information

MDPI

Article

WDN: A One-Stage Detection Network for Wheat Heads with High Performance

Pengshuo Sun, Jingyi Cui, Xuefeng Hu and Qing Wang *

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China;
2019308250104@cau.edu.cn (P.S.); jingle.cui@cau.edu.cn (J.C.); 2018308130302@cau.edu.cn (X.H.)
* Correspondence: wangqingait@cau.edu.cn

Abstract: The counting of wheat heads is labor-intensive work in agricultural production. At present, it is mainly done by humans. Manual identification and statistics are time-consuming and error-prone. With the development of machine vision-related technologies, it has become possible to complete wheat head identification and counting with the help of computer vision detection algorithms. Based on the one-stage network framework, the Wheat Detection Net (WDN) model was proposed for wheat head detection and counting. Due to the characteristics of wheat head recognition, an attention module and feature fusion module were added to the one-stage backbone network, and the formula for the loss function was optimized as well. The model was tested on a test set and compared with mainstream object detection network algorithms. The results indicate that the mAP and FPS indicators of the WDN model are better than those of other models. The mAP of WDN reached 0.903. Furthermore, an intelligent wheat head counting system was developed for iOS, which can present the number of wheat heads within a photo of a crop within 1 s.

Keywords: wheat heads; object detection; one-stage model

check for updates

Citation: Sun, P.; Cui, J.; Hu, X.; Wang, Q. WDN: A One-Stage Detection Network for Wheat Heads with High Performance. *Information* **2022**, *13*, 153. https://doi.org/10.3390/info13030153

Academic Editor: Stefano Berretti

Received: 1 February 2022
Accepted: 9 March 2022
Published: 16 March 2022
Retracted: 18 October 2022

1. Introduction

Wheat belongs to the family Gramineae. It is one of the three grains, and it is a widely planted cereal crop [1]. The caryopsis of wheat is a staple food of humans—most caryopsis production is used for human consumption, and only about one-sixth is used as feed. Wheat can be ground into flour to make bread, biscuits, noodles, and other food, or it can be fermented into beer, alcohol, liquor, or biofuel. The wheat growth stage is usually divided into the green, jointing, heading, filling, and maturity stages. From the heading stage to the mature stage, water and fertilizer management of the wheat field significantly influence wheat yield and quality. Spike number per unit of ground area is one of the main agronomic traits related to the grain yield of wheat [1]. Rapid assessment based on this trait can help monitor crop management measures' efficiency and facilitate early prediction of food yield. This trait can also serve as a phenotypic trait in breeding programs.

Accurate wheat head counting is the critical step to obtaining wheat head characteristics and detecting wheat phenotype automatically. With the development of machine vision and deep learning technology [2–4], the number of ears of wheat can theoretically be counted automatically and accurately by machine vision. However, using machine vision technology to identify wheat heads is a complex problem, as the appearance of wheat heads—including the shape, size, texture, and posture—varies significantly between different wheat varieties and growth stages. The edges of wheat heads are irregular, and the color of the ears is similar to the color of the leaves. In the complex environment of a wheat field, mutual sheltering between different wheat organs and the uneven and constantly changing natural sunlight severely hinder the automatic identification of wheat heads. The influence of varying growth environments also needs to be considered. There is a great need for a machine learning model capable of effectively detecting wheat heads in diverse environments.

This paper has been retracted. A retraction notice was published on 18 October 2022 in *Information* **2022**, *13*, 502. https://doi.org/10.3390/info13100502

Some researchers have made progress in the field of image-based wheat head recognition. Jose et al. [5,6] used traditional image processing algorithms, such as a Laplacian frequency filter and a median filter, to recognize wheat heads based on RGB images automatically. In a test set, the recognition accuracy reached more than 90%, and in a field experiment, the recognition accuracy was higher than that of the artificial wheat head recognition method. Pouria et al. [7] proposed a method called DeepCount, which can automatically recognize and count wheat heads based on digital images of wheat heads under natural field conditions. This method uses simple linear iterative clustering to segment images into superpixels to obtain canopy-related features. It then builds feature models and inputs them into deep convolutional neural networks (CNNs) for classification. DeepCount achieved a maximum R^2 of 0.89 on an experimental dataset. Tan et al. [8] used simple linear clustering to identify wheat heads. The experimental results showed that the recognition accuracy was 94% on a wheatear image set containing wheat that was given a high level of nitrogen, and 80% on a wheatear image set containing wheat that received no additional nitrogen. Zhou et al. [9] used vehicle-mounted RGB camera equipment to collect data samples in a wheat field dynamically and trained a wheat head recognition model with the twin-support-vector-machine segmentation model. The accuracy of automatic wheat head recognition was almost the same as that of manual recognition. Zhou et al. [10] used unmanned aerial vehicles to collect rice spike images. They adopted CNNs based on improved region-based fully convolutional neural networks, and the model reached 87% recognition accuracy. Hayal et al. [11] used an unsupervised Bayesian learning method to identify the rice spikes based on images of rice collected by unmanned aerial vehicles. It had a recall rate of 96% and an accuracy rate of 72%. Deng et al. used the CNN model to analyze the number of grains in a panicle of rice. The model integrated the feature pyramid network (FPN) [12] into the faster region-based CNN (faster R-CNN) network, and the model's accuracy reached 99% [13]. In conclusion, existing studies have made some valuable attempts at using deep learning methods for wheat head recognition. Still, recognition speed and accuracy are not high enough, so there is room for improvement.

The scientific problem of wheat spike detection is target detection in images, which involves two main challenges: detection speed and recognition accuracy. The detection rate depends on the architecture and type of the model. For example, the two-step detection algorithm approach is usually slower than the one-step detection model. The accuracy of recognition depends on whether the model can distinguish the characteristics of wheatears at different growth stages, that is, the shape, size, texture, and growth posture of wheatears. Modeling challenges include the number of annotated data samples, the quality of annotations, and the model training method. Without a sufficient number of high-quality labeled samples, the practical training of deep learning class models cannot be completed, and without practical training, the models cannot achieve optimal effects.

