International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 39, No. 1, February, 2022, pp. 291-298
Journal homepage: http://iieta.org/journals/ts

Recognition and Detection of Greenhouse Tomatoes in Complex Environment

Guohua Gao[1], Shuangyou Wang[1,2*], Ciyin Shuai[1], Zihua Zhang[1], Shuo Zhang[1], Yongbing Feng[1]

[1] Faculty of Materials and Manufacturing, Beijing University of Technology, Beijing 100124, China
[2] School of Software, Handan University, Handan 056005, China

Corresponding Author Email: wsyhdc@emails.bjut.edu.cn

https://doi.org/10.18280/ts.390130

Received: 10 December 2021
Accepted: 24 January 2022

Keywords:
you only look once v5 (YOLO v5), recognition and detection, tomato, deep learning

ABSTRACT

In the complex environment of greenhouses, it is important to provide the picking robot with accurate information. For this purpose, this paper improves the recognition and detection method based on you only look once v5 (YOLO v5). Firstly, adding data enhancement boosts the network generalizability. On the input end, the k-means clustering (KMC) was utilized to obtain more suitable anchors, aiming to increase detection accuracy. Secondly, it enhanced multi-scale feature extraction by improving the spatial pyramid pooling (SPP). Finally, non-maximum suppression (NMS) was optimized to improve the accuracy of the network. Experimental results show that the improved YOLO v5 achieved a mean average precision (mAP) of 97.3%, a recall of 90.5%, and an F1-score of 92.0%, while the original YOLO v5 had a mAP of 95.9% and a recall of 85.6%; the improved YOLO v5 took 57ms to identify and detect each image. The recognition accuracy and speed of the improved YOLOv5 are much better than those of faster region-based convolutional neural network (Faster R-CNN) and YOLO v3. After that, the improved network was applied to identify and detect images take in unstructured environments with different illumination, branch/leave occlusions, and overlapping fruits. The results show that the improved network has a good robustness, providing stable and reliable information for the operation of tomato picking robots.

1. INTRODUCTION

China is the largest producer and consumer of tomatoes. Greenhouse grown tomatoes have been developing rapidly across the country. These tomatoes are mainly picked manually. The manual picking takes up around 50% of the total workload of growing tomato in greenhouses, and the labor cost reaches 50% to 70% of the income of the harvest. Moreover, the efficiency of manual picking is rather low. As the Chinese population ages, there is a growing shortage of agricultural labor for the high-labor intensity operation of picking greenhouse grown tomatoes [1]. In some cases, it is impossible to pick the tomatoes in time, resulting in loss of agricultural production. To realize modern, mechanized, and intelligent agriculture, the research and development of agricultural robots provides a solution to the above problem [2]. Replacing manual operators with tomato picking robots can reduce labor cost, adapt to the needs of the high-intensity operation, and improve labor productivity.

The difficulty of picking robot design lies in identifying and detecting tomato targets. The accuracy of recognition and detection bears on the work efficiency of picking robots [3]. In the natural environment, tomato fruits vary in pose, size, sparsity, and light condition, and might overlap each other. In many cases, the fruits are severely occluded by branches and leaves. All these factors bring difficulty to the recognition and positioning by picking robots. Many domestic and foreign scholars have explored the recognition and detection of tomatoes. However, most research and development results remain in the lab phase, rather enter the actual production. The recognition and detection techniques are not sufficiently stable or generalizable. Therefore, it is urgent to solve the technical problem of rapid and precise recognition and detection of tomato fruits in complex environments.

