agronomy

MDPI

Article

Tomato Maturity Recognition Model Based on Improved YOLOv5 in Greenhouse

Renzhi Li [1,2], Zijing Ji [1,2], Shikang Hu [3], Xiaodong Huang [1], Jiali Yang [4] and Wenfeng Li [2,3,*]

[1] College of Big Data, Yunnan Agricultural University, Kunming 650201, China
[2] Key Laboratory of Yunnan Provincial Department of Education for Crop Simulation and Intelligent Regulation, Kunming 650201, China
[3] College of Mechanical and Electrical Engineering, Yunnan Agricultural University, Kunming 650201, China
[4] College of Foreign Languages, Southwest Forestry University, Kunming 650224, China
* Correspondence: liwenfeng@ynau.edu.cn; Tel.: +86-158-8719-2600

Abstract: Due to the dense distribution of tomato fruit with similar morphologies and colors, it is difficult to recognize the maturity stages when the tomato fruit is harvested. In this study, a tomato maturity recognition model, YOLOv5s-tomato, is proposed based on improved YOLOv5 to recognize the four types of different tomato maturity stages: mature green, breaker, pink, and red. Tomato maturity datasets were established using tomato fruit images collected at different maturing stages in the greenhouse. The small-target detection performance of the model was improved by Mosaic data enhancement. Focus and Cross Stage Partial Network (CSPNet) were adopted to improve the speed of network training and reasoning. The Efficient IoU (EIoU) loss was used to replace the Complete IoU (CIoU) loss to optimize the regression process of the prediction box. Finally, the improved algorithm was compared with the original YOLOv5 algorithm on the tomato maturity dataset. The experiment results show that the YOLOv5s-tomato reaches a precision of 95.58% and the mean Average Precision (mAP) is 97.42%; they are improved by 0.11% and 0.66%, respectively, compared with the original YOLOv5s model. The per-image detection speed is 9.2 ms, and the size is 23.9 MB. The proposed YOLOv5s-tomato can effectively solve the problem of low recognition accuracy for occluded and small-target tomatoes, and it also can meet the accuracy and speed requirements of tomato maturity recognition in greenhouses, making it suitable for deployment on mobile agricultural devices to provide technical support for the precise operation of tomato-picking machines.

Keywords: tomato; maturity recognition; deep learning; improved YOLOv5; loss function

Citation: Li, R.; Ji, Z.; Hu, S.; Huang, X.; Yang, J.; Li, W. Tomato Maturity Recognition Model Based on Improved YOLOv5 in Greenhouse. *Agronomy* **2023**, *13*, 603. https://doi.org/10.3390/agronomy13020603

Academic Editors: Xiuliang Jin, Hao Yang, Zhenhai Li, Changping Huang and Dameng Yin

Received: 27 January 2023
Revised: 14 February 2023
Accepted: 17 February 2023
Published: 20 February 2023

1. Introduction

The tomato is one of the world's three most-traded vegetables, with a wide planting area and rich nutrition [1]. China ranks first in the world in terms of both production and scale of tomato cultivation, accounting for 35% of global tomato production. The tomato fruit is perishable. The rate of tomato fruit rot and loss reaches 20% to 30% in logistics links such as picking and storage. Tomato fruits in different maturity stages are stored together, which is the main cause of depletion and loss [2]. Ripeness is one of the most important indicators in determining the tomato picking time. Harvest maturity has an important impact on tomato composition and quality, which can determine tomato processing, transportation, sales methods, and shelf life. Prices vary from tomatoes at different maturity stages [3]. Tomatoes with different maturities are usually occluded and overlapped in greenhouses by each other in the same plant and population [4]. The accurate identification of the maturity of tomato fruit can not only prevent misharvesting and reduce costs, but it can also increase the utilization rate, which is needed to improve the automation and accuracy of tomato production management [5].

In recent years, scholars have conducted significant research on tomato maturity recognition. Including traditional recognition methods and deep-learning recognition methods.

The traditional methods usually apply the differences between the fruits and background in terms of color, texture, and shape to extract features through algorithms to achieve fruit recognition. Muhammad Hammad Malik et al. created the red tomato detection algorithm for natural light conditions using an improved HSV (Hue, Saturation, Value) and watershed detection localization method [6]. Guoxu Liu et al. trained a support vector machine (SVM) classifier using histograms of oriented gradient (HOG) descriptors and reduced the effect of different light levels on tomato identification in greenhouses [7]. Han Li et al. completed fruit recognition of green tomatoes when the background colors of leaves and stalks are similar to each other by the fusing fast normalized cross-correlation function (FNCC) and the Hough transform circle detection method (Hough) [8]. Li Liu et al. produced a color classifier for ripe, semi-ripe, and unripe tomatoes by distributing the chromaticity values of tomatoes with different ripeness. The average classification precision was 90.7% [9]. Tomato maturity recognition through algorithms is feasible, but the decision-making process of such algorithms is usually complicated. The fixed threshold value is difficult to adapt to environmental background and light condition changes. Magnetic resonance imaging, hyperspectral imaging, and other techniques have also been used for the identification of tomato ripeness. Lu Zhang et al. applied magnetic resonance imaging (MRI) to classify tomato fruit maturity stages by the partial least squares discriminant analysis model (PLS-DA) [10]. Yiping Jiang et al. adopted hyperspectral imaging technology to establish a sparse representation model of class probabilistic information (CSR) and divided tomato fruits into four maturity stages [11]. Yuping Huang et al. adopted a 550–1650 nm spatially resolved spectroscopic system to recognize tomato fruit at six maturity stages, with the precision of this method being 98.3% [12]. However, these methods are time-inefficient in identification and have high equipment costs. Therefore, they cannot meet the needs of greenhouse harvesting recognition.

Deep learning is a research hotspot in the field of agricultural harvesting and has been extensively studied in fruit maturity recognition research. A deep-learning model indicates the advantages of high recognition accuracy and strong generalization ability. Mahmoud Ali Alajrami et al. proposed a tomato classification model based on a convolutional neural network (CNN) to detect tomato varieties [13]. Jiehua Long et al. proposed a tomato fruit recognition method based on an improved Mask R-CNN model to achieve tomato fruit maturity recognition in greenhouses [14]. Guoxu Liu proposed the improved YOLO-Tomato recognition model based on YOLOv3 to achieve the correct recognition of yellow tomatoes under mild occlusion conditions [15]. Tianhua Li et al. proposed a recognition method combining YOLOv4 and HSV to achieve the recognition of mature-stage tomatoes in greenhouses [16]. Yuhao Ge et al. proposed a YOLO–DeepSort model based on the Yolov5s. The model realized the identification and counting of tomatoes at different growth stages, including flowers, green tomatoes, and red tomatoes [17]. Compared with traditional methods, deep learning avoids complex steps such as manual feature extraction. In addition, its recognition accuracy and robustness appear stronger. However, the existing deep-learning methods still present some disadvantages, such as low recognition accuracy, a complex network structure, more parameters, a slow running speed, and the requirement of powerful GPU computing power for real-time detection, which is difficult to deploy in an actual picking equipment. Therefore, it is important to develop a model to recognize tomato fruit maturity in greenhouse conditions. The YOLOv5 series target detection algorithm is representative of one-stage target detection algorithms. Whether its application can improve the accuracy, stability, and real-time performance of tomato maturity recognition for tomatoes under overlapping and occlusion conditions in complex environments, and whether the maturity can be effectively distinguished, remains to be studied further.

