0%
9	92	92	100.0%
10	93	93	100.0%
Mean			94.9%

4. Discussion

Since the aim of this paper was to directly detect sea rice grains in complex rice fields, it was inevitable that there would be many external interference factors that may directly affect the accuracy of the detection results. Therefore, it was necessary to analyze the impact of these problems on the detection accuracy of the sea rice grain detection model. In the experiment, the factors that may affect the detection accuracy mainly included blurred rice panicle images, occlusion grains, and a complex background.

4.1. Effect of Blurred Rice Panicle Images

Since the images were taken in a complex field environment, the quality of the images could be easily affected by many aspects, resulting in the blurring of the grains to be detected and unclear boundaries. Through data analysis, it was known that the accuracy of blurred images was 8% to 10% lower than that of the images with clear grains. To address this type of problem, image enhancement technology was used to blur the images in the data set so that the model could learn the characteristics of blurred grains, thereby enhancing the

robustness of the model. By learning the characteristics of blurred grains, the established model could also correctly predict blurred grains in actual detection (Figure 12a). This showed that the generalization ability and adaptability of the model could be effectively enhanced through data enhancement. Also, the proposed method could efficiently identify sea rice grains in practical applications and meet the requirements of application promotion and online identification.

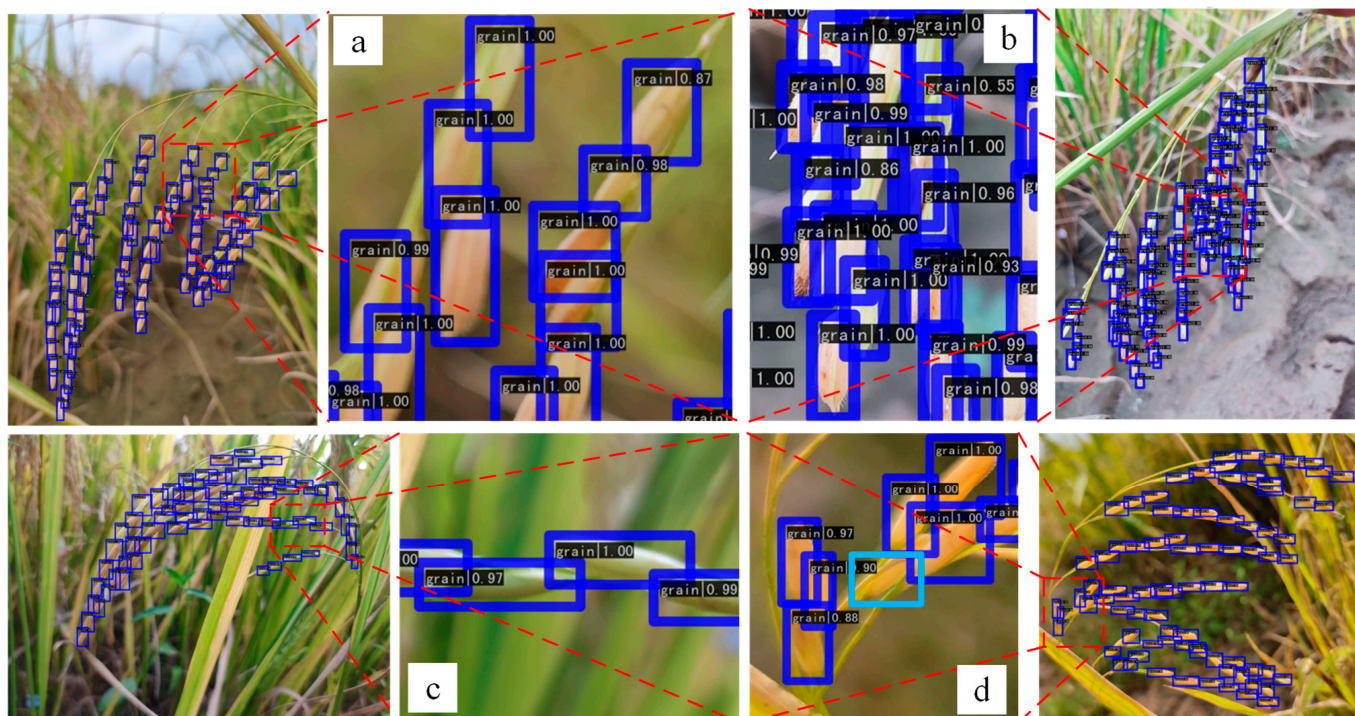


Figure 12. Details of grain detection in different situations: (a) grains are blurred; (b) grains overlap densely; (c) grains are similar to the background; (d) grain features are missing; dark blue boxes represent the sea rice grains that are correctly detected; sky blue boxes represent the sea rice grains that are missed.