2. LITERATURE REVIEW

Computer vision-based object detection has been widely used to harvest robots across the world. In recent years, many detection methods are developed by scholars. Li et al. [4] captured images with a red-green-blue-depth (RGB-D) camera, preprocessed the images to obtain the fruit contours, separated the contours of overlapping fruits, and fitted them into circles. The k-means clustering (KMC) was combined with self-organizing map (SOM) neural network algorithm for tomato recognition. Experimental results show that their approach correctly recognized 87.2% of tomatoes. However, the contour extraction is affected by the illumination. Xiang et al. adopted iterative Otsu's method to segment the clustered regions, computed the depth difference between the front and rear regions, and treated them as overlapping regions if the different is greater than the threshold [5, 6]. Next, edge curvature analysis was caried out to identify the overlapping tomatoes. When the branch occlusion was less than 25%, their method could correctly recognize 87.9% of tomatoes. Yu et al. developed a machine vision deep learning algorithm called Mask-RCNN [7]. With ResNet50 as the backbone network, their algorithm selects the candidate boxes through region proposal network (RPN), and realizes automatic detection of strawberries. Their report indicates that Mask-RCNN achieved an average precision of 95.78% and a recall of 95.41% under different illumination and fruit occlusion. Each strawberry was identified by processing an average of 8 frames. Despite the high stability, their approach needs to be further improved in real-time performance. Through wavelet fusion analysis, Zhao et al. [8, 9] extracted the a*-component image and I-component feature map from the L*a*b and luma-in-phase-quadrature (YIQ) color space models, carried out pixel fusion of these images, and recognized tomatoes through optimal threshold segmentation. Experimental results show that their strategy can correctly segment and recognize 93% of tomatoes, and reduce the influence of illumination and branch/leave occlusion over segmentation in unstructured environments. To realize the automatic recognition of tomatoes, Yamamoto et al. [10] captured images with a traditional RGB camera, established a classification model based on the colors, shapes, textures, and sizes of the images, and automatically determined the optimal number of clusters through KMC. The experimental report suggests that their method achieved a recall of 80%, a precision of 0.88, and a recognition rate of 100%, 80% and 78% on mature, immature, and young fruits, respectively. Muhammad Hammad Malik et al. adopted the improved hue-saturation-value (HSV) color space algorithm to detect red tomatoes, and separated the tomatoes from the unevenly lighted and complex background, using the improved watershed segmentation algorithm. Experimental results show that their method recognized 81.6% of red tomatoes [11]. Hu et al. [12] segmented the tomato regions from the background with the H and S Gaussian density functions of the HSV color space, identified the edges of tomatoes through adaptive threshold intuitive fuzzy set (IFS), and improved the recognition of overlapping tomatoes with faster region-based convolutional neural network (Faster R-CNN). Experimental results show that the mean relative error (MRE) of horizontal and vertical displacements were 0.261% and 1.179%, respectively, indicating that their approach improves the detection accuracy of tomatoes. Liu et al. (2020) extended you only look once v3 (YOLO v3) into a YOLO-tomato model to solve problems of occlusion, overlapping, and illumination. Their algorithm introduces the densely connected network to optimize YOLOv3, and predict and position tomatoes with C-box. Experimental results show that their method recognized 94.58% of tomatoes under slight occlusion [13]. Using the relationship between fruit color difference components, Ma et al. employed the Otsu's method to obtain the complete binary image of each fruit, and marked the split lines of the binary image by an algorithm coupling improved ultimate corrosion and marker-controlled watershed segmentation. Experimental results show that their approach can correctly segment 96.5% of overlapping apples [14]. Taking images collected by stereo cameras as a dataset, Magalhaes S.A. et al. proposed deep learning models like sing-shot detector (SSD) and YOLO, and conducted training and benchmark testing of five deep learning models. The results show that SSD MobileNet v2 achieved the best performance with a mean average precision (mAP of 51.46% and an F1-score of 66.15. But the network performance should be further improved under branch occlusion [15]. Chen et al. captured lychee images with a binocular camera, and improved YOLO v3 into YOLO-DenseNet34 for detecting lychee strings. The lychee strings were matched under the constraint of sequential consistency in the same row. The mean precision and detection speed of their approach reached 94.3% and 22.11 frames per second (fps), respectively [16].

To sum up, there is a significant progress in tomato recognition and detection at home and abroad. Nevertheless, the accuracy and real-time performance need to be further improved. The traditional algorithm has great influence on light and occlusion, and can not recognize tomato accurately. With the rapid development of artificial intelligence, the deep learning approach of convolutional neural networks (CNNs) demonstrate better performance than traditional machine vision strategies [17-20]. For example, Faster-RCNN, a detector based on region proposal generation, boasts a low false recognition rate and a low false negative rate. However, this detector is too slow to meet actual demand. The weight parameters obtained by training are also larger. As a regression-based detector, the YOLO supports fast and real-time recognition, and its accuracy meets the requirements of field applications. Therefore, this paper proposes an improved YOLO v5 tomato recognition and detection algorithm, which overcomes the poor recognition accuracy, timeliness, and robustness of tomato fruits under overlapping fruits, branch occlusion, and uneven illumination in the complex natural environment of agricultural greenhouses.

3. MATERIALS AND METHODS

3.1 Image dataset

Our dataset comes from Yujiawu Tomato Greenhouse Base, Tongzhou District, Beijing. The images were collected by a binocular stereo camera (Sony IMX307). According to the structural layout of the greenhouse, the images were captured at 50-100cm. After segmentation, the left and right images each has 640*480 pixels, with black marginal areas. All images were calibrated by the binocular camera, and used for robotic picking. Figure 1 shows a collected image.

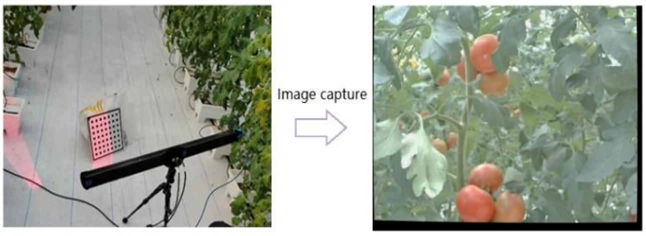
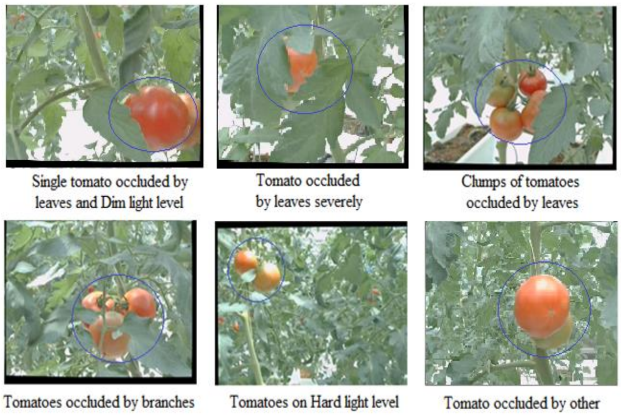