In this study, a one-stage target detection model called Wheat Detection Net (WDN) was developed for wheat head detection. Some component units of the model structure were borrowed from the YOLO model, and some module units were designed according to the characteristics of wheat head detection. In the training stage of the model, various methods were adopted to improve the training effect of the model. The detection speed of WDN was 37 FPS, and the mAP value of detection accuracy was 0.903, those values being better than those of other models. The main contributions of this study follow. (1) One-stage architecture was adopted to ensure the detection speed of the model. (2) An attention refinement module was designed to ensure the perceptual ability of wheat spike features captured by the model. (3) A feature fusion module was designed to fuse features of different scales to improve the performance of target recognition. (4) Various model optimization techniques were used comprehensively, such as loss function selection, label smoothing application, and OOF threshold calculation. (5) In model training, the warm-up method and pseudo-label training were used to accelerate model convergence. (6) Based on the proposed WDN, an easy-to-use mobile app was designed, which can be downloaded from the App Store.

The rest of this article is organized into four sections. The Section 3 (Materials and Methods) introduces the dataset used in the research and the design details of the model; the Section 4 (Experiment) describes the experimental process, results, and analysis; and the Section 6 (Conclusions) summarizes the study and its findings.

2. Related Work

Object detection is a vibrant research areas in computer vision and an essential part of image content analysis and understanding. It plays a notable role in automatic driving and medical image diagnosis applications. This section primarily introduces the deep learning techniques adopted in object detection.

The excellent performance of AlexNet [14] on ImageNet has led scientists to recognize that convolutional neural networks are an efficient framework for processing image data. Thanks to the flexibility of CNNs, they are used in various computer vision tasks, including object detection. Regarding CNNs, object detection algorithms can be divided into two categories: two-stage algorithms and one-stage algorithms.

2.1. Two-Stage Algorithms

Two-stage algorithms' specific stages are shown as follows:

Stage 1 : Generate regional proposals from images.
Stage 2 : Generate final object borders from region proposals.

Ross et al. [15] proposed R-CNN in 2014, which first selects possible object frames from a set of object candidate frames using the selective search algorithm Selective Search. Then it resizes the images in these selected object frames to a particular fixed size and feeds them to a CNN model (trained on ImageNet). Eventually, the extracted features are fed to a classifier to predict whether a target is detected in that object frame, and then further predicts which class the detected target belongs to. Although the R-CNN algorithm has made meaningful progress, the redundant computation of overlapping frame features makes the detection of the whole network slow. In order to reduce the redundant computations caused by a large number of overlapping frames, K. He et al. [16] suggested SPP-Net, which has a unique structure, including a spatial pyramid pooling layer (SPP). The core idea is to divide an image into blocks of several scales (one image into 1, 4, 8, etc.) and then fuse the extracted features of each block to take into account the features of multiple scales. When using SPP-Net for object detection, the entire image is computed only once to generate the corresponding feature map, which avoids repetitive computations of convolutional feature maps. SPP-Net employs support vector machines (SVMs) for classification, which requires enormous storage space, and the model is trained only for the fully connected layer.

In 2015, Ross et al. presented Fast R-CNN [17], which refined R-CNN and SPP-Net. It starts with an input image, which is passed to the CNN to extract features and return potential ROIs, after which an ROI pooling layer is applied to the ROIs to ensure that each region has the same size. Ultimately, the features of these regions are passed to the fully connected layer of the network for classification. Although Fast R-CNN takes only two seconds to process an image (R-CNN takes 14 s), its speed is still not fast enough to be used in actual production. For the consideration of using CNN models to generate candidate frames directly, Ren et al. [18] proposed faster R-CNN, which is the first end-to-end deep learning detection algorithm that achieves close to real-time performance. This network's primary innovation is a region selection network for generating candidate frames, significantly improving the generation speed of detection frames. In 2017, Lin et al. [12] suggested a feature pyramid network FPN based on faster R-CNN. FPN proposes a top-down network architecture with lateral connections to build high-level semantic information. It immensely improved detection network accuracy (especially for some datasets with large-scale variations in the objects to be detected).

2.2. One-Stage Algorithms

YOLO-v1 [19] is the first one-stage deep learning detection algorithm that is extremely fast, and the algorithm's idea is to divide each image into multiple grids and then predict the bounding box for each grid simultaneously and give the corresponding probability. Although YOLO-v1 is remarkably fast compared to two-stage algorithms, its accuracy is lower than the latter, especially with small target objects. After that, Liu et al. [20] suggested the SSD algorithm. The principal innovations of this algorithm are the proposed multi-reference and multi-resolution detection techniques. The difference between the SSD algorithm and some previous detection algorithms is that partial previous detection algorithms only detect at the deepest branch of the network. In contrast, SSD has multiple different detection branches, which can detect multiple scales of targets. Consequently, SSD dramatically improves the accuracy of multi-scale target detection and is much more pleasing for small target detection. YOLO-v4 [21] is the fourth version of the YOLO algorithm. Specifically, (1) mosaic data enhancement, cmBN, and SAT self-adversarial training are introduced on the input side; (2) YOLO-v4 incorporates various new approaches on the feature extraction network, containing CSPDarknet53, Mish activation function, and Dropblock; (3) in the detection head, the SPP module is introduced. Overall, YOLO-v4 has outstanding engineering significance—introducing the latest research achievements to YOLO-v4 in the field of deep learning in recent years, and having made a giant stride on the basis of YOLO-v3.

3. Materials and Methods

3.1. Dataset Analysis

The dataset we used was from Kaggle. The wheat images in the dataset were collected outdoors. It is divided into two parts: a training set and a test set. There are more than 3000 images in the training set, which covers multiple regions, including Europe (France, the United Kingdom, and Switzerland) and North America (Canada). The test set includes around 1000 images from Australia, Japan, and China. As shown in Figure 1, it contains images taken under a variety of weather conditions and lighting conditions, and at different growth periods of wheat.

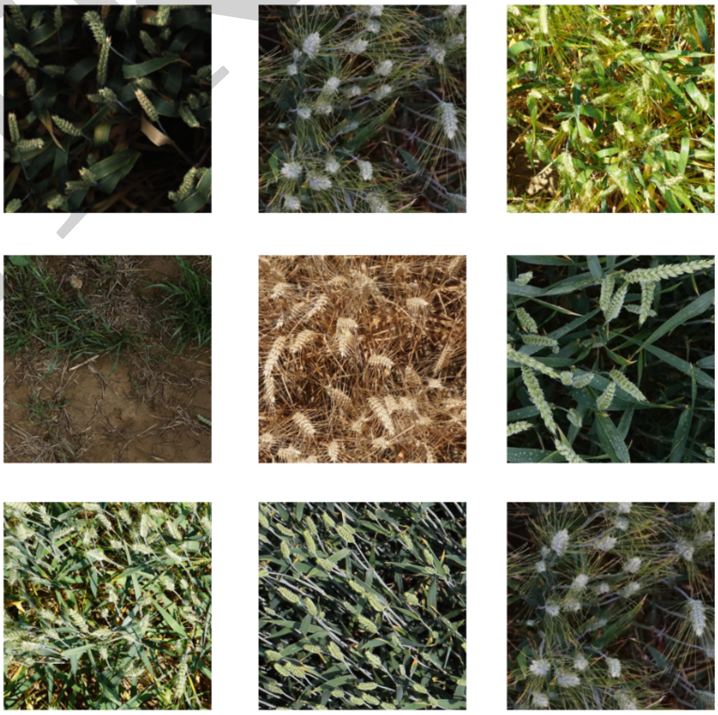


Figure 1. Samples of the dataset.