In summary, to further improve tomato fruit recognition accuracy and speed in complex backgrounds, in this study, a tomato maturity recognition model YOLOv5s-tomato based on improved YOLOv5 is proposed. Depending on the characteristics of tomato fruits in greenhouses, using the advantages of fast running speed, high recognition accuracy, and small memory cost of YOLOv5s in target detection, and using Mosaic data enhancement

and other methods, the tomato maturity recognition accuracy for the tomato fruits in the state of occlusion and small targets was improved. The CIoU Loss in the original network was replaced by EIoU Loss to improve the ability of extracting the tomato's maturity characteristics. Finally, it achieved the tomato fruit maturity recognition result in greenhouses. Furthermore, by comparing with the recognition effects of YOLOv5s-tomato, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and Faster RCNN, the real-time performance and accuracy of the improved model can be verified, and it can serve as a reference for the accurate operation of tomato-picking machines.

2. Materials and Methods

2.1. Data Acquisition

In this study, the tomato variety with red fruits at maturity stage was selected as the research object. The experiment was conducted in the greenhouse of the vegetable base of Yunnan Agricultural University (102°75′ N, 25°13′ E), Kunming City, Yunnan Province, China. The tomato variety is named "Yun kang 21." Image acquisition device is a Nikon Z6 camera equipped with Sigma 24 mm prime lens. The camera is set to aperture priority mode with aperture F 1.4, resolution 6048 × 4024 pixels. The image format is JPEG. The images were taken at a distance of 0.5–1 m away from tomato fruits and collected from January 2021 to October 2021. The tomato fruit image collection environment is shown in Figure 1.

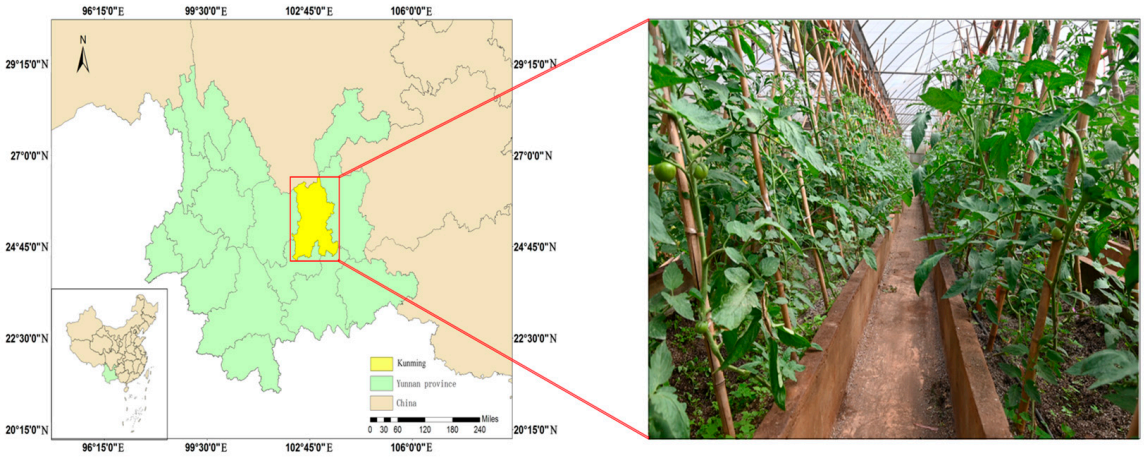

Figure 1. Tomato fruit image collection plot.

Considering the influence of different lighting conditions on tomato fruit color, experiments were conducted under different lighting conditions, such as downlight, sidelight, and backlight, and in different weather conditions, such as sunny and cloudy. Images were taken at different times in the morning, at noon, and in the evening. In addition, considering factors such as the tomato fruits occluding and overlapping with each other, the experiment images were taken from different angles to increase the variability of input images.

Tomato maturity levels are divided into four stages: mature green, breaker, pink, and red, with the reference to the Chinese industry standard GH/T 1193-2021—Tomato [18]. The mature green stage is defined as the period when the fruit is set, the fruit surface changes from green to white-green and shiny, and the seed has grown completely with gelatinous surroundings. The breaker stage is defined as the period when the tomato fruit is mostly green or yellow, or when reddish haloes begin to appear around the umbilicus and the flesh begins to soften. The pink stage is defined as the period when the fruit umbilicus turns red, with most of the area (about 3/4) appearing orange. The red stage is defined as the period when the fruit surface and flesh turn completely red and maintain a certain hardness. The tomatoes in the four different maturity stages are shown in Figure 2.

(**a**) (**b**) (**c**) (**d**)

Figure 2. Different maturity-stage tomato samples. (**a**) mature green; (**b**) breaker; (**c**) pink; (**d**) red.

2.2. Dataset Construction and Pre-processing

2.2.1. Sample Enhancement

To enrich the datasets to extract the features of the images better, the image data enhancement is required. In this experiment, sample enhancement was performed by flipping, randomly rotating, and adjusting the brightness, saturation, and contrast ratios [19]. The effect of sample enhancement is shown in Figure 3.

(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 3. Sample enhancement effect plots: (**a**) original image; (**b**) flip/rotate; (**c**) brightness adjustment; (**d**) saturation adjustment; (**e**) contrast adjustment.

2.2.2. Dataset Construction

According to the maturity classification criteria, more than 6000 collected samples were classified into different maturity stages in the experiment. The number of samples at the mature green stage, breaker stage, pink stage, and red maturity stage are 2480, 635, 1280, and 1572, respectively. The dataset was manually annotated using the image visualization annotation tool "Labeling." The dataset images and labels are divided into a training set and a testing set. The training set comprises 80%, while the testing set comprises 20%. The verification set comprises 25% of the testing set for cross-validation in model training.

2.3. Tomato Maturity Recognition Methods

2.3.1. YOLOv5 Target Detection Network

YOLO series of target detection algorithms has been iterated and updated continuously. Considering the model size and stability, the YOlOv5 was selected as the recognition model in the condition of recognition accuracy close [20]. YOLOv5 improves the network structure and training skills of the YOLOv3 algorithm. Its detection performance is equivalent to that of YOLOv4, yet the model size is reduced by nearly 90% [21,22]. YOLOv5 contains four network structures: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, of which YOLOv5s has the smallest network structure, the fastest speed, and the lowest accuracy. The other three networks deepen and widen the network, with accuracy increasing accordingly yet speed also slowing down [23]. The overall network structure of the YOLOv5s target detection algorithm is shown in Figure 4. YOLOv5s consists of four components: input, backbone, neck, and prediction [24,25].