Figure 1. An image collected by the binocular camera


Single tomato occluded by leaves and Dim light level

Tomato occluded by leaves severely

Clumps of tomatoes occluded by leaves

Tomatoes occluded by branches

Tomatoes on Hard light level

Tomato occluded by other

Figure 2. Tomato dataset under natural environment

The images were shot in the natural environment of greenhouses. The complexity of tomato growth is well reflected in these images, e.g., the difference in size and illumination, the occlusion between tomatoes, and the occlusion by branches. A total of 1,000 tomato images were captured randomly under conditions like cloudy day, sunny day, sidelight, backlight, shade, etc. Some images are about a single tomato, and some are about a cluster of tomatoes. In some images, the tomatoes are occluded by other fruits and branches. In some other images, the tomatoes are directly under the sunlight. Figure 2 shows the established dataset of tomatoes.

3.2 Data enhancement

In the complicated natural environment of greenhouses, the light intensity and angle of natural daylight vary significantly. The learning ability and generalizability of the training models for deep learning neural networks depend on the training dataset [21]. This puts a high requirement on the dataset for model training: the dataset must be diverse and complete at the same time. To improve the universality of the deep network model, the collected images were enhanced through flipping, illuminance balance, and rotation. Specifically, flipping and rotation enhance the detection ability and stability of the network model, while illuminance balance prevents the model performance from being affected by sensor difference and ambient light variation [22, 23]. Finally, a total of 2,000 tomato images were obtained, and divided into a training set (1,600), a validation set (200), and a test set (200).

3.3 Improvement of YOLO v5

In 2015, Joseph Redmon and Ali Farhadi put forward the YOLO, a target detection system based on a single neural network. As a one-stage end-to-end deep learning network detector [24-27], the YOLO regards object detection as the solution to a regression problem, and combines region proposal and classification into one network, such that the position, class, and corresponding confidence probability of every object in the input image can be determined through only one inference [28, 29]. In this way, the YOLO greatly improves the detection efficiency of objects. In 2020, YOLO v5 was released, and hailed for its precision and speed. According to the depth_multiple and width_multiple of parameters, YOLO v5 can be divided into four versions: YOLO v5l, YOLO v5m, YOLO v5x, and YOLO v5s. Among them, YOLO v5s boasts the fastest detection and the fewest parameters. Figure 3 shows the structure of YOLO v5s.

Tomato recognition is mainly adopted for robotic picking, calling for a ultralightweight and real-time recognition model. Our model was derived by improving the structure of YOLO v5s in the following aspects.

3.3.1 Improvement of spatial pyramid pooling (SPP)

The SPP involves three pooling operations [30]. While expanding the receptive field, max pooling would lower the resolution and sacrifice some details, resulting in the loss of local information. Drawing on the idea of atrous SPP (ASPP) [31], two dilated convolutional layers (Conv2D1 and Conv2D2) were added, each with three 3x3 kernels and three 3x3, 5x5, and 7x7 kernels. Without losing the sampled data, the dilated convolutional layers (Conv2D) can effectively capture multiscale global information under different sampling rates, thereby expanding the solution space and improving the detection accuracy of the model. Figure 4 shows the structure of the improved SPP.