The dataset used in this paper has several difficulties for processing:

1. Dense wheat plants overlap frequently;
2. Wind blurs the photos occasionally;
3. The appearance varies with maturity, color, genotype, and head orientation.

Through further analysis of the data samples, it was found that the numbers of detection bound boxes in the samples from the training set were distributed normally, as shown in Figure 2.

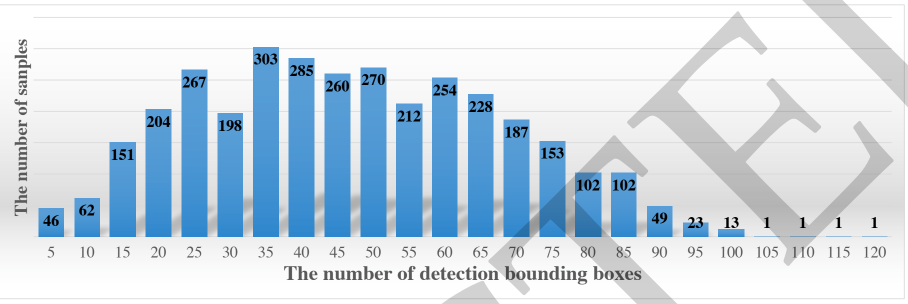


Figure 2. Distribution of the detection bounding boxes.

Most samples contained 20–60. Forty-nine samples did not contain detection bound boxes, and there was a single sample containing 116 bounding boxes. These bounding boxes were too sparse or too dense, which negatively influenced model training. In this dataset, the mAP value was only 0.7–0.8 using YOLO or SSD network, but that is already higher than the first ranking in global wheat detection when fusing the WDN512 + YOLO series(v3, v4, and v5) + MaskRCNN model.

3.2. Data Augmentation

Data enhancement was adopted to solve the problem of insufficient network training due to inadequate data or performance degradation caused by overfitting. Methods of the data enhancement included image translation, image rotation, image horizontal and vertical flip, image cropping, and repositioning. Referring to the method proposed by Alex et al. [14], first, we cropped the image into five parts, and then flipped the five images horizontally and vertically, so each original image would eventually generate fifteen expanded images. The outer bound boxes of the cropped training set images were counted in order to prevent the outer bound boxes from being cropped. Next, HSV [22] channel color changes were performed on the dataset, to use hue and saturation. Value space was used to represent the RGB color space.

3.2.1. Cutout

The cutout [23] method is randomly cutting out some areas in a sample and fill them with certain pixel values, and the classification labels should remain unchanged. As shown in Figure 3A, the specific operation involves using a fixed-size rectangle to block the image. Within the rectangle, all values are set to zero or other pure color values. Cutout enables the convolutional neural network to use the global information of the entire image rather than the local information composed of some detailed features. The cutout method is able to simulate the effect of the wheat being obscured. Moreover, cutout creates a similar effect with dropout in the preprocessing stage. Dropout randomly discards neurons, and cutout randomly discards image pixels. Through this preprocessing method, the robustness of the model could be effectively improved.

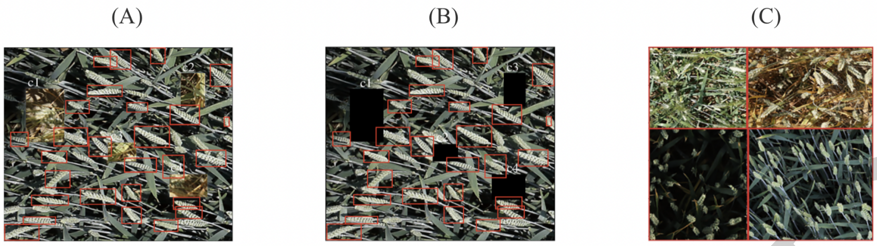


Figure 3. Illustrations of three data enhancement methods. (**A**) Cutout method; (**B**) cutmix method; (**C**) mosaic method.

3.2.2. Cutmix

The cutmix [24] method cuts out a part of an area and fills it with the training set randomly instead of 0 pixels, as shown in Figure 3B. Cutmix enables the model to identify two objects from a partial view of an image, which improves the efficiency of training. Cutout could make the model focus on the areas where the objects are difficult to distinguish, but some areas have no information, which would negatively affect the training efficiency. In contrast, cutmix makes full use of all the pixel information. In order to make full use of the background image that does not contain wheat heads in the dataset, when the cutmix was used in this study, the areas with the wheat heads and the areas without the wheat heads were subjected to a 1:1 cutmix operation.

3.2.3. Mosaic

Mosaic [25] can use multiple pictures at once, and its most significant advantage lies in the fact that it can enrich the backgrounds of the detected objects, and it calculates the data of multiple pictures during the BN calculation, which effectively promotes the generalization of the model. Its processing method is shown in Figure 3C.

3.3. Wheat Detection Network

Although the current mainstream one-stage models, such as YOLO [26] and SSD, have achieved excellent performance on COCO and VOC datasets, they still need to be improved when it comes to wheat detection. The main reasons are as follows:

1. The aforementioned algorithms were used on the COCO and VOC datasets, so the anchor points of the algorithms are not universal and need to be adjusted.
2. The detection accuracy of the aforementioned networks, especially when it comes to small objects, is low. Therefore, based on the idea of a one-stage network, the Wheat Detection Net model was proposed, with the aim of wheat labeling. Its main features include: (1) adding an attention module to the backbone networks to enhance the ability to extract features; (2) adding a multi-scale feature fusion module to the backbone network, and referring to the ideas of two feature fusion networks, Feature Pyramid Networks (FPN) and the Path Aggregation Network (PANet), to optimize the fusion module.
3. The loss functions cannot perform different loss calculations on the wheat heads and the background, that is, the foreground and the background.
4. Smooth activation functions allow better information penetration into the neural network, resulting in better accuracy and generalization. Therefore, we replaced the LeLU and LeakyReLU commonly used in CNN with the Mish function, as shown in Figure 4.