4.2. Effect of Covering Grains on Sea Rice Panicle

The experimental rice panicle data consisted of unharvested rice panicles. Thus, the rice panicles were in a natural state when the images were taken, and the grains on the rice panicles could not be separated. Additionally, since some grains had not yet reached maturity, they had not spread out, leading to thorny covering problems. This article addresses this issue from two aspects: data collection and model training.

Since the samples used were rice panicles in fully mature and withered stages, the yellow color of most of the rice husks faded away, and the grain weight of the rice panicles reached the maximum. The grains, therefore, clumped together due to gravity, resulting in mutual occlusion. To achieve a more ideal recognition effect, the panicle grains were separated and covered, which could more quickly and effectively solve the problem of being unable to be identified as a positive sample due to the lack of panicle grain features. Therefore, when collecting data, the camera shooting plane must be parallel to the plane produced by the bending of the rice panicle to better obtain more characteristic information about the rice grains. In most cases, it was necessary to manually lift the tip of the rice panicle so that both ends of the rice panicle had support points. This allowed the grains to be distributed as widely as possible. This method was effective both in model training and practical applications.

The second issue was the mutual covering between rice panicles, which cannot be directly solved by the above method. This is mainly due to immature rice panicles, causing the grains to remain stick together. Directly separating the grains would damage the rice

panicles. Therefore, the original state of panicles in this situation must be retained for training to enhance the model's robustness, but a large amount of such data is required for effective training.

Figure 12b shows the detection results of overlapping grains on the rice panicles. Due to the multi-scale feature fusion characteristics of the sea rice grain model, richer feature information can be obtained to better understand the positional semantic information between rice grains. However, the model may miss the detection of grains where grain features were missing (Figure 12d). Since the features of obscured and blurred grains were greatly reduced compared to normal grain features, it was difficult for the proposed model to detect such grains.

4.3. Effect of Background in Complex Rice Fields

Compared with normal-sized targets in general target detection tasks, grains on rice panicles as small targets had less feature information, so it was difficult for the model to distinguish them from the background. To solve this problem, a multi-scale learning method was adopted, which could not only obtain higher-scale features of the grain but also ensure that the spatial position information of the grain was obtained. However, in a field environment, the original color of rice panicles was very similar to the background, and it was easy to mistakenly identify the background as a positive sample. To solve this problem, the layers of the neural network were deepened so that the model could learn richer feature information and better fit the features, but the training time also increased as a result.

Since the grains on other rice panicles share the same characteristic information as those on the target rice panicle, if the model identifies grains from other rice panicles in the background as positive samples, it cannot be considered to have produced false positives. However, the research purpose of this paper is to detect only the grains on one rice panicle for finally obtaining the number of grains per rice panicle. Therefore, this problem cannot be solved by existing methods. The model must first distinguish different rice panicles and then identify the grains on a complete bunch of rice panicles. Therefore, some measures were taken to reduce the occurrence of this situation as much as possible, such as avoiding collecting images of non-target rice panicles during the data collection process or making sure that there were no other rice panicles in the same focal section of the target rice panicle. In this way, the grain itself, which was a small target, also lacked characteristic information due to being out of focus. Even though some grains appeared turquoise, which was similar to the background color, the model was still sensitive to color information and could effectively separate the rice panicles from the background (Figure 12c).

4.4. Improvement of Model Performance

In the detection task of this paper, it was difficult to directly collect data from unharvested rice panicles in the rice field to obtain relatively high-quality rice panicle images since factors such as a complex background, unbalanced light, out-of-focus photographs, and swinging rice panicles may affect the quality of the image. Even though some shooting techniques had been put forward, there were still some difficult problems, such as the lack of features caused by too-dense rice panicles. However, if excessive efforts were made to ensure image quality during collection, it would increase researchers' learning costs and reduce efficiency. Therefore, in response to the above problems, it was necessary to consider the training of the model, add rice panicle images collected under different conditions, and input them into the model for training to further enhance the robustness of the model. Adding an attention mechanism to the network could also slightly improve the model's performance. However, due to the complex image background in this study, the attention mechanism would focus significantly on noise points, increasing the training time required for detection.