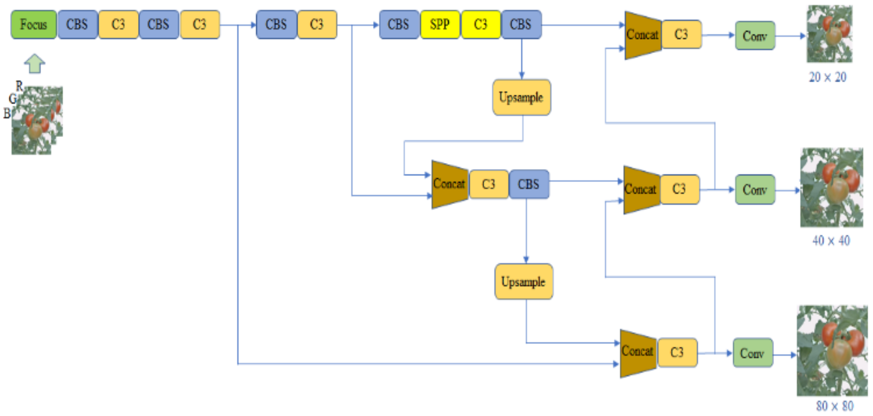


Figure 3. Structure of YOLO v5s

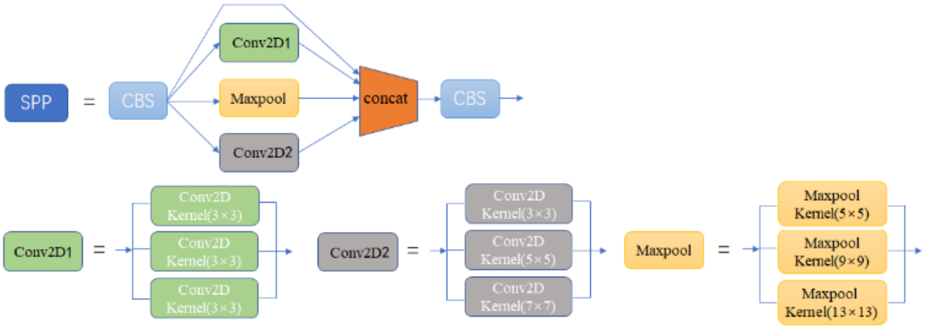


Figure 4. Structure of the improved SPP

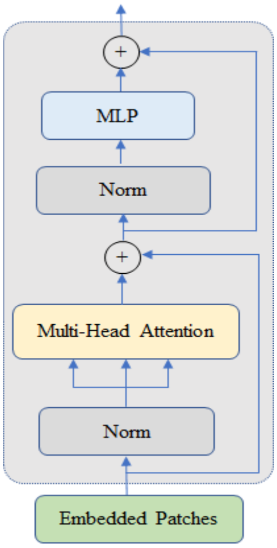


Figure 5. Structure of transformer encoder

3.3.2 Addition of transformer module

In the improved YOLO v5, module C3 after the SPP module of the backbone network is replaced by the transformer encoder module. Relying on the autonomous attention mechanism, the transformer encoder module directly compares the features of all spatiotemporal positions, and captures local features as well as global information. Facing static tomato images, the additive module can fit a stronger network model with the same computing resources, and thus speed up the inference. Figure 5 shows the structure of the transformer encoder [32].

3.3.3 Anchor box of KMC

Figure 6 shows the distribution of the image dataset. The widths of the targets are basically linearly correlated with their heights, the aspect ratios of the targets are generally stable, and the targets scatter across the entire image, i.e., the sample data are evenly distributed.

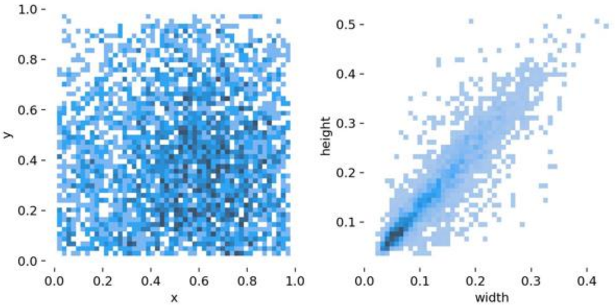


Figure 6. Distribution of the image dataset

To improve the detection accuracy of tomatoes, the anchor box should be similar to the target in size. For this purpose, the labeled boxes of the tomato dataset were clustered by the KMC [33]. All the images were shot by a binocular camera, and are of the same size. During the box clustering, the box width and height were taken as the only features. The intersection over union (IoU) between anchor and box was emphasized over the box size. Hence, the IoU was selected as the metric:

$$\mathrm{d}(gt, anchor) = 1 - IOU(gt, anchor) \tag{1}$$

$$IOU(gt, anchor) = A_{\cap}(gt, anchor)/A_{\cup}(gt, anchor) \tag{2}$$

where, d is the metric; gt is the ground truth box of the dataset; $anchor$ is the anchor box; $A_{\cap}$ and $A_{\cup}$ are the areas of the intersection and the union, respectively. The clustering process is as follows:

Step 1. Randomly select 9 gt boxes as the initial anchors.

Step 2. Assign the closet anchor to each gt box by formulas (1) and (2).

Step 3. Compute the mean width and height of each gt box in each class, and update the anchor.

Step 4. Repeat Steps 2-3 until the anchor changes no more.

The clustering results of our dataset were (19, 20) (22, 28) (27, 24) (33, 31) (40, 40) (52, 48) (65, 63) (85, 83) and (115, 107); the avr_IoU precision was 82.5%,13.9% higher than the default anchor (68.6%) of the YOLO. The new anchor obtained through clustering was taken as three feature maps, 80*80, 40*40, and 20*20, producing the prior box.

3.3.4 Improvement of non-maximum suppression (NMS)

The NMS is a technique for target detection algorithms to filter out redundant boxes. Traditionally, the NMS measures positioning precision with the IoU, computes the IoU between the detection box with the highest score and every other box, and deletes all the boxes with an IoU surpassing the threshold. Relying on the IoU alone, the NMS often incorrectly deletes the occluded objects. Apart from the IoU, the NMS with Distance IoU (DIoU) (DIoU-NMS) considers the distance between the centers of two boxes [34]. When two distant boxes have a large IoU, it means two objects have been detected, and the boxes should not be deleted. The DIoU-NMS can be defined as follows:

$$S_i = \begin{cases} S_i, & IOU - R_{DIOU}(M, B_i) < N_t \\ 0, & IOU - R_{DIOU}(M, B_i) \geq N_t \end{cases} \tag{3}$$

where, S_i is the confidence score of the current class; M is the box with the highest confidence; B_i is all the contrastive boxes in the current class; R_{DIOU} is the penalty term of DIoU loss function; N_t is a preset threshold. The distance R_{DIOU} between the centers of the two boxes can be expressed as:

$$R_{DIOU} = \frac{\rho^2(b, b^{gt})}{c^2} \tag{4}$$

where, ρ(·) is the Euclidean distance; b and b^{gt} are the center coordinates of the two boxes, respectively; c is the length (number of pixels) of the diagonal of the smallest box containing the two boxes. Length of the diagonal of the smallest box containing the two boxes shown in Figure 7, d = $\rho(b, b^{gt})$ is the distance between the centers of the two boxes.