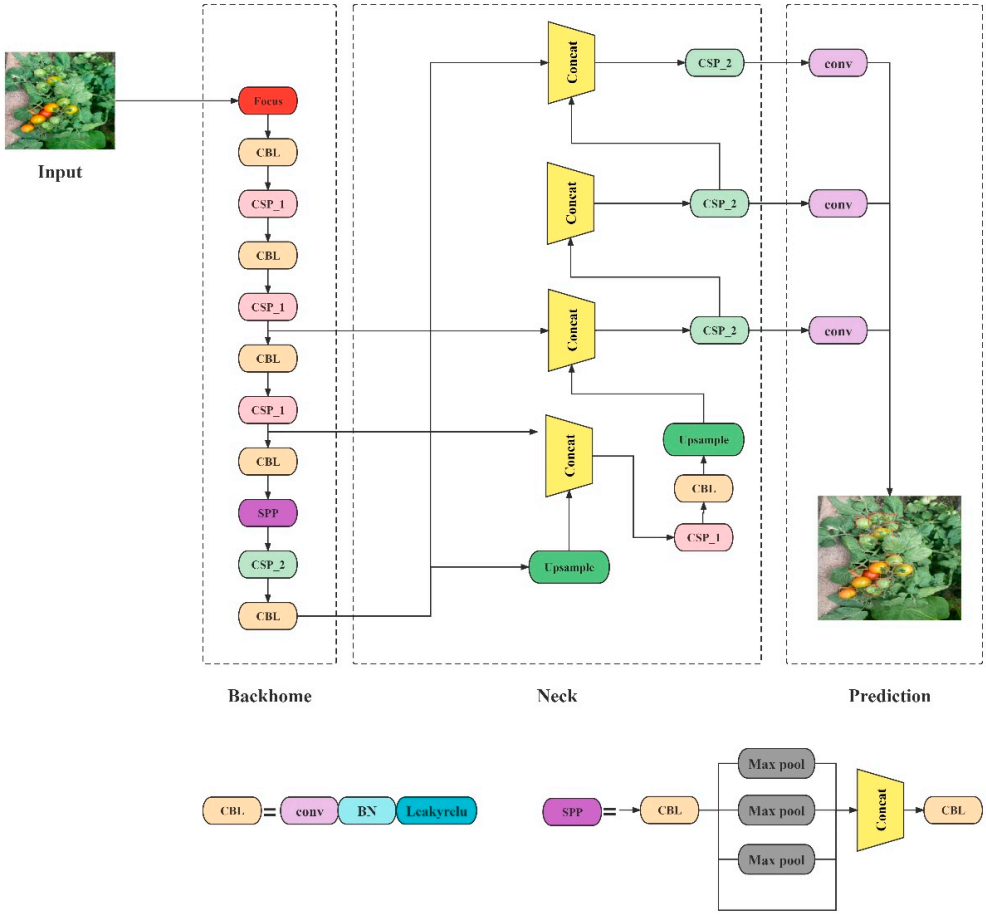

Figure 4. YOLOv5 network model structure diagram.

In the above YOLOv5 network model structure diagram, "conv" represents convolution, "BN" represents batch normalization, and "concat" represents the feature fusion method of adding up the number of channels. "CBL" represents a synthesis module that includes "conv," "BN," and "Leaky relu," which represents the activation functions.

Tomato fruit have the characteristics of dense distribution, small size, and similar morphology and color, which seriously affects the accuracy of target recognition. In this experiment, the Mosaic data enhancement method was applied to train the model to improve recognition ability for small targets in overlapping and occlusion conditions [26]. Four non-duplicate images were extracted from the dataset using this method. The process of flipping, scaling, changing brightness, and changing saturation was carried out for each image, respectively. The four images processed above were recombined, and gray filling

was carried out in the end shown in Figure 5. Random cropping can enrich the local target features and image background of the dataset. Random scaling can create a large number of target images so that the robustness of the model is increased and the small-target detection performance is enhanced [27].

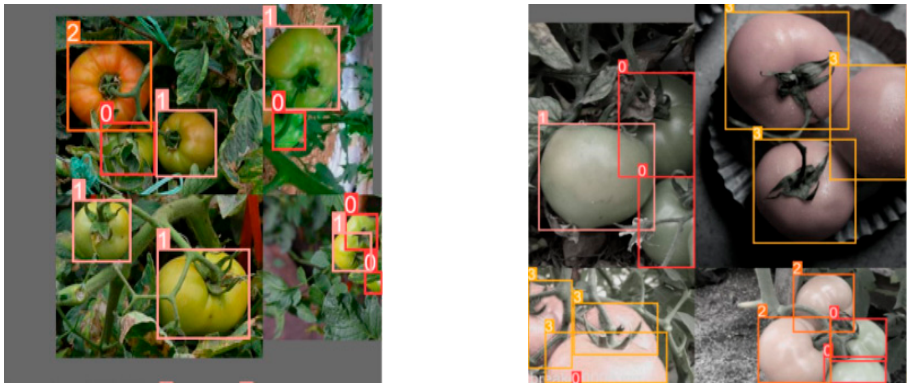

Figure 5. Mosaic data enhancement processed images.

The adaptive anchor is the network output prediction box based on the initial box. The reverse update was done by calculating the difference between the prediction box and ground truth box. Finally, the anchor box size that best fits the dataset is obtained. During model training, the network calculates the optimal anchor box in different training sets adaptively [28]. The anchor boxes can be defined by the aspect ratio and the bounding box size. The bounding box calculation formula is shown in Equation (1):

$$\begin{cases} w \times h = s \\ \frac{w}{h} = ratio \end{cases} \qquad \begin{cases} w = ratio \times h \\ ratio \times h^2 = s \end{cases} \tag{1}$$

In the calculation formula, "w" means the width of the bounding box, "h" means its height, and "s" means the area of the anchor. The adaptive anchor cooperates well with the network training, improving the accuracy of the model, reducing the amount of computation, and improving the speed of target detection [29].

The Focus module is utilized in the first layer of the backbone network. The key step is the slicing operation. In the YOLOv5s algorithm, an ordinary image sized 3 × 608 × 608 is input into the network. It is then converted into a feature map sized 12 × 304 × 304 after the slicing operation and finally converted into a feature map sized 32 × 304 × 304 after a convolution operation with 32 convolution kernels shown in Figure 6. The main purpose of this algorithm is to improve the detection speed [30].

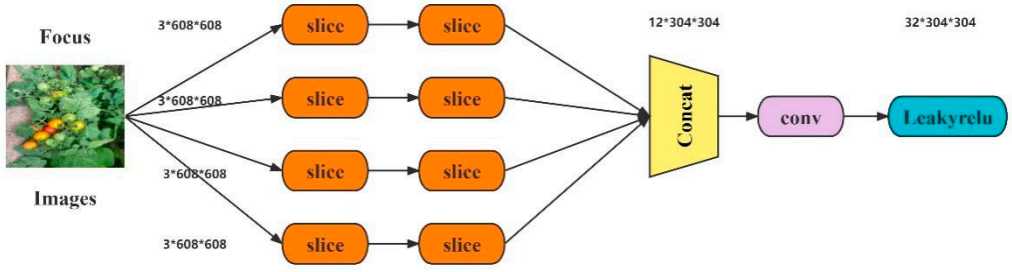

Figure 6. Focus module schematic diagram.

There are two CSPNet in the YOLOv5s network architecture, namely CSP1 and CSP2, shown in Figure 7. The CSP1 is applied in the backbone network, and the CSP2 is applied in the neck network. The main purpose of the structure is to divide the feature maps into two parts. One part is applied to continue the convolution operation to obtain more profound characteristic information. The other part is combined with the feature maps that were

convolutionally operated upon in the first part. The advantages of the cross-stage design are enhancing the network learning capability, allowing the network to maintain higher accuracy, and reducing the number of parameters. It also can improve the inference speed and reduce the memory cost [31]. Res unit borrows the residuals structure in the ResNet network to build the deep network.

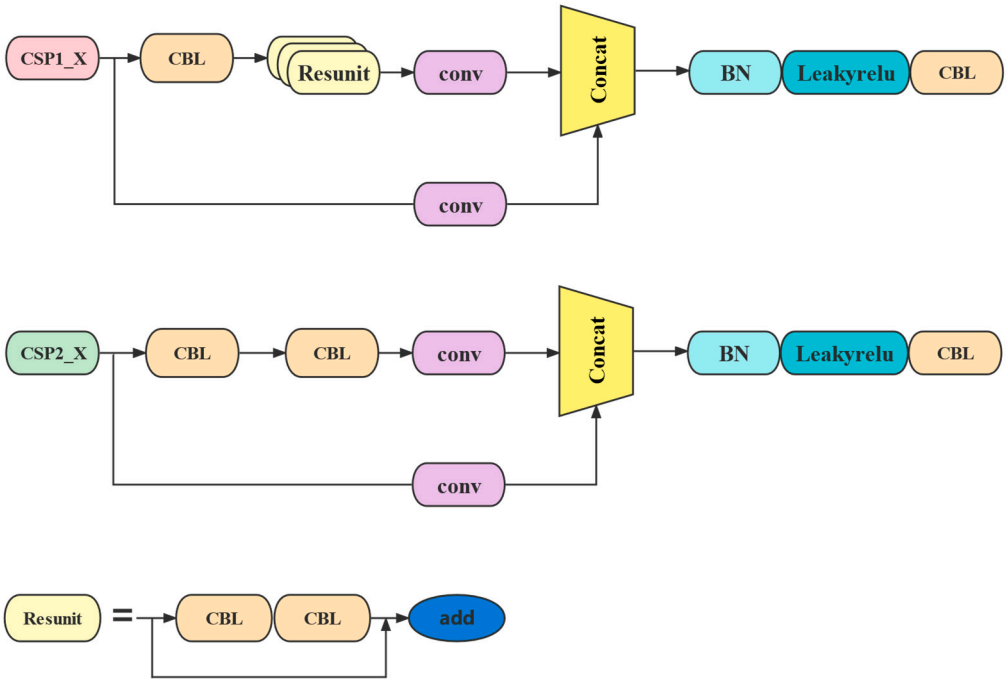

Figure 7. CSP module schematic.