The improved network's results are shown in Figure 5.

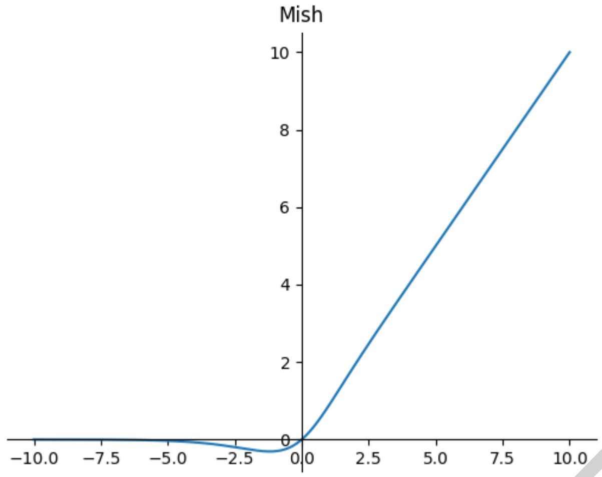


Figure 4. Mish activation function.

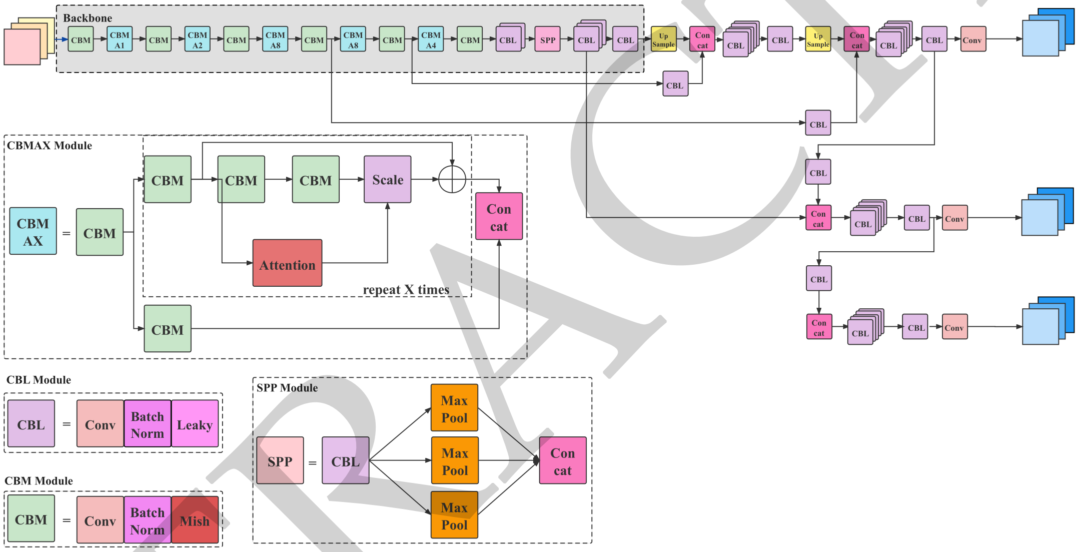


Figure 5. Wheat Detection Net.

3.3.1. Attention Refinement Module

The visual attention mechanism is a brain signal processing mechanism, a unique mechanism of human vision, as shown in Figure 6. For human vision, the object area that needs to be focused on could be located by quickly scanning the global image, and it is commonly referred to as the focus of attention. Then we pay more attention in this area to obtain more valuable detailed information of the object, and suppress other useless information. This attention mechanism could also be widely applied in the field of computer vision. The attention model [27] here functions similarly to human visual selective attention. When outputting a certain word entity, attention will be focused on the corresponding area.

Based on the above facts, it is proposed in this paper that an attention refinement module could be applied to contextual information branching, to refine the output of the last two stages. Global average pooling was adopted at the beginning to obtain the maximum receptive field, thereby integrating the global contextual semantic information. Next, the attention module training network was supposed to learn intensively, even if the features had different weights. Specifically, the attention refining module calculated the weight for each channel of the feature map, and then weighted each original output channel with the corresponding weight to obtain the new weighted feature, which played a role in re-adjusting the integrated features. This attention mechanism achieved the effect of refining and optimizing the output of the two stages of the context information branch (i.e., the fourth and fifth stages of ResNet18 [28] downsampling) with only a small number of calculations, and was capable of obtaining the global contextual semantic information in a simple and fast way. In the attention refining module, through a series of operations, the response weight matrix of each position in the feature map with respect to all positions was obtained, and then the sigmoid function was used to map the weight to a number between 0 and 1. Then, the weight was multiplied by the feature, which was the weighted response feature.

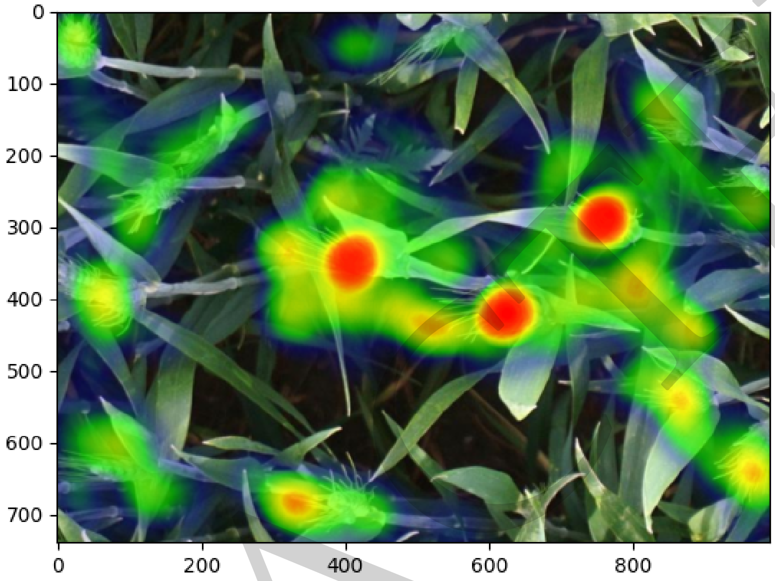


Figure 6. Illustration of human visual attention.