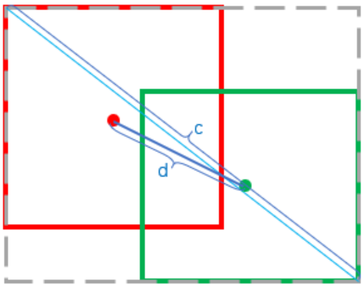


Figure 7. Length of the diagonal of the smallest box containing the two boxes

The DIoU-NMS was realized as follows: According to the demand of our project, the threshold was set to 0.5. Firstly, the boxes were ranked by confidence. Then, the box with the highest score was selected, and the DIoU between the box and every other box was computed. If the DIoU was greater than the threshold, the contrastive box was set to 0 and removed; otherwise, the contrastive box was retained. After that, the box with the second highest score was selected, and the DIoU between the box and every other remaining box was computed. If the DIoU was greater than the threshold, the contrastive box was set to 0 and removed; otherwise, the contrastive box was retained. The above steps were executed repeated.

4. EXPERIMENTS AND RESULTS ANALYSIS

4.1 Training and testing flow

To obtain a network model satisfying our project demand, the following steps were gone through: dataset generation, model training, model prediction, and model testing (Figure 8).

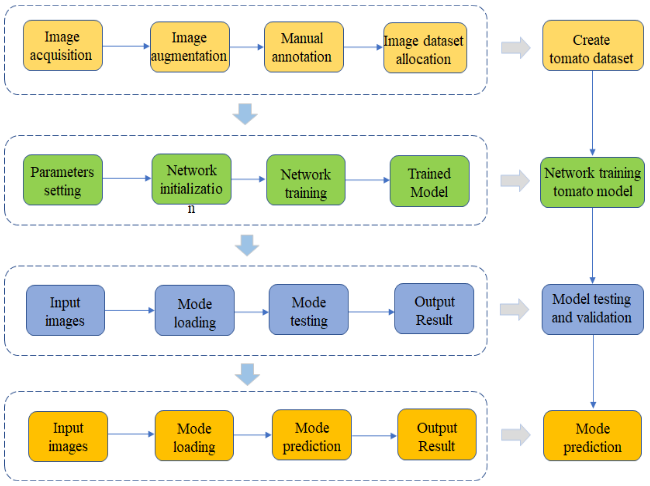


Figure 8. Training and testing flow of our model

4.2 Experimental platform

Our experiments were conducted on the following platform: Intel i5-9400F processor, Nvidia GeForce RTX 1650 graphics card, Windows 10, 8GB memory, PyTorch application framework, and image resolution of 640*640.

4.3 Evaluation metrics

The test performance was evaluated by mAP, precision, and recall. The closer the mAP is to 1, the better the overall performance of the network model. Since our research focuses on single-class detection, mAP equals average precision (AP), i.e., the area under the precision-recall (PR) curve:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

$$\text{mAP} = \int_0^1 P_r(d_r) \tag{7}$$

4.4 Experimental results

Figure 9 shows the training loss curve on the validation set. It can be observed that the loss converged quickly at the beginning of network training, the descent slowed down after 100 iterations, and stabilized at the 300th iteration.

Figure 10 shows the mAP on the training set. It can be observed that the curve approximated the peak after 100 iterations, and tended to be stable after 300 iterations. The process is relatively smooth, and the training is very stable, without overfitting. The results on the validation set indicates that the improved model is stable and reliable.