2.3.2. Improved Loss Function

The quality of the loss function directly affects the training speed and detection performance of the model. The CIoU Loss was adopted to calculate the localization loss in the initial YOLOv5, in which the overlapping area of bounding box regression, the distance of the center points, and the side-length aspect ratio were all added as penalty factors, based on IoU (Intersection Over Union) [32]. The CIoU Loss is defined as follows:

$$\mathrm{IoU} = \frac{\mathrm{area}(\mathrm{ar} \cap \mathrm{tr})}{\mathrm{area}(\mathrm{ar} \cup \mathrm{tr})} \tag{2}$$

$$\alpha = \frac{\nu}{(1 - \mathrm{IoU}) + \nu} \tag{3}$$

$$L_{\mathrm{CIoU}} = 1 - \mathrm{IoU} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha\nu \tag{4}$$

where "ar" and "tr" represent the anchor box and the bounding box, "b" and $b^{gt,}$ respectively, represent the center point between ground truth box and the prediction box. ρ represents the Euclidean distance between the two center points. c represents the diagonal length of the minimum enclosing rectangle of the anchor and bounding box. α represents the weight function. ν represents the similarity of the aspect ratio of the two boxes.

Although, in the CIoU Loss, the distance of the center points between ground truth box and the prediction box, as well as the two boxes' ratio of width-to-height, was considered, the width-to-height ratio description is relative and cannot be accurately located. To solve this problem, the EIoU Loss was introduced as a bounding-box loss function in this study [33]. The width-to-height ratio loss in the CIoU Loss was replaced by the EIoU Loss with the width and height loss of the minimum enclosing rectangle, which accelerated the

convergence rate of the loss function and improved the regression accuracy. The calculation formula is shown in Equation (5)

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(\omega, \omega^{gt})}{C_\omega^2} + \frac{\rho^2(h, h^{gt})}{C_h^2} \quad (5)$$

where C_ω and C_h mean the width and height of the smallest enclosing rectangle covering the two boxes. In bounding-box regression loss, the loss of width and height in EIoU Loss makes the convergence speed faster and the precision higher. EIoU Loss is better than CIoU Loss in the original network. Therefore, EIoU Loss with better performance was adopted as the loss function in this study.

2.3.3. Experimental Comparison Models

To further verify the recognition performance of YOLOv5s-tomato, in this experiment, YOLOv5s-tomato, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and the mainstream two-stage target detection algorithm Faster R-CNN were compared with each other. The same training and validation sets were used to train the models, and the same testing sets were utilized for evaluation. All the training and testing experiments were conducted on the same testbed. Finally, precision, recall, mean average precision, detection time, and model size were calculated to compare the test results [34].

2.4. Experiment Platform and Parameter Setting

2.4.1. Experiment Platform

All model training and testing were carried out with the same computer with the following specifications: hardware configuration, AMD 3700x processor, 16 GB running memory, graphics card, GeForce GTX 2070 super GPU, and Windows 10 operating system. The deep-learning frameworks Pytorch and Tensor Flow were utilized to adapt to different network training requirements. In addition, all the models were trained with the pre-training weight files provided by the developers.

2.4.2. Experiment Parameter

In this experiment, the hyperparameter optimization method was hyperparametric evolution. A new offspring with a combination of the best parents from all previous generations were created by using a 90% probability and 0.04 variance mutates [35,36]. Through this method, more suitable hyperparameters for this experiment were obtained. A visualization of the results of hyperparameter evolution after 200 iterations is shown in Figure 8.

The pre-training model was applied to train the YOLOv5s-tomato, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and Faster R-CNN, respectively. After the training was completed, the training model was saved and the test set was used to verify the model. After a comparison of the preset hyperparameters provided by the platform and the training results of the hyperparameter evolution, the hyperparameter settings in this study are shown in Table 1.

Table 1. Hyperparameter settings.

Hyperparameter	Value	Hyperparameter	Value	Hyperparameter	Value
lr0	0.00902	cls	0.486	hsv_s	0.529
lrf	0.183	cls_pw	1.03	hsv_v	0.344
momentum	0.98	obj	0.421	translate	0.102
weight_decay	0.00039	obj_pw	0.824	scale	0.308
warmup_epochs	2.86	iou_t	0.2	fliplr	0.5
warmup_momentum	0.899	anchor_t	6.99	mosaic	1
warmup_bias_lr	0.112	anchors	2	Epochs	200
box	0.0378	hsv_h	0.0186	Batch Size	12

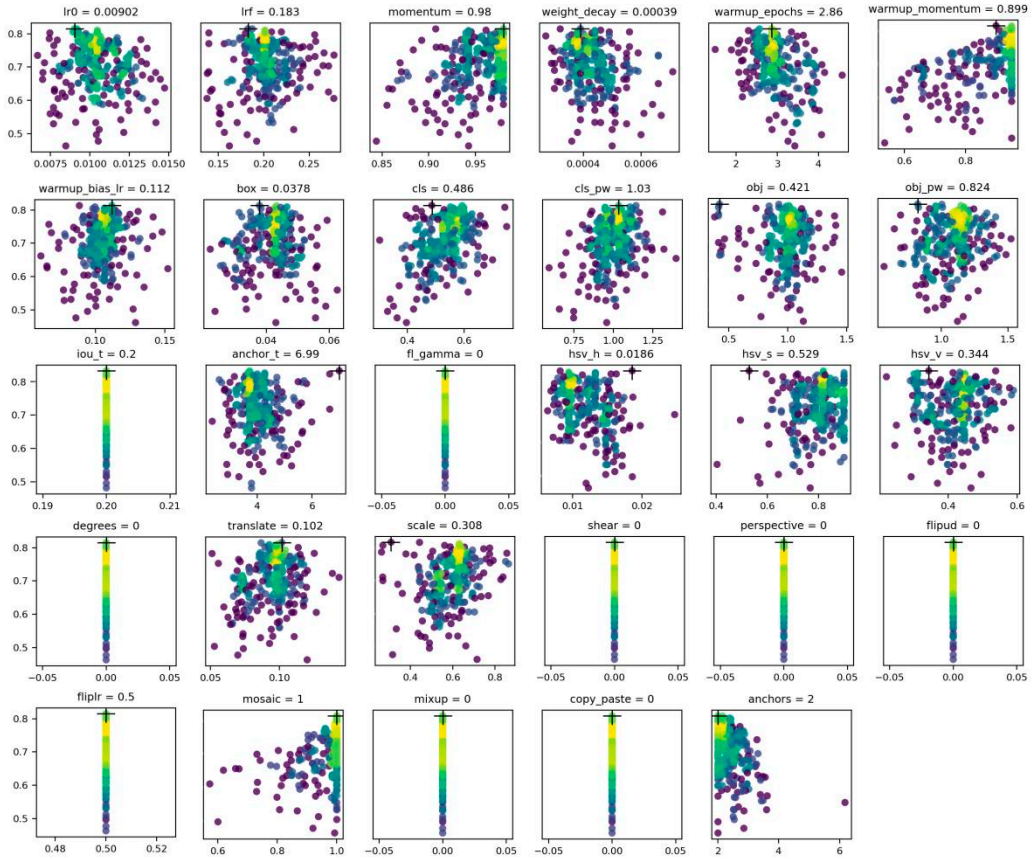

Figure 8. Visualizing hyperparameter evolution.