3.3.2. Feature Fusion Module

The spatial details captured by the spatial information branch were more abundant, and the features captured by the contextual information branch contained rich contextual information. The features they output included shallower layer and deeper layer, which were not at the same level. Hence, they could not be directly merged, and instead required a module that specialized in fusing features of these different scales—exactly the feature fusion module designed in this article. The FFM learning attention mask is used to select and combine features. First of all, it concatenates different input features and then performs conventional convolution operations. Next, it copies SENet [27] by using the attention mechanism for feature optimization. Similarly to the aforementioned ARM structure, the eigenvectors of features after concatenation are obtained through global average pooling. The weights of different features are calculated through the convolution and activation function. Then, weights are added to the features to generate the new weighted features. Finally, it they are added to the original features.

3.3.3. Loss Function

Aiming to label wheat heads, the loss function was divided into three parts: regression box loss, CIOU loss, and classification loss. The calculation process is shown in Formulas (1)–(4). Among them, $\hat{C}_i = Pr(Object) \times CIoU_{pred}^{truth}$.

$$Loss = Loss_{bounding\_box} + Loss_{ciou} + Loss_{classification} \tag{1}$$

$$Loss_{bounding\_box} = \lambda_{coord} \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{obj} (2 - w_i \times h_i)[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{obj} (2 - w_i \times h_i)[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] \quad (2)$$

$$Loss_{ciou} = \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{obj} [\hat{C}_i log(C_i) + (1 - \hat{C}_i log(1 - C_i)] + \lambda_{noobj} \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{noobj} [\hat{C}_i log(C_i) + (1 - \hat{C}_i log(1 - C_i)] \quad (3)$$

$$Loss_{classification} = \sum_{i=0}^{K \times K} I_{ij}^{obj} \sum_{c \in classes} [\hat{p}_i(c) log(p_i(c)) + (1 - \hat{p}_i(c) log(1 - p_i(c))] \quad (4)$$

In the training process, pred_bbox was divided into positive examples and negative examples. For any ground truth, we calculated the IoU with all pred_bboxes, and the largest IoU was a positive example. One pred_bbox could only be assigned to one ground truth. For instance, if the first ground truth matched the pred_bbox, the next ground truth needed to find the largest IoU among the remaining pred_bboxes, as a positive example. If the IoU with all ground truth was less than the threshold, it was a negative example. All prediction boxes that were neither positive nor negative were discarded.

In this way, the loss function could reduce the weights of easy-to-classify samples so that the model could focus more on difficult-to-classify samples during training. Through this improvement, the accuracy of the network could be promoted while the inference speed of the network was maintained.

It can be inferred from Formula (5) that if the two prediction boxes do not intersect, their IoU value is 0. Then, this value could not reflect the distance between the two, that is, the degree of coincidence. At the same time, the corresponding loss is 0, and the gradient of back propagation is 0, and learning and training could not be performed. In the CVPR2019 paper [29], GIoU is proposed, and its calculation is shown in Formula (6), where A_c represents the smallest rectangular area that contains both the prediction frame and ground truth. From the formula, it could be inferred that when the prediction frame completely covers the ground truth, GIoU could not well reflect the coincidence of the two. In order to consider the distance and overlap rate at the same time, DIoU [30] is proposed, and its calculation process is shown in Formula (7), where $\boldsymbol{b}$ and $\boldsymbol{b}^{gt}$ represent the center points of the prediction frame and ground truth, respectively, p represents the Euclidean distance between these two center points, and c represents the diagonal distance of the smallest rectangle that could simultaneously contain the prediction frame and ground truth. However, due to the fact that the expression method does not consider the aspect ratio of the outer frame, on the basis of DIoU, CloU is proposed [30], which is the measurement method used in the loss function in this paper, and the penalty term is shown in Formula (8). Where α is the weight function, and ν, defined as $\nu = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2$, is used to measure the similarity of the aspect ratio. The gradient of CIoU loss is similar to DIoU, and when the length and width are in $[0,1]$, the value of $w^2 + h^2$ is usually very small, which results in an explosion of the gradient. Thus, when it came to the implementation, $\frac{1}{w^2+h^2}$ was replaced with 1. The loss function of CIoU is defined in Formula (9).

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (5)$$

$$GIoU = IoU - \frac{|A_c - U|}{|A_c|} \quad (6)$$

$$DIoU = IoU - \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^2} + \alpha\nu \quad (7)$$

$$\mathcal{R}_{CIoU} = \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b^{gt}})}{c^2} + \alpha v \tag{8}$$

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b^{gt}})}{c^2} + \alpha v \tag{9}$$

4. Results

4.1. Training

4.1.1. Warm-Up

Warm-up [28] is a training concept. In the pre-training stage, first use a low learning rate to train some epochs or steps, such as four epochs or 10,000 steps, and then move to a preset learning rate for training.

In this article, *exp* warm-up was tested; that is, the learning rate linearly increased from a small value to the preset learning rate, and then decayed according to the *exp* function law. Meanwhile, *sin* warm-up was also tested; that is, the learning rate increased linearly from a very small value, and after reaching the preset value, it decayed according to the *sin* function law.

4.1.2. Label-Smoothing

In this study, the backbone network would output a confidence score that the current data corresponded to the foreground—wheat. These scores were normalized by the *softmax* function, and ultimately the probability that the current data belongs to each category was obtained. The calculation is shown in Formula (10).

$$q_i = \frac{exp(z_i)}{\sum_{j=1}^{K} exp(z_j)} \tag{10}$$

Then, calculate the cross-entropy cost function:

$$Loss = -\sum_{i=1}^{K} p_i log q_i \tag{11}$$

The calculation method for p_i is shown in Formula (12).

$$p_i = \begin{cases} 1, \; if(i = y) \\ 0, \; if(i \neq y) \end{cases} \tag{12}$$

For the loss function, the predicted probability was supposed to be adopted to fit the true probability. However, fitting the one-hot true probability function would bring about a problem: The generalization ability of the model could not be guaranteed, and it would be likely to result in over-fitting.

Based on this, the regularization method of label smoothing was used to solve this problem. After label smoothing process, the probability distribution changed from Formulas (12) to (13):

$$p_i = \begin{cases} 1 - \epsilon, \; i = y \\ \dfrac{\epsilon}{K-1}, \; i \neq y \end{cases} \tag{13}$$

4.1.3. Pseudo-Label

Since the number of datasets was insufficient, the pseudo-label method was adopted to make full use of the verification data to enhance the training process. Three pseudo-label methods were tested, as shown in Figure 7. Among them, M represents a supervised model trained with labeled data, and M' represents a model trained with labeled data and pseudo-labeled data. PL-B used M', replaced M, and repeated until the model did not improve. PL-C replaced the loss function with what is shown in Formula (14).