5. Conclusions

This paper explores the feasibility of using deep-learning-based computer vision methods to detect and count the grain number of sea rice panicles in complex field environments. Using the deep learning convolutional neural network (CNN) model, a high-precision and high-efficiency method was developed and evaluated using images collected in a field. The following conclusion was drawn. The developed sea grain detection model was capable of identifying and calculating the grain number per sea rice panicle. The sea grain detection model was found to be the most efficient model for recognizing and counting grains in a complex field environment in terms of the mean Average Precision (mAP). When compared to other deep learning models, the sea grain detection model had the highest mAP of 90.1%. The accuracy of the sea grain detection model was 94.9%. This method can provide a reference for the computer-aided calculation of the number of sea grains per panicle, help rice breeders to automatically collect yield-related data, and, thus, help agricultural researchers to predict rice yields. However, more tests may be required to further verify the sea grain detection model.

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

Funding: This research was funded by the Guangdong Basic and Applied Basic Research Foundation (2022A1515110468) and the Program for Scientific Research Start-Up Funds of Guangdong Ocean University (060302062106).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Acknowledgments: The authors thank their partner, the Agricultural Research Institute of Huguang Campus, Guangdong Ocean University, for providing the rice panicles for image collections.

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

References

1. Wang, Z.; Li, N.; Xu, Y.; Wang, W.; Liu, Y. Functional activity of endophytic bacteria G9H01 with high salt tolerance and anti-Magnaporthe oryzae that isolated from saline-alkali-tolerant rice. *Sci. Total. Environ.* **2024**, *926*, 171822. [[CrossRef](#)] [[PubMed](#)]
2. Qin, H.; Li, Y.; Huang, R. Advances and Challenges in the Breeding of Salt-Tolerant Rice. *Int. J. Mol. Sci.* **2020**, *21*, 8385. [[CrossRef](#)] [[PubMed](#)]
3. Hoang, T.M.L.; Tran, T.N.; Nguyen, T.K.T.; Williams, B.; Wurm, P.; Bellairs, S.; Mundree, S. Improvement of Salinity Stress Tolerance in Rice: Challenges and Opportunities. *Agronomy* **2016**, *6*, 54. [[CrossRef](#)]
4. Huong, C.T.; Anh, T.T.T.; Tran, H.-D.; Duong, V.X.; Trung, N.T.; Khanh, T.D.; Xuan, T.D. Assessing Salinity Tolerance in Rice Mutants by Phenotypic Evaluation Alongside Simple Sequence Repeat Analysis. *Agriculture* **2020**, *10*, 191. [[CrossRef](#)]
5. Wu, J.; Yang, G.; Yang, X.; Xu, B.; Han, L.; Zhu, Y. Automatic Counting of in situ Rice Seedlings from UAV Images Based on a Deep Fully Convolutional Neural Network. *Remote Sens.* **2019**, *11*, 691. [[CrossRef](#)]
6. Lu, Y.; Chuan, M.; Wang, H.; Chen, R.; Tao, T.; Zhou, Y.; Xu, Y.; Li, P.; Yao, Y.; Xu, C.; et al. Genetic and molecular factors in determining grain number per panicle of rice. *Front. Plant Sci.* **2022**, *13*, 964246. [[CrossRef](#)]
7. Duan, L.; Yang, W.; Bi, K.; Chen, S.; Luo, Q.; Liu, Q. Fast discrimination and counting of filled/unfilled rice spikelets based on bi-modal imaging. *Comput. Electron. Agric.* **2011**, *75*, 196–203. [[CrossRef](#)]
8. Yu, L.; Shi, J.; Huang, C.; Duan, L.; Wu, D.; Fu, D.; Wu, C.; Xiong, L.; Yang, W.; Liu, Q. An integrated rice panicle phenotyping method based on X-ray and RGB scanning and deep learning. *Crop J.* **2021**, *9*, 42–56. [[CrossRef](#)]
9. Yang, W.; Guo, Z.; Huang, C.; Duan, L.; Chen, G.; Jiang, N.; Fang, W.; Feng, H.; Xie, W.; Lian, X.; et al. Combining high-throughput phenotyping and genome-wide association studies to reveal natural genetic variation in rice. *Nat. Commun.* **2014**, *5*, 5087. [[CrossRef](#)]
10. Li, J.; Li, Y.; Qiao, J.; Li, L.; Wang, X.; Yao, J.; Liao, G. Automatic counting of rapeseed inflorescences using deep learning method and UAV RGB imagery. *Front. Plant Sci.* **2023**, *14*, 1101143. [[CrossRef](#)]