2.5. Evaluation Indices

The performance evaluation indices applied in this experiment are precision, recall, mean average precision, detection speed and model size. The calculations for each evaluation index are shown in Equations (6)–(9) [37]:

$$P = \frac{TP}{TP + FP} \times 100\% \tag{6}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{7}$$

$$AP = \int_0^1 P(R)dR \times 100\% \tag{8}$$

$$MAP = \frac{1}{n}\sum_{i=1}^{n} AP \times 100\% \tag{9}$$

where TP (true positive) means positive samples with positive predictions. TN (true negative) means negative samples with negative predictions. FP (false positive) means negative samples with positive predictions, which means incorrect detections. FN (false negative) means positive samples with negative predictions. N means the number of categories. Precision is measured in terms of prediction results and refers to the number of predicted positive samples that are actually positive samples. Recall refers to how many positive samples are recognized out of the total positive samples. AP (average precision) means the area under the P–R curve for a single category. This index can comprehensively indicate the precision and recall of a model. Generally, the higher the AP value is, the better the model performance. MAP is the average AP value obtained in all categories, reflecting the overall detection accuracy of the model and serving as the most important performance evaluation index. Detection speed and model size determine whether the model can be

applied to an automatic picking machine in greenhouse. The detection time is the average time that the model consumes to detect a single image. The unit is "ms" here.

3. Results and Analysis

3.1. Different Maturity Recognition Results

The P–R curve is a curve in which the precision is plotted on the longitudinal axis and the recall is plotted on the horizontal axis. The P–R curve reflects the comprehensive performance of a target detection network. The P–R curves of YOLOv5s-tomato for four maturity stages in the testing set are shown in Figure 9. The model achieved good results in the recognition of each maturity stage. It also achieved a good detection precision while maintaining a high recall. The precision of the model in mature green stage and red stage was higher than that of in breaker stage and pink stage.

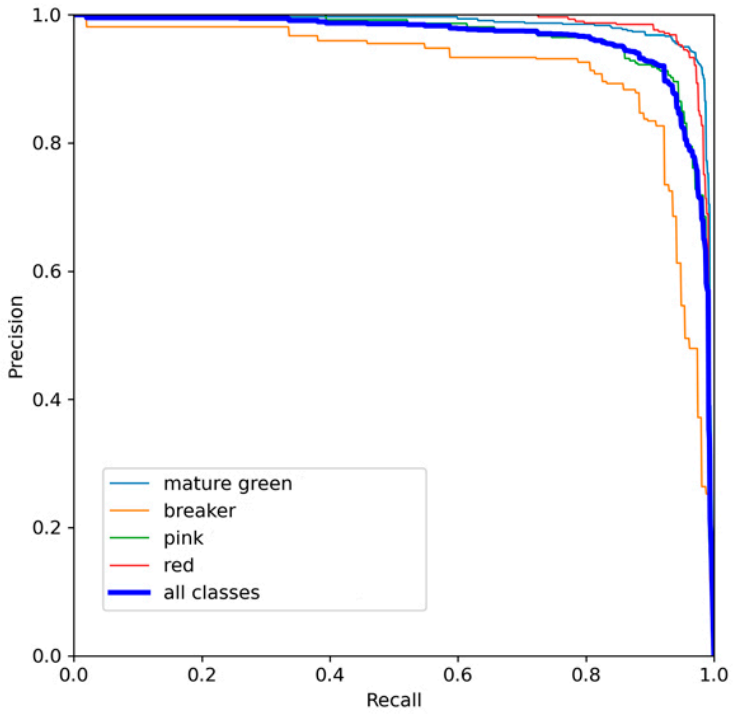

Figure 9. The YOLOv5s-tomato P–R curves.

The YOLOv5s-tomato confusion matrix is shown in Figure 10. The recall of the YOLOv5s-tomato model at the mature green, breaker, pink, and red stages were 97%, 90%, 92%, and 94%, respectively. Recognition errors mainly occurred in the following situations: About 6% of the breaker-stage tomatoes were misrecognized as being in the mature green stage, and 3% were misrecognized as being in the pink stage; about 6% of the pink-stage tomatoes were misrecognized as being in the breaker stage; about 2% were misrecognized as being in the red stage; and about 4% of the red stage were misrecognized as being in the pink stage. The confusion matrix indicates missed detections occurred. It can be seen that the missed detections in four maturity stages were all lower than 2% of the total amount of detections. The last column in the confusion matrix indicates false detections occurred. The reason for the false detections is that the model recognized some small, unlabeled tomato fruits in the dataset.

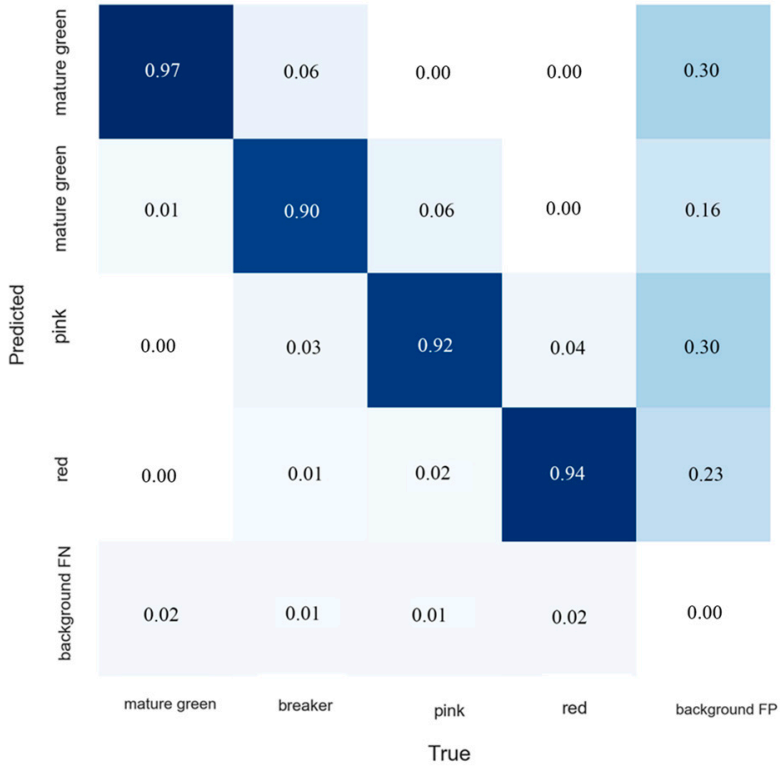

Figure 10. YOLOv5s-tomato prediction results confusion matrix.

3.2. Different Model Detection Performance Comparison

In order to fully verify the validity and robustness of the model for tomato maturity detection, 200 rounds of training were conducted on the model. The precision and loss function were adopted to judge whether the training trend is correct so as to ensure that the model can converge normally with the increase of the number of iterations. It can be seen from the precision curve image and loss function curve image of the model shown in Figure 11 that the training convergence of the model is in normal. The different model training-loss function curves indicate as the number of iterations increases the network loss value gradually decreases and tends to be stable. The loss value of the model decreases rapidly at 0 to 50 iterations, and it decreases slow down at 50 to 200 iterations. After 200 iterations, the loss value tends to be stable near 0.03, then the model reaches the optimal state.

