Multi-Growth Period Tomato Fruit Detection Using Improved Yolov5

Yingyan Yang[1], Yuxiao Han[2], Shuai Li[2], Han Li[1,*] and Man Zhang[2]

[1]*Key Laboratory of Smart Agriculture System Integration Research, Ministry of Education, China Agricultural University, Beijing 100083, China*

[2]*Key Laboratory of Agricultural Information Acquisition Technology, Ministry of Agriculture and Rural Affairs, China Agricultural University, Beijing 100083, China*

Abstract: In agricultural mechanized production, in order to ensure the efficiency of hand-eye cooperative operation of tomato picking robot, the recognition accuracy and speed of multi-growth period tomato fruit is an important basis. Therefore, in order to improve the recognition speed of multi-growth period tomato fruit while ensuring or improving the accuracy, this paper improves the Yolov5s model by adding the architecture of the lightweight mobilenetv3 model. Firstly, the deep separable convolution is replaced in the backbone network of Yolov5s, which reduces the amount of convolution operation. Secondly, the linear bottleneck inverse residual structure is fused to obtain more features in high-dimensional space and perform convolution operation in low-dimensional space. Third, the attention mechanism is inserted into the last layer of the network to highlight features and improve accuracy. The research results show that the recognition accuracy of the improved Yolov5 model remains above 98%, the CPU recognition speed is 0.88f·s^{-1} faster than Yolov5s, and the GPU recognition speed is 90 frames per second faster than Yolov5s. Finally, a set of the recognition software system of multi-growth period tomato fruit is designed and developed by using RealSense D435i depth camera and PYQT. The software system further verifies the feasibility of the improved Yolov5 model, and lays a foundation for the visual software design of agricultural picking robot picking recognition.

Keywords: Visual recognition, Tomato fruit detection, Yolov5, Lightweight.

1. INTRODUCTION

As the second most consumed vegetable after potato, tomato is the most popular family garden in the world [1]. According to the official data of FAOSTAT, China is the country with the largest planting scale and the highest output of tomatoes every year: in 2019, the planting area reached 1.08 million hectares and the output reached 62.86 million tons [2]. However, tomato picking in natural environment is a labor-intensive work [3]. The harvesting link faces two problems: large labor demand and short cycle demand. Therefore, the research of tomato picking robots are of great significance, which can not only greatly reduce the amount of manual work, but also shorten the working time to improve efficiency.

In the research of automatic picking, visual recognition of mature fruit is an important part. Over the years, there are a lot of work focusing on this field. The main recognition methods are traditional fruit image processing and deep learning algorithm.

Traditional fruit image processing and recognition include image color segmentation, texture boundary segmentation, etc. Based on the threshold method using R-G equation, Aref *et al*. [4] extracted mature tomatoes by removing the background in RGB color space. The classification accuracy obtained referring to RGB and shape features was more than 90%. Chunhua Hu *et al*. [5] used Gaussian density function of H and S in the HSV color space to help segment tomato regions from the background, and used an adaptive threshold intuitionistic fuzzy set (IFS) to identify the tomato's edge. As a result, the detection accuracy of tomatoes was improved. Payne a *et al*. [6] also made some improvements to the image color algorithm in 2014, which reduced the weight of color processing and increased the application of texture filtering. Although the effect of image segmentation was obviously improved, this method used texture features to recognize fruits requires a large number of texture information and samples.

It can be seen that the traditional image processing methods mainly rely on color and texture analysis to segment the image and achieve the purpose of extracting the target fruit. The recognition effect of this method is enough for single fruit in the experimental environment, but the actual production environment is much more complex, and the clustering or sheltered fruits will greatly reduce the recognition accuracy of this method. In addition, the workload of texture extraction and color segmentation is very large. Therefore, in comparison, deep learning algorithm can be better

*Address correspondence to this author at the College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China; Tel: :+86-13041254819; E-mail: cau_lihan@cau.edu.cn

E-ISSN: 2409-9694/22

applied to fruit recognition because it has a strong data representation ability for images.

Sa *et al*. [7] first applied R-CNN algorithm to RGB images and NIR images in 2016 to detect the number of sweet peppers. Finally, the average (F1) of accuracy and recall rate was 0.84. Jia w K *et al*. [8] improved the mask R-CNN model by combining RESNET and DENSENET in order to make it more suitable for the recognition of overlapping fruits. Kang HW *et al*. [9] also developed an automatic label generation module framework and a fruit detector based on deep learning to realize real-time detection of apples in the orchard. The detection results were reliable and efficient.

With the continuous improvement of recognition accuracy, more and more researchers began to pursue the optimization of real-time detection speed. Yolo algorithm was first proposed and applied as one stage method in 2015. In 2019, Kolrala A *et al*. [10] proposed a 'MangoYolo' model based on Yolov2 and Yolov3 for fruit detection. Its F1 reached 0.97, and the proportion of model memory was reduced. Based on Yolov3, Liu *et al*. [11] proposed a tomato recognition method in complex environment in 2020. The accuracy of this model was 94.58%, and the image processing time was 0.054 seconds, which was 0.17 seconds less than that of fast R-CNN.In 2021, Gai *et al*. [12] added dense connection structure to Yolov4's cspdarknet53 network to identify cherry fruits. The average accuracy of the improved network to identify cherries in complex environments increased by 0.15.

In summary, Yolo series algorithms are more suitable for real-time detection, and the optimized network structure can better improve the detection accuracy. In the actual production and picking process of tomato, considering the speed and accuracy requirements of visual recognition, this paper selects Yolov5s model with adaptive anchor box clustering algorithm, adds convolution structure and attention mechanism, and integrates lightweight mobilenetv3 architecture to optimize the backbone network. Finally, the improved Yolov5 model was used to identify tomato fruits with different maturity.