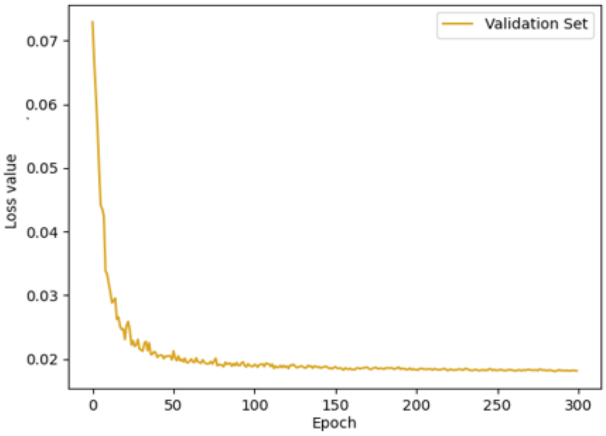


Figure 9. Loss curve of the improved model

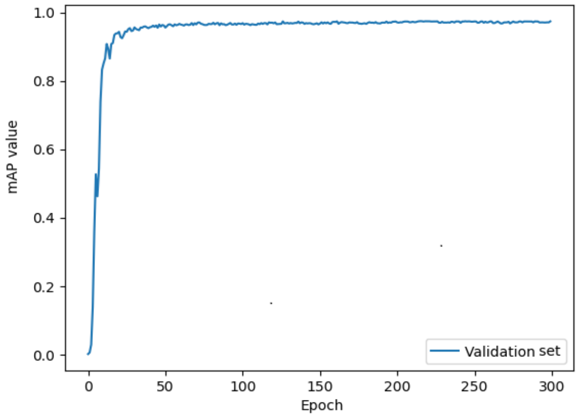


Figure 10. mAP curve of the improved model

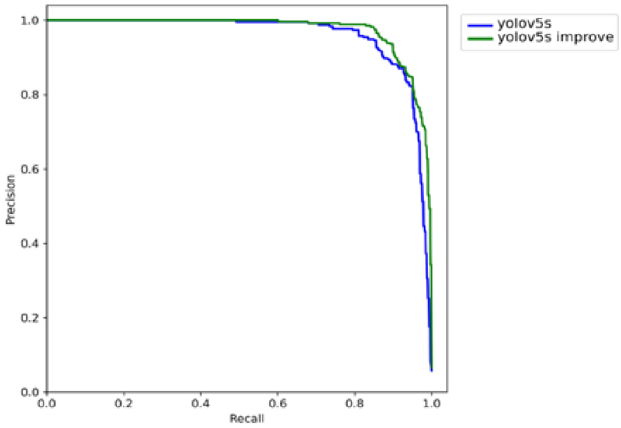


Figure 11. P-R curves of original and improved YOLO v5s

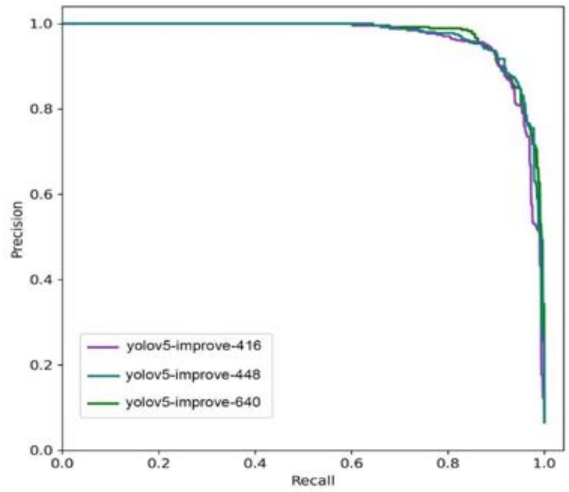


Figure 12. P-R curves at different resolutions

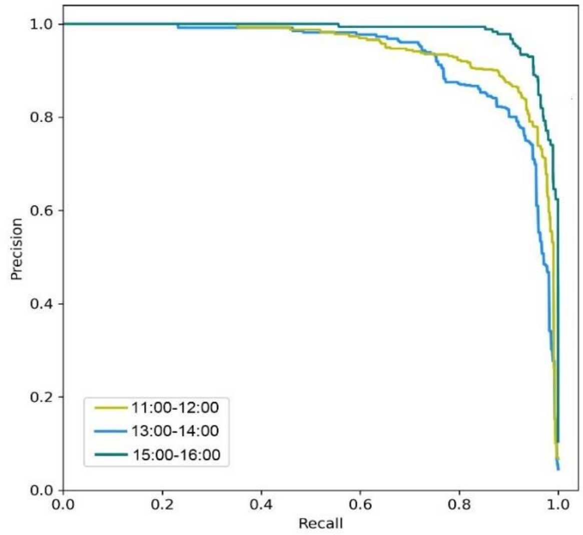


Figure 13. P-R curves at different illumination

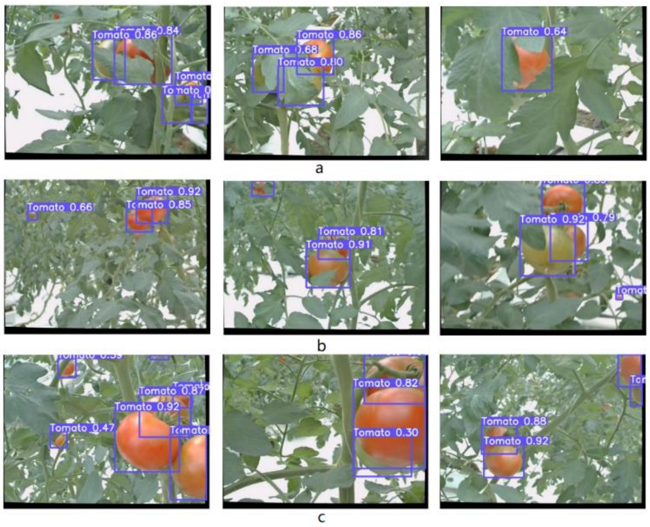


Figure 14. Recognition and detection results (a)Stem/leave occlusions; (b)Overlapping fruits; (c)Marginal areas

As shown in Figure 11, the improved YOLO v5 achieved a mAP of 97.3%, 1.4% higher than that of the original YOLO v5. Hence, our model improves the accuracy of tomato recognition, and supports the precise detection of tomatoes during picking.

The tomato recognition and detection model should be robust in the face of images of different resolutions. Our model was trained by images of three different resolutions 416*416, 448*448, and 640*640. Figure 12 presents the P-R curves at different resolutions, and Table 1 lists the performance data at different resolutions. The mAPs of the models trained at the three different resolutions were all above 96%. The higher the resolution, the larger the mAP. The inference time was positively correlated with the resolution. None of the three models exceeded 60ms. Thus, the inference time satisfies the demand of tomato picking. Overall, the improved model is relatively stable, with a strong generalization ability.

Considering the different illumination in greenhouses, the improved model was adopted to compute the data on three periods in a week. The P-R curves are displayed in Figure 13. Table 2 lists the result index of different illumination of the model.