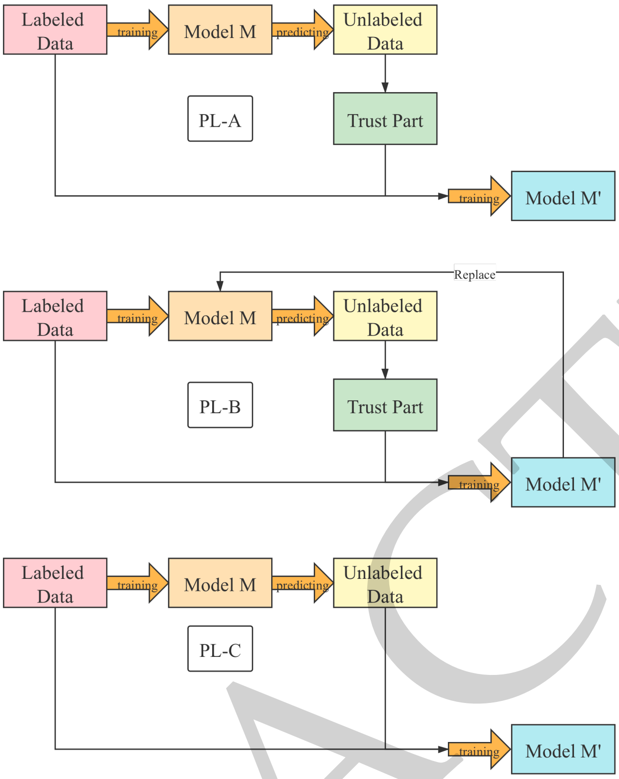


Figure 7. Flow chart of different pseudo-label models.

$$Loss = (1 - \alpha) \times Loss(labeled\_data) + \alpha \times Loss(unlabeled\_data) \tag{14}$$

4.2. *Test Time Augmentation*

4.2.1. Out of Fold

After the object detection model generated the prediction bounding boxes, the bounding boxes below the confidence threshold were discarded before the Non-Maximum Suppression (NMS) [31] algorithm was used. However, the setting of this threshold usually depends on experience. The core idea of Out of Fold (OoF) is to calculate the mAP of the validation set by traversing different thresholds and obtain the optimal threshold with the highest score of mAP during the traversal process. The process is shown in Figure 8.

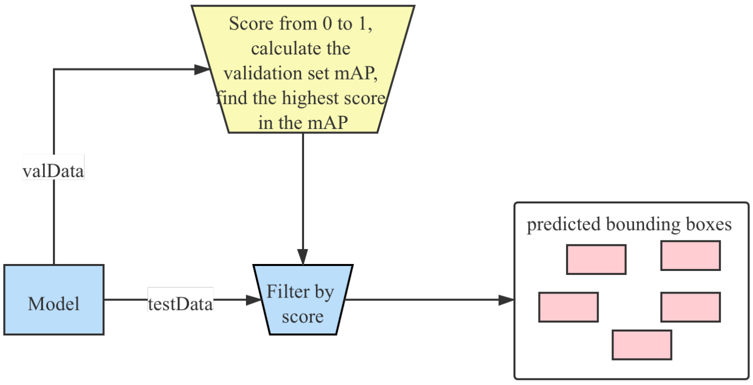

Score from 0 to 1, calculate the validation set mAP, find the highest score in the mAP
valData
Model
testData
Filter by score
predicted bounding boxes

Figure 8. The flow chart of the OoF method.

4.2.2. Optimization for NMS

In classic object detection algorithms, in order to improve the recall rate, many anchors are generated in the anchor generation stage. This results in many redundant bounding boxes corresponding to the same object in subsequent processing. Therefore, NMS is an indispensable step to remove redundant bound boxes in post-processing. However, NMS possesses the following problems:

1. When objects overlap, there will be a frame with the highest score. When several objects overlap, there will be a bounding box with the highest score. In this case, if NMS is used, bounding boxes representing other objects whose confidences are lower and the overlap with a bounding box with a higher score will be deleted.
2. Sometimes all the bounding boxes around an object are marked, but they are inaccurate.
3. The NMS method is based on the confidence score, so only the prediction bounding box with the highest score can remain. Nevertheless, in most cases, the IoU and the classification score are not strongly correlated, and many boxes with high confidences for classification labels are not highly accurate.

Based on the above analysis, soft NMS [32] and Weighted Boxes Fusion (WBS) [33] were adopted in the training phase to replace the traditional NMS method. The core of introducing soft NMS lies in the fact that it would not directly remove redundant outer boxes due to an NMS threshold. Instead, the highly redundant detection results were suppressed by a penalty function so that their scores were reduced. The higher the IoU coincidence degree was, the lower the score was. WBF integrated the outer boxes whose IoU were higher than the set threshold to get new outer boxes, which in turn contributed to reducing the final number of outer boxes. his method was adopted in multi-model fusion.

4.2.3. Model Ensemble

The method of model fusion was used to improve mAP. The method of fusion is to take the intersection of the results identified by different network models, fuse the learning capabilities of each model, prevent false detection, improve accuracy, and improve the generalization ability of the final model.

4.3. Experiment Results

The hardware platform for this experiment was: intel i9 CPU, NVIDIA RTX 3080 10 G graphics card, 16 G memory. The software platform was based on Python 3.9 and PyTorch 1.9.

4.3.1. Yolo and SSD Transfer Learning

Transfer learning means to transfer knowledge from one domain (i.e., the source domain) to another domain (i.e., the target domain) so that the target domain can achieve better learning results. Generally, the amount of data in the source domain is sufficient, whereas the amount of data in the target domain is relatively small. Transfer learning needs to transfer the knowledge learned under the condition of sufficient data to a new environment with a small amount of data. The principle is shown in Figure 9.

The YOLO series and SSD were first proposed in 2016–2017 and have been widely applied in agricultural image annotation [34–37]. In this study, the YOLO series and SSD were used for migration learning. We used the parameters obtained after training based on the VOC dataset to initialize the model, and then used the aforementioned method to complete the transfer process.