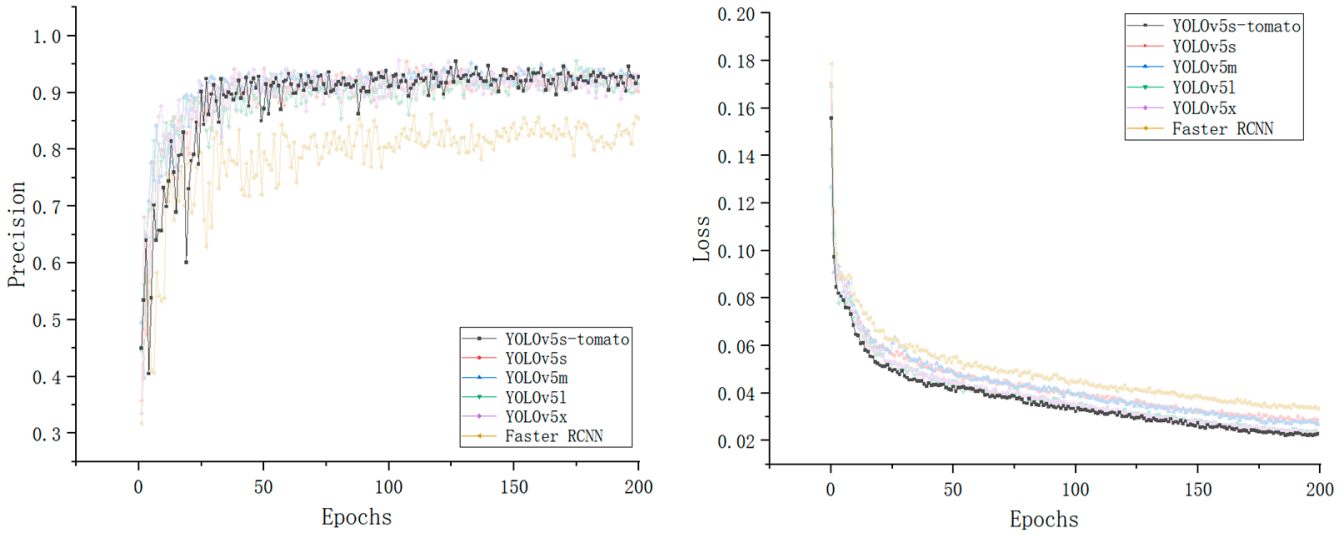

Figure 11. Variation curve of Model precision curve and Loss value.

After the models were tested completely, the four network structures of YOLOv5s-tomato and YOLOv5 generally achieved excellent performance, displayed in Table 2. Their precisions all were over 95%, recalls all were around 90%, and mean-average precisions all were over 96%. Faster RCNN obtained the lowest mAP, and its detection effect is not ideal. The detection performance of the YOLOv5s-tomato is significantly higher than that of Faster RCNN, YOLOv5s, YOLOv5m, and YOLOv5l, and slightly lower than that of the YOLOv5x. In terms of precision, recall, and mean average precision, the results of the YOLOv5s-tomato are 0.11%, 0.33%, and 0.09% lower than those of YOLOv5x with the best detection results, respectively. Yet the YOLOv5s-tomato model size and per-image detection time are only 8.88% and 34.06% that of YOLOv5x. The comparison results show that the YOLOv5s-tomato achieved higher recognition precision, faster detection speed, and smaller model size, compared with the other five models.

Table 2. The tomato maturity recognition results comparison.

Models	Precision P (%)	Recall R (%)	Mean Average Precision (%)	Test Time (ms)	Memory (MB)
YOLOv5s-tomato	95.58	90.07	97.42	9.2	23.9
YOLOv5s	95.47	89.19	96.76	9.2	23.9
YOLOv5m	95.51	91	96.94	12.2	68
YOLOv5l	95.56	90.84	97.33	16.9	146
YOLOv5x	95.69	90.4	97.51	27.3	269
Faster RCNN	86.2	80.7	83.42	13.7	116

In the test, the maturity detection was conducted under the shade, occlusion, and density conditions. The detections of each model are shown in Figure 12.

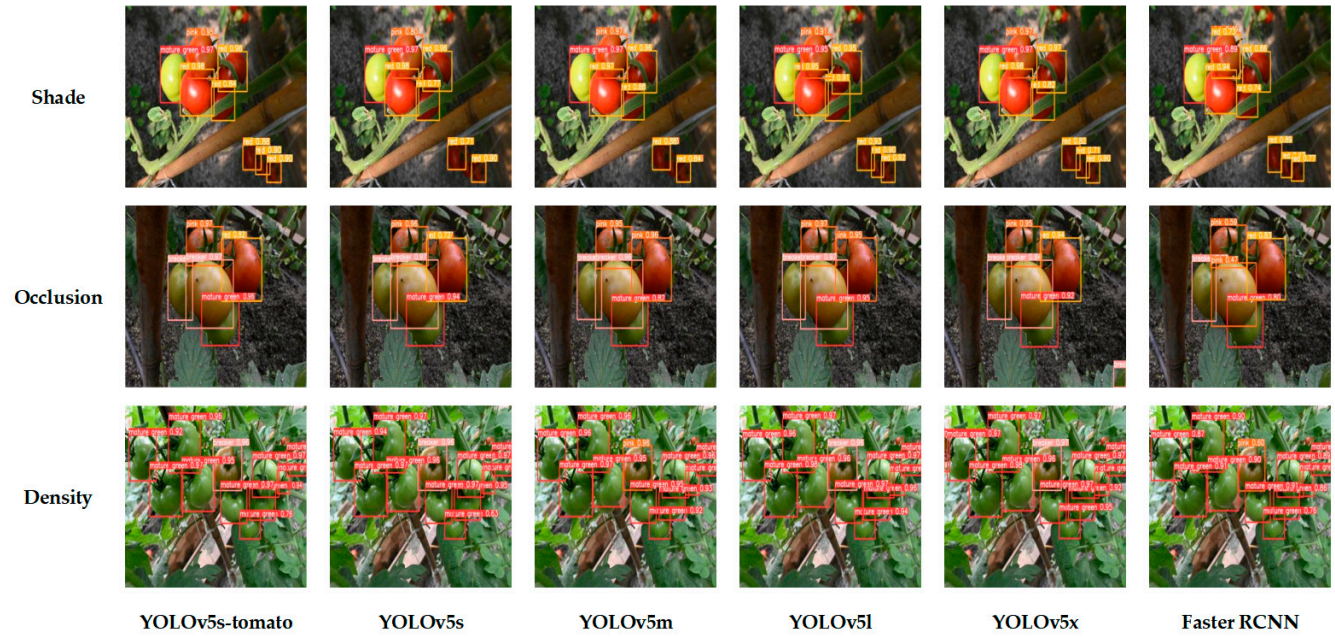

Figure 12. The recognition effects comparison of models.

In this experiment, the small-target tomato is selected to test. The tomato maturity recognition results under shade and small-target states are shown in Figure 12. The result shows that the YOLOv5s-tomato model obtained the highest maturity recognition accuracy. YOLOv5x and YOLOv5l can also recognize tomato fruits' maturity in shade and small-target states, and the recognition accuracies of theirs are slightly lower than that of the YOLOv5s-tomato for the small-target tomato. Yet the YOLOv5s and YOLOv5m occurred

miss-recognition on the tomato fruits in the shade and occlusion states. The Faster RCNN occurred false-recognition on some tomato fruits with lower recognition accuracy.