The highest mAP (98%) was achieved in the afternoon, when the illumination was 4,650lux-8,862lux. The mAP (93%) was the lowest at noon, when the illumination was 11,330lux-35,940lux. The experimental results show that, with the growth of illumination, the precision fell slightly. The slight variation of precision is normal, as the collected images are inevitably different in angle and background. The overall precision (>90%) is enough to detect all the tomatoes, when the picking range is normal. Hence, the improved model is stable under different illumination.

Figure 14 shows the recognition and detection results. As shown in Figure 14(a), the tomatoes could be recognized correctly, even if over 70% were occluded by stems and leaves. As shown in Figure 14(b), the recognition effect was good, when the fruits seriously overlapped each other. As shown in Figure 14(c), the fruits in marginal areas were well detected. These fruits appear in the intersection between views, which is inevitable during tomato picking. The above results show that the improved model can work effectively on fruits occluded by stems/leaves, overlapped by other fruits, and in marginal areas.

Tomato picking raises a high requirement on the precision of identifying the center of tomatoes. Formulas (8)-(10) derive the positioning error by computing the distance from the predicted center to the actual center of tomatoes. Through the calculation of 200 tomatoes, the mean squared error (MSE) was obtained as 0.06. The sample data were all normalized data ∈(0, 1). That is, the positioning error was 0.06mm, when the detection precision was 1mm. The positioning precision fully meets the requirements of picking robots.

$$d_i = \sqrt[2]{(X_i^P - X_i^G)^2 + (Y_i^P - Y_i^G)^2} \tag{8}$$

$$\bar{d} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{9}$$

$$\sigma = \frac{1}{N}\sum_{i=1}^{N}(d_i - \bar{d})^2 \tag{10}$$

To better evaluate the tomato recognition performance, the improved model was compared further with other detection models (Table 3).

Table 1. Performance data at different resolutions

Models	mAP (%)	Inference time per image (ms)
Improve-yolov5-416	96.6	53ms
Improve-yolov5-448	97.0	55ms
Improve-yolov5-640	97.3	57ms

Table 2. The result index of different illumination

Period	Peak illumination (lux)	Lowest illumination (lux)	P (%)	R (%)	F1 (%)	mAP (%)
11: 00-12: 00	23250	10000	86.5	90.2	88	94.6
13: 00-14: 00	35940	11330	83.7	87.5	86	92.7
15: 00-16: 00	8862	4650	92.6	94.1	93	98.1

Table 3. Performance indices of different models

Model	P (%)	R (%)	mAP50 (%)	F1(%)	Inference time per image (ms)	Model size
Faster-RCNN	82.9	58.4	66.2	69	426	98M
YOLO v3	93.1	67.3	87.2	78	189	78M
YOLO v5	94.5	85.6	95.9	90	38	14M
Improve-YOLO v5	93.9	90.5	97.3	92	57	29M

As shown in Table 3, our improved YOLO v5 model achieved the highest mAP, which was 31.1%, 10.1%, and 1.4% higher than that of Faster-RCNN, YOLO v3, and YOLO v5, respectively. It took 57ms for our improved model to infer on each image, which is 19ms longer than YOLO v5, the most time efficient model. In terms of size, the improved YOLO v5 is 29MB, which is 15M larger than YOLO v5, the smallest model, and 69MB smaller than Faster-RCNN, the largest model. The time cost and size of our model can satisfy the needs of tomato picking robots. In general, our improved YOLO v5 is the best algorithm for tomato picking robots.

5. CONCLUSIONS

This paper introduces deep learning to improve YOLO v5. Specifically, data enhancement was added to the original network, the SPP was improved, and NMS was optimized. These improvements enhance the ability to extract features at multiple resolutions, to fuse multiscale features, and to detect objects precisely. The improved model achieved a mAP of 97.3%, a precision of 93.9%, a recall of 90.5%, and an F1-score of 92%. Experimental results show that our model can excellently identify and detect tomatoes under different illumination, branch/leave occlusions, and fruit overlapping in the complex environment of greenhouses. The detection results of our model fully satisfy the operational requirements of tomato picking robots. The future work will probe into the recognition and detection of tomatoes at night, aiming to realize 24/7 operations of tomato picking robots.

ACKNOWLEDGMENTS

The work is supported by S&T Program of Hebei (Grant No.: 21327212D).

REFERENCES

[1] Clark, B., Jones, G., Kendall, H., Taylor, J., Cao, Y., Li, W., Frewer, L. (2018). A proposed framework for accelerating technology trajectories in agriculture: a case study in China. Frontiers of Agricultural Science and Engineering, 5(4): 485-498. https://doi.org/10.15302/J-FASE-2018244

[2] Wang, H.N., Yi, J.G., Zhang, X.H. (2020). Research progress on recognition and localization technology of tomato picking robot. Journal of Chinese Agricultural Mechanization, 41(5): 188-196. 10.13733/j.jcam.issn.2095-5553.2020.05.031

[3] Liu, J.Z. (2017). Research progress analysis of robotic harvesting technologies in greenhouse. Nongye Jixie Xuebao/Transactions of the Chinese Society of Agricultural Machinery, 48(12): 1-18

[4] Li, H., Tao, H.X., Cui, L.H., Liu, D.W., Sun, J.T., Zhang, M. (2021). Recognition and localization method of tomato based on SOM-K-means algorithm. Nongye Jixie Xuebao/Transactions of the Chinese Society of Agricultural Machinery, 52(1): 23-29.