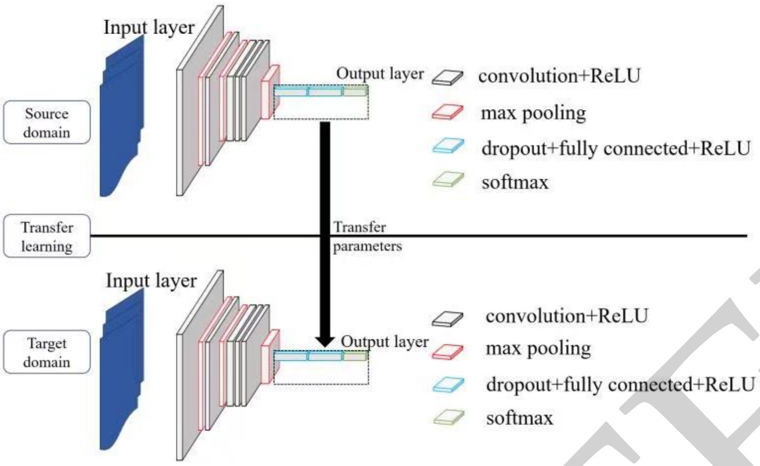


Figure 9. Transfer learning schematic.

4.3.2. Experiment Results

In order to verify the effectiveness of the model proposed in this paper, it was compared with one-stage EfficientDet, YOLO series with SDD, and mainstream two-stage network models. The results are shown in Table 1.

Table 1. A comparison of other models and our model.

Method	mAP	FPS	Batch Size	Input Resolution
FasterRCNN	0.8396	17	2	600 × 600
MaskRCNN	0.8493	19	2	600 × 600
EfficientDet	0.8520	37	8	512 × 512
YOLOv3	0.880	23	2	608 × 608
YOLOv4	0.838	47	2	608 × 608
YOLOv5	0.867	51	2	608 × 608
SSD300	0.846	35	2	300 × 300
SSD300	0.846	32	8	300 × 300
SSD512	0.847	19	2	512 × 512
SSD512	0.847	21	8	512 × 512
WDT512	0.882	41	2	512 × 512
WDT512	0.903	37	8	512 × 512
WDT1024	0.875	29	2	1024 × 1024

5. Discussion

5.1. Ablation Experiments

In order to verify the effectiveness of the various pre-processing techniques proposed in this article, such as various data augmentation methods, many ablation experiments were performed on both WDT512, the model input of which was 512 × 512, and WDT1024, the model input of which was 1024 × 1024. The experimental results are shown in Tables 2 and 3.

Table 2. Ablation experiment results on WDT512.

Cutout	Cutmix	Mosaic	Warm-Up	Label-Smoothing	Pseudo Label	mAP
			✓	✓	PL-A	0.5020
✓	✓	✓	✓	✓	PL-C	0.903
✓		✓	✓	✓	PL-A	0.873
✓	✓	✓	✓	✓	PL-B	0.877
✓	✓		✓	✓	PL-C	0.870
✓		✓	✓		PL-C	0.888
✓		✓		✓	PL-C	0.895

Table 3. Ablation experiment results on WDT1024.

Cutout	Cutmix	Mosaic	Warm-Up	Label-Smoothing	Pseudo Label	mAP
✓	✓	✓	✓	✓	PL-C	0.887
✓		✓	✓	✓	PL-A	0.903
✓	✓	✓	✓	✓	PL-B	0.894
✓		✓	✓		PL-B	0.871
✓		✓		✓	PL-C	0.880

It was found that the data augmentation methods such as cutout, cutmix, and mosaic are of great assistance to improving the performance of the model. The principles of cutmix and mosaic are similar. It could also be concluded that compared with adopting the combination of those two methods, using cutmix or mosaic alone exerts a more significant effect on the improvement of model performance. It could be seen that the model functions best when warm-up, label-smoothing, and PL-A-type pseudo-label methods are used as well.

In order to test the effectiveness of test time augmentation, a large number of ablation experiments were performed. The experimental results are shown in Table 4.

Table 4. Ablation experiment results of TTA.

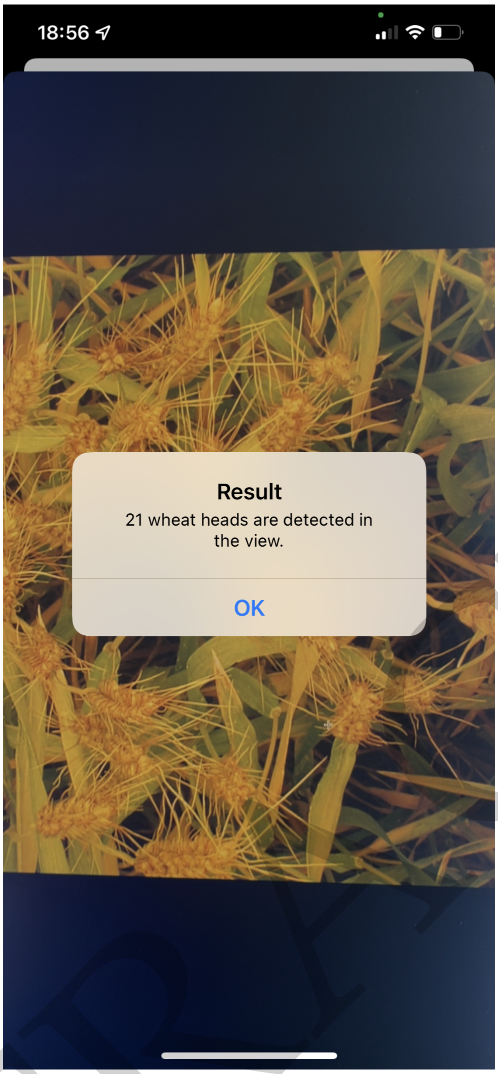
Models	OoF	NMS Method	mAP
WDT512		NMS	0.891
WDT1024		NMS	0.879
WDT512	✓	soft NMS	0.903
WDT512 + YOLO series + MaskRCNN	✓	WBF	0.917

According to the results, the performance of the model using OOF, WBF, and WDT512 + YOLO series + MaskRCNN for fusion was the best.

5.2. Intelligent Wheat Detection System

In order to realize the end-to-end model of wheat detection and promote the efficiency of recognizing and labeling, we developed an intelligent diagnosis system for iOS using the WDN model, with the development language Swift, and the development tool Xcode.

First, we retrieve the video input stream from an iOS device; then, we extract the representative frame and send it to the server; next, the server transfers the received images to the trained model; finally, the output of the model is returned to the iOS side, and the iOS side draws a detection frame based on the returned parameters. The annotation interface is shown in Figure 10.



Result

21 wheat heads are detected in the view.

Figure 10. Screenshot of the intelligent wheat detection system on an iOS device.