In actual greenhouses, tomato fruits often yield a lot and occlude with each other. YOLOv5x obtained the highest maturity recognition accuracy for tomato fruits in mutual occlusion state. YOLOv5x also achieved maturity recognition on tomatoes locating at the edge of the image, even with only a small portion of them appearing in the image. The YOLOv5s-tomato and YOLOv5s also achieve tomato fruit maturity recognition under the occlusion state. Incorrect recognition occurred with YOLOv5l, YOLOv5m, and Faster RCNN for some tomato fruits in breaker and pink stages.

The tomatoes often are in states of dense distribution, which influences the tomato maturity recognition accuracy in certain degrees. YOLOv5x obtained the highest maturity recognition accuracy on the tomato fruit in dense distribution state. The YOLOV5s-tomato, as well as YOLOv5l and YOLOv5s, also achieved the tomato fruit maturity recognition under the dense distribution condition. The YOLOv5m and Faster RCNN obtained the lower recognition accuracy and occurred incorrect recognition.

4. Discussion

The deep-learning model indicates the advantages of being not needed to manually select features, the good generalization performance, etc. In addition, with the increase of the amount of data, the accuracy of the deep-learning model will become more accurate. The existing deep-learning model weight file appears too large, and the real-time detection speed appears slow. The recognition accuracy to the small-target and in occlusion tomato fruits appears not high. By comparison, the tomato maturity recognition model YOLOv5s-tomato proposed in this study based on the improved YOLOv5 can recognize four different tomato maturities in greenhouse. Compared with similar studies, in this experiment, tomato maturity recognition is divided into four maturity levels, which is conducive to tomato picking with different maturity requirements in actual production. Compared with the laboratory recognition models, the proposed model in this study obtained the features of faster recognition speed, smaller model size, and lower hardware configuration requirements, relatively. Compared with other recognition models in greenhouses, the proposed model obtained the characteristics of higher recognition precision and reduced model size, which was reduced by about 80% of that of other models in greenhouses. Furthermore, the tomato fruit-detection effect on occlusion and the small target of the proposed model is better than those of others. After the comparative analysis shown in Table 3, the YOLOv5s-tomato model obtains advantages for embedded deployment or use in mobile equipment for greenhouse tomato picking.

Table 3. The comparison of proposed method with existing others.

Source	Method	Environment	Maturity Level	mAP	Speed
Li Liu [9]	HSV	Laboratory	Three maturity levels	90.70%	68.7 ms
Yuping Huang [12]	SR-SVMDA	Laboratory	Six maturity levels	98.30%	
Jiehua Long [14]	Improved Mask R-CNN	Greenhouse	Three maturity levels	95.45%	658 ms
Guoxu Liu [15]	YOLOv3	Greenhouse	All tomatoes	96.40%	54 ms
Fei Su [38]	SE-YOLOv3-Mobile	Greenhouse	Four maturity levels	87.70%	227.1 ms
Tianhua Li [16]	YOLO v4+HSV	Greenhouse	Ripe tomato	94.77%	22.18 ms
Yuhao Ge [17]	YOLO-Deep-Sort	Greenhouse	Three maturity levels	95.8%	
Proposed method	YOLOv5s-tomato	Greenhouse	Four maturity levels	97.42	9.2 ms

In the experiment, recognition errors in the stages of tomato maturity mainly occurred when the breaker-stage fruits were misrecognized as the mature green and the pink stage were misrecognized as the breaker stage. The main reasons for the misrecognition mostly are that during the ripening process, the tomato maturity changes are continuous. The transformation between the breaker stage and the pink stage appears very quick, with less data being collected. Due to the characteristics of the tomato itself, although there are some

difficulties in recognizing the breaker stage and the pink stage, which don't influence the tomato maturity recognition much.

The research on tomato maturity recognition in this experiment is just the beginning. There are still some limitations in practical applications. Firstly, the model YOLOv5s-tomato proposed in the experiment obtains good applicability for tomato varieties with red fruits when maturing. For tomato varieties with other color fruits at maturing stage, the data relabeling and retraining should be carried out when applying this model. Secondly, in this experiment, the images used in the training were collected from the same greenhouse and compared with greenhouse images, yet the images collected from outdoors indicate a more complex background and more variable lighting conditions. Therefore, the tomato maturity recognition performance of the YOLOv5s-tomato model for the outdoor tomato still needs to be verified further. Finally, the deep-learning algorithms themselves require a lot of data to improve the maturity recognition accuracy, robustness, and generalization ability of the model YOLOv5s-tomato. More images will be obtained from different kinds of greenhouses and outdoor areas to supplement the data diversity in future studies.

5. Conclusions

The tomato maturity recognition model YOLOv5s-tomato based on improved YOLOv5 is proposed in this experiment. Based on the characteristics of tomato fruit maturity stages, the tomato maturity dataset on greenhouse tomatoes was established. The small-target detection performance of the model was improved by enhancement methods, such as Mosaic data enhancement, etc. In the prediction period, the EIoU Loss was adopted to replace the initial CIoU Loss, which effectively reduced the differences between the ground truth box and the prediction box. The experiment results show that the tomato maturity recognition precision of the YOLOv5s-tomato is 95.58%, the recall is 90.07%, the mAP is 97.42%, the model size is only 23.9 MB, and the detection time per image is only 9.2 ms.

Considering the characteristics of tomatoes in greenhouses where the tomato branches, leaves, and fruits occlude with each other and have relatively dense distribution, the algorithms of YOLOv5s-tomato, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, and Faster RCNN were compared with each other in this experiment, respectively. The experimental results show that the models mentioned above all could achieve the maturity recognition of the tomato fruits in shade, occluded, and dense distribution, of which the model YOLOv5s-tomato and YOLOv5x obtained the best recognition effect and Faster RCNN obtained a worse recognition effect. The precision, recall, and mean average precision of the model YOLOv5s-tomato are 0.11%, 0.33%, and 0.09% lower than that of model YOLOv5x, respectively. However, the YOLOv5s-tomato model size and per-image detection time are only 8.88% and 34.06% that of YOLOv5x.

Compared with the traditional methods, the model YOLOv5s-tomato demonstrates strong robustness and could recognize the tomato fruits' maturity under the conditions of branches, leaves, and fruits occlusion accurately. Compared with the deep-learning model, YOLOv5s-tomato has a higher precision and has obvious advantages in recognition speed and model size. Considering the limitations and recognition efficiency of greenhouse mechanical picking equipment, the YOLOv5s-tomato algorithm is considered as a suitable model for tomato picking machines in greenhouses.

Author Contributions: Conceptualization, R.L., S.H. and W.L.; Methodology, R.L. and W.L.; Software, R.L. and X.H.; Validation, R.L. and S.H.; Formal analysis, J.Y.; Investigation, R.L., Z.J. and X.H.; Resources, Z.J. and W.L.; Data curation, R.L. and Z.J.; Writing—original draft, R.L.; Writing—review & editing, R.L., J.Y. and W.L.; Supervision, J.Y.; Project administration, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Maize Growth simulation and yield prediction based on Data Assimilation of Plant Phenotypic, the National Natural Science Foundation of China, Project number: 32160420. This research was also partially supported by Major Science and Technology Special Projects in Yunnan Province, Project number: 202202AE09002103, and Yunnan Province lnnovative Team Projects, Project number: 2020tdxmy14.

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

Acknowledgments: Thanks to all partners in Key Laboratory of Yunnan Provincial Department of Education for Crop Simulation and Intelligent Regulation, for their support.