11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference; Proceedings, Part I 14, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
12. Wang, H.; Wang, T. Multi-Scale Residual Aggregation Feature Pyramid Network for Object Detection. *Electronics* **2022**, *12*, 93. [[CrossRef](#)]
13. Wang, Q.; Zhou, L.; Yao, Y.; Wang, Y.; Li, J.; Yang, W. An Interconnected Feature Pyramid Networks for object detection. *J. Vis. Commun. Image Represent.* **2021**, *79*, 103260. [[CrossRef](#)]
14. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28.
15. Khaki, S.; Pham, H.; Han, Y.; Kuhl, A.; Kent, W.; Wang, L. Convolutional Neural Networks for Image-Based Corn Kernel Detection and Counting. *Sensors* **2020**, *20*, 2721. [[CrossRef](#)] [[PubMed](#)]
16. Wang, C.; Zhong, C. Adaptive Feature Pyramid Networks for Object Detection. *IEEE Access* **2021**, *9*, 107024–107032. [[CrossRef](#)]
17. Wu, W.; Yang, T.-L.; Li, R.; Chen, C.; Liu, T.; Zhou, K.; Sun, C.-M.; Li, C.-Y.; Zhu, X.-K.; Guo, W.-S. Detection and enumeration of wheat grains based on a deep learning method under various scenarios and scales. *J. Integr. Agric.* **2020**, *19*, 1998–2008. [[CrossRef](#)]
18. Gong, Y.; Xiao, Z.; Tan, X.; Sui, H.; Xu, C.; Duan, H.; Li, D. Context-aware convolutional neural network for object detection in VHR remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 34–44. [[CrossRef](#)]
19. Dandriofosse, S.; Ennadifi, E.; Carlier, A.; Gosselin, B.; Dumont, B.; Mercatoris, B. Deep learning for wheat ear segmentation and ear density measurement: From heading to maturity. *Comput. Electron. Agric.* **2022**, *199*, 107161. [[CrossRef](#)]
20. Wang, X.; Yang, W.; Lv, Q.; Huang, C.; Liang, X.; Chen, G.; Xiong, L.; Duan, L. Field rice panicle detection and counting based on deep learning. *Front. Plant Sci.* **2022**, *13*, 966495. [[CrossRef](#)] [[PubMed](#)]
21. Deng, R.; Qi, L.; Pan, W.; Wang, Z.; Fu, D.; Yang, X. Automatic estimation of rice grain number based on a convolutional neural network. *J. Opt. Soc. Am. A* **2022**, *39*, 1034–1044. [[CrossRef](#)]
22. Gedraite, E.S.; Hadad, M. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. In Proceedings of the ELMAR-2011, Zadar, Croatia, 14–16 September 2011; pp. 393–396.
23. Keller, A.; Eng, J.; Zhang, N.; Li, X.; Aebersold, R. A uniform proteomics MS/MS analysis platform utilizing open XML file formats. *Mol. Syst. Biol.* **2005**, *1*, 2005-0017. [[CrossRef](#)]
24. Marcel, S.; Rodriguez, Y. Torchvision the machine-vision package of torch. In Proceedings of the 18th ACM International Conference on Multimedia, Firenze, Italy, 25–29 October 2010; pp. 1485–1488.
25. Jung, A.B.; Wada, K.; Crall, J.; Tanaka, S.; Graving, J.; Reinders, C.; Yadav, S.; Banerjee, J.; Vecsei, G.; Kraft, A.; et al. *Imgaug*; GitHub: San Francisco, CA, USA, 2020. Available online: <https://github.com/aleju/imgaug> (accessed on 9 July 2024).
26. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. [[CrossRef](#)]
27. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
28. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
30. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
31. Yan, J.; Wang, H.; Yan, M.; Diao, W.; Sun, X.; Li, H. IoU-Adaptive Deformable R-CNN: Make Full Use of IoU for Multi-Class Object Detection in Remote Sensing Imagery. *Remote Sens.* **2019**, *11*, 286. [[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.