[5] Xiang, R., Jiang, H., Ying, Y. (2014). Recognition of clustered tomatoes based on binocular stereo vision. Computers and Electronics in Agriculture, 106: 75-90. https://doi.org/10.1016/j.compag.2014.05.006

[6] Xi, H.Y., Zhang, D., Zhou, T., Yang, Y.X. (2021). Research on the method of picking up overlapping tomato fruits by picking robot. Journal of Agricultural Mechanization Research, 43(12): 17-23.

[7] Yu, Y., Zhang, K., Yang, L., Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. Computers and Electronics in Agriculture, 163: 104846.

[8] Zhao, Y., Gong, L., Huang, Y., Liu, C. (2016). Robust tomato recognition for robotic harvesting using feature images fusion. Sensors, 16(2): 173-185. https://doi.org/10.3390/s16020173

[9] Zhao, Y., Gong, L., Zhou, B., Huang, Y., Liu, C. (2016). Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. Biosystems Engineering, 148: 127-137. https://doi.org/10.1016/j.biosystemseng.2016.05.001

[10] Yamamoto, K., Guo, W., Yoshioka, Y., Ninomiya, S. (2014). Detection of Intact Tomato Fruits Using Image Analysis and Machine Learning Methods. Sensors, 14(7): 12191-12206. https://doi.org/10.3390/s140712191

[11] Malik, M.H., Zhang, T., Li, H., Zhang, M., Shabbir, S.,

Saeed, A. (2018). Mature tomato fruit detection algorithm based on improved HSV and watershed algorithm. IFAC-PapersOnLine, 51(17): 431-436. https://doi.org/10.1016/j.ifacol.2018.08.183

[12] Hu, C., Liu, X., Pan, Z., Li, P. (2019). Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set. IEEE Access, 7: 154683-154696. https://doi.org/10.1109/ACCESS.2019.2949343

[13] Liu, G., Nouaze, J. C., Touko Mbouembe, P.L., Kim, J.H. (2020). YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3. Sensors, 20(7): 2145-2165. https://doi.org/10.3390/s20072145

[14] Ma, Z.H., Shen, G.R., Lyu, J.D. (2017). Study on the method of separating apple fruits based on limiting corrosiont. Jiangsu Journal of Agricultural Sciences, 33(6): 1372-1378.

[15] Magalhães, S.A., Castro, L., Moreira, G., Dos Santos, F.N., Cunha, M., Dias, J., Moreira, A.P. (2021). Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse. Sensors, 21(10): 3569-3593. https://doi.org/10.3390/s21103569

[16] Chen, Y., Wang, J.S., Zeng, Z.Q., Zou, X.J., Chen, M.Y. (2019). Vision pre-positioning method for litchi picking robot under large field of view. Transactions of the Chinese Society of Agricultural Engineering, 35(23): 48-54.

[17] Liu, G., Mao, S., Kim, J.H. (2019). A mature-tomato detection algorithm using machine learning and color analysis. Sensors, 19(9): 2023. https://doi.org/10.3390/s19092023

[18] Mu, Y., Chen, T.S., Ninomiya, S., Guo, W. (2020). Intact detection of highly occluded immature tomatoes on plants using deep learning techniques. Sensors, 20(10): 2984. https://doi.org/10.3390/s20102984

[19] Wan, P., Toudeshki, A., Tan, H., Ehsani, R. (2018). A methodology for fresh tomato maturity detection using computer vision. Computers and electronics in agriculture, 146: 43-50. https://doi.org/10.1016/j.compag.2018.01.011

[20] Kamilaris, A., Prenafeta-Boldú, F.X. (2018). A review of the use of convolutional neural networks in agriculture. The Journal of Agricultural Science, 156(3): 312-322. https://doi.org/10.1017/S0021859618000436

[21] Kamilaris, A., Prenafeta-Boldú, F.X. (2018). Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147: 70-90. https://doi.org/10.1016/j.compag.2018.02.016

[22] Ma, J., Li, Y., Chen, Y., Du, K., Zheng, F., Zhang, L., Sun, Z. (2019). Estimating above ground biomass of winter wheat at early growth stages using digital images and deep convolutional neural network. European Journal of Agronomy, 103: 117-129. https://doi.org/10.1016/j.eja.2018.12.004

[23] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Computers and Electronics in Agriculture, 157: 417-426. https://doi.org/10.1016/j.compag.2019.01.012

[24] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779-788.

[25] Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7263-7271.

[26] Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint, arXiv1804.02767. https://doi.org/10.48550/arXiv.1804.02767

[27] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint, arXiv2004.10934. https://doi.org/10.48550/arXiv.2004.10934

[28] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779-788.

[29] Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 7263-7271. https://doi.org/10.1109/CVPR.2017.690

[30] He, K., Zhang, X., Ren, S., Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9): 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[31] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4): 834-848. https://doi.org/10.1109/TPAMI.2017.2699184

[32] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint, arXiv2010.11929. https://doi.org/10.48550/arXiv.2010.11929

[33] Wang, X., Liu, J. (2021). Tomato anomalies detection in greenhouse scenarios based on YOLO-dense. Frontiers in Plant Science, 12: 533. https://doi.org/10.3389/fpls.2021.634103

[34] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D. (2020). Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, 34(7): 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999