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

References

1. Costa, J.; Miguel; Heuvelink, E.P. The global tomato industry. In *Tomatoes*; CABI: Wallingford, UK, 2018; pp. 1–26.
2. Ghezavati, V.R.; Hooshyar, S.; Tavakkoli-Moghaddam, R. A Benders' decomposition algorithm for optimizing distribution of perishable products considering postharvest biological behavior in agri-food supply chain: A case study of tomato. *Cent. Eur. J. Oper. Res.* **2017**, *25*, 29–54. [CrossRef]
3. Zhang, L.; Jia, J.; Gui, G.; Hao, X.; Gao, W.; Wang, M. Deep learning based improved classification system for designing tomato harvesting robot. *IEEE Access* **2018**, *6*, 67940–67950. [CrossRef]
4. Fenn; Matthew, A.; James, J.; Giovannoni. Phytohormones in fruit development and maturation. *Plant J.* **2020**, *105*, 446–458. [CrossRef]
5. Teka; Tilahun, A. Analysis of the effect of maturity stage on the postharvest biochemical quality characteristics of tomato (*Lycopersicon esculentum* Mill.) fruit. *Int. Res. J. Pharm. Appl. Sci.* **2013**, *3*, 180–186.
6. Malik, M.H.; Zhang, T.; Li, H.; Zhang, M.; Shabbir, S.; Saeed, A. Mature tomato fruit detection algorithm based on improved HSV and watershed algorithm. *IFAC-PaperOnLine* **2018**, *51*, 431–436. [CrossRef]
7. Liu, G.; Mao, S.; Kim, J.H. A mature-tomato detection algorithm using machine learning and color analysis. *Sensors* **2019**, *19*, 2023. [CrossRef] [PubMed]
8. Li, H.; Zhang, M.; Gao, Y.; Li, M.; Ji, Y. Green ripe tomato detection method based on machine vision in greenhouse. *Trans. Chin. Soc. Agric. Eng.* **2017**, *33*, 328–334.
9. Liu, L.; Li, Z.; Lan, Y.; Shi, Y.; Cui, Y. Design of a tomato classifier based on machine vision. *PLoS ONE* **2019**, *14*, e0219803. [CrossRef] [PubMed]
10. Zhang, L.; McCarthy, M.J. Measurement and evaluation of tomato maturity using magnetic resonance imaging. *Postharvest Biol. Technol.* **2012**, *67*, 37–43. [CrossRef]
11. Jiang, Y.; Chen, S.; Bian, B.; Li, Y.; Wang, X. Discrimination of tomato maturity using hyperspectral imaging combined with graph-based semi-supervised method considering class probability information. *Food Anal. Methods* **2021**, *14*, 968–983. [CrossRef]
12. Huang, Y.; Si, W.; Chen, K.; Sun, Y. Assessment of tomato maturity in different layers by spatially resolved spectroscopy. *Sensors* **2020**, *20*, 7229. [CrossRef] [PubMed]
13. Alajrami, M.A.; Abu-Naser, S.S. Abu-Naser. Type of tomato classification using deep learning. *Int. J. Acad. Pedagog. Res. (IJAPR)* **2020**, *3*, 21–25.
14. Long, J.H.; Zhao, C.J.; Lin, S.; Guo, W.Z.; Wen, C.W.; Zhang, Y. Segmentation method of the tomato fruits with different maturities under greenhouse environment based on improved Mask R-CNN. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 100–108.
15. Liu, G.; Nouaze, J.C.; Touko Mbouembe, P.L.; Kim, J.H. YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. *Sensors* **2020**, *20*, 2145. [CrossRef] [PubMed]
16. Li, T.H.; Sun, M.; Ding, X.; Li, Y.; Zhang, G.; Shi, G.; Li, W. Tomato recognition method at the ripening stage based on YOLO v4 and HSV. *Trans. Chin. Soc. Agric. Eng. (Trans. CSAE)* **2021**, *37*, 183–190.
17. Ge, Y.; Lin, S.; Zhang, Y.; Li, Z.; Cheng, H.; Dong, J.; Shao, S.; Zhang, J.; Qi, X.; Wu, Z. Tracking and Counting of Tomato at Different Growth Period Using an Improving YOLO-Deepsort Network for Inspection Robot. *Machines* **2022**, *10*, 489. [CrossRef]
18. *GH/T 1193-2021*; Tomato. Standards Press of China: Beijing, China, 2021.
19. Nagaraju, M.; Chawla, P.; Kumar, N. Performance improvement of Deep Learning Models using image augmentation techniques. *Multimedia Tools Appl.* **2022**, *81*, 9177–9200. [CrossRef]
20. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* **2016**, 779–788. Available online: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html (accessed on 26 January 2023).
22. Song, Q.; Li, S.; Bai, Q.; Yang, J.; Zhang, X.; Li, Z.; Duan, Z. Object detection method for grasping robot based on improved YOLOv5. *Micromachines* **2021**, *12*, 1273. [CrossRef]
23. Zhu, X. Design of Barcode Recognition System Based on YOLOV5. Journal of Physics: Conference Series. *IOP Publ.* **2021**, *1995*, 012052.
24. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. *Proc. IEEE/CVF Int. Conf. Comput. Vis.* **2021**, 2778–2788.
25. Xing, L.; Fan, X.; Dong, Y.; Xiong, Z.; Xing, L.; Yang, Y.; Bai, H.; Zhou, C. Multi-UAV cooperative system for search and rescue based on YOLOv5. *Int. J. Disaster Risk Reduct.* **2022**, *76*, 102972. [CrossRef]
26. Hao, W.; Zhili, S. Improved mosaic: Algorithms for more complex images. *J. Phys. Conf. Series. IOP Publ.* **2020**, *1684*, 012094. [CrossRef]

27. Yao, J.; Qi, J.; Zhang, J.; Shao, H.; Yang, J. A real-time detection algorithm for Kiwifruit defects based on YOLOv5. *Electronics* **2021**, *10*, 1711. [CrossRef]
28. Thuan, D. Evolution of Yolo Algorithm and Yolov5: The State-of-the-Art Object Detention Algorithm. Available online: https://urn.fi/URN:NBN:fi:amk-202103042892 (accessed on 1 January 2021).
29. Zhao, Y.; Shi, Y.; Wang, Z. The Improved YOLOV5 Algorithm and Its Application in Small Target Detection. In Proceedings of the Intelligent Robotics and Applications: 15th International Conference, ICIRA 2022, Harbin, China, 1–3 August 2022; Proceedings, Part IV. Springer International Publishing: Cham, Switzerland, 2022.
30. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs. *Sensors* **2022**, *22*, 464. [CrossRef]
31. Wang, M.; Zhu, Y.; Liu, Y.; Deng, H. X-Ray Small Target Security Inspection Based on TB-YOLOv5. *Secur. Commun. Netw.* **2022**, *2022*, 1–16. [CrossRef]
32. Wang, Q.; Ma, Y.; Zhao, K.; Tian, Y. A comprehensive survey of loss functions in machine learning. *Ann. Data Sci.* **2020**, *9*, 187–212. [CrossRef]
33. Zhang, Y.F.; Ren, W.; Zhang, Z. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
34. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 1995; pp. 1440–1448.
35. Mantau, A.J.; Widayat, I.W.; Adhitya, Y.; Prakosa, S.W.; Leu, J.S.; Köppen, M. A GA-Based Learning Strategy Applied to YOLOv5 for Human Object Detection in UAV Surveillance System. In *2022 IEEE 17th International Conference on Control & Automation (ICCA)*; IEEE: Piscataway, NJ, USA, 2022; pp. 9–14.
36. Zhang, X.; Feng, Y.; Zhang, S.; Wang, N.; Mei, S. Finding Nonrigid Tiny Person with Densely Cropped and Local Attention Object Detector Networks in Low-Altitude Aerial Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 4371–4385. [CrossRef]
37. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*; IEEE: Piscataway, NJ, USA, 2020; pp. 237–242.
38. Su, F.; Zhao, Y.; Wang, G.; Liu, P.; Yan, Y.; Zu, L. Tomato Maturity Classification Based on SE-YOLOv3-MobileNetV1 Network under Nature Greenhouse Environment. *Agronomy* **2022**, *12*, 1638. [CrossRef]