6. Conclusions

In this paper, the Wheat Detection Network based on the mainstream single-stage object detection network model was constructed to achieve the goal of rapid wheat detection. The performance of the model was improved by adding the attention mechanism and multi-scale feature fusion module and optimizing the activation function. In order to make full use of the training dataset, data enhancement methods such as cutout, cutmix, and mosaic, and technical methods such as label smoothing and pseudo-label, were adopted. Additionally, test time augmentation, OoF, WBF, model fusion, etc., were used to get the most out of the model.

In order to verify the effectiveness of the model, comparative experiments and ablation experiments were performed. The results indicate that the WDN inference time could reach 25 ms. As for the problem of wheat head detection, the network model proposed in this paper could even increase mAP to 0.903.

Eventually, in order to enable the network model proposed in this article to be applied in the agricultural production environment, our team built a set of intelligent systems, including front and back ends based on Swift and PHP, to cover the usage scenarios involving iOS mobile devices.

Author Contributions: Conceptualization, P.S.; methodology, P.S.; validation, P.S., X.H.; formal analysis, P.S. writing—original draft preparation, P.S.; writing—review and editing, P.S. and J.C.;

visualization, P.S.; funding acquisition, Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: Funder National Natural Science Foundation of China Grant Number 61202479.

Acknowledgments: We are grateful to the ECC of CIEE in China Agricultural University for their strong support during P.S's thesis writing.

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

References

1. Fischer, R. Wheat physiology: A review of recent developments. *Crop Pasture Sci.* **2011**, *62*, 95–114. [CrossRef]
2. Zhang, Y.; Wa, S.; Liu, Y.; Zhou, X.; Sun, P.; Ma, Q. High-Accuracy Detection of Maize Leaf Diseases CNN Based on Multi-Pathway Activation Function Module. *Remote Sens.* **2021**, *13*, 4218. [CrossRef]
3. Zhang, Y.; Wa, S.; Sun, P.; Wang, Y. Pear Defect Detection Method Based on ResNet and DCGAN. *Information* **2021**, *12*, 397. [CrossRef]
4. Zhang, Y.; He, S.; Wa, S.; Zong, Z.; Liu, Y. Using Generative Module and Pruning Inference for the Fast and Accurate Detection of Apple Flower in Natural Environments. *Information* **2021**, *12*, 495. [CrossRef]
5. Tang, L.; Gao, H.; Yoshihiro, H.; Koki, H.; Tetsuya, N.; Liu, T.S.; Tatsuhiko, S.; Zheng-Jin, X.U. Erect panicle super rice varieties enhance yield by harvest index advantages in high nitrogen and density conditions. *J. Integr. Agric.* **2017**, *16*, 1467–1473. [CrossRef]
6. Tan, Y.; Ouyang, C. Image recognition of rice diseases based on deep convolutional neural network. *J. Jinggangshan Univ. (Nat. Sci.)* **2019**, *40*, 38–45.
7. Allego, J.F.; Lootens, P.; Borralog, E.I.; Derycke, V.; Kefauver, S.C. Automatic wheat ear counting using machine learning based on RGB UAV imagery. *Plant J.* **2020**, *103*, 1603–1613.
8. Fernandez-Gallego, J.A.; Kefauver, S.C.; Gutiérrez, N.; Nieto-Taladriz, M.T.; Araus, J.L. Wheat ear counting in-field conditions: High throughput and low-cost approach using RGB images. *Plant Methods* **2018**, *14*, 22. [CrossRef] [PubMed]
9. Grbovi, E.; Pani, M.; Marko, O.; Brdar, S.; Crnojevi, V. Wheat Ear Detection in RGB and Thermal Images Using Deep Neural Networks. In Proceedings of the International Conference on Machine Learning and Data Mining, MLDM 2019, New York, NY, USA, 15 January 2019.
10. Liu, Z.-Y.; Sun, H.-S. Classification of Empty and Healthy Panicles in Rice Plants by Hyperspectral Reflectance Based on Learning Vector Quantization(LVQ)Neural Network. *Chin. J. Rice Sci.* **2007**, *21*, 664–668.
11. Zhou, C.; Ye, H.; Hu, J.; Shi, X.; Hua, S.; Yue, J.; Xu, Z.; Yang, G. Automated Counting of Rice Panicle by Applying Deep Learning Model to Images from Unmanned Aerial Vehicle Platform. *Sensors* **2019**, *19*, 3106. [CrossRef] [PubMed]
12. 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.
13. Uddin, S.; Mia, J.; Bijoy, H.I.; Raza, D.M. *Real Time Classification and Localization of Herb's Leaves Using*; Daffodil International University: Dhaka, Bangladesh, 2020.
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
15. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
17. Girshick, R. Fast r-cnn. In Proceedings of the IEEE international Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
20. 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 European Conference on Computer Vision, Munich, Germany, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
21. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
22. Sural, S.; Qian, G.; Pramanik, S. Segmentation and histogram generation using the HSV color space for image retrieval. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; Volume 2, p. 2.
23. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552

24. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6023–6032.
25. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430
26. Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 687–694. [CrossRef]
27. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 23–28 June 2018; pp. 7132–7141.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
29. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
30. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
31. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 3, pp. 850–855.
32. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS–improving object detection with one line of code. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5561–5569.
33. Solovyev, R.; Wang, W.; Gabruseva, T. Weighted boxes fusion: Ensembling boxes for object detection models. *arXiv* **2019**, arXiv:1910.13302
34. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. *Apple Detection during Different Growth Stages in Orchards Using the Improved YOLO-V3 Model*; Elsevier: Amsterdam, The Netherlands, 2019; Volume 157, pp. 417–426.
35. Morbekar, A.; Parihar, A.; Jadhav, R. Crop disease detection using YOLO. In Proceedings of the 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 5–7 June 2020; pp. 1–5.
36. Yuan, T.; Lv, L.; Zhang, F.; Fu, J.; Gao, J.; Zhang, J.; Li, W.; Zhang, C.; Zhang, W. Robust Cherry Tomatoes Detection Algorithm in Greenhouse Scene Based on SSD. *Agriculture* **2020**, *10*, 160.
37. Liang, Q.; Zhu, W.; Long, J.; Wang, Y.; Sun, W.; Wu, W. A real-time detection framework for on-tree mango based on SSD network. In Proceedings of the International Conference on Intelligent Robotics and Applications, Aachen, Germany, 6–8 December 2011; Springer: Berlin/Heidelberg, Germany, 2018; pp. 423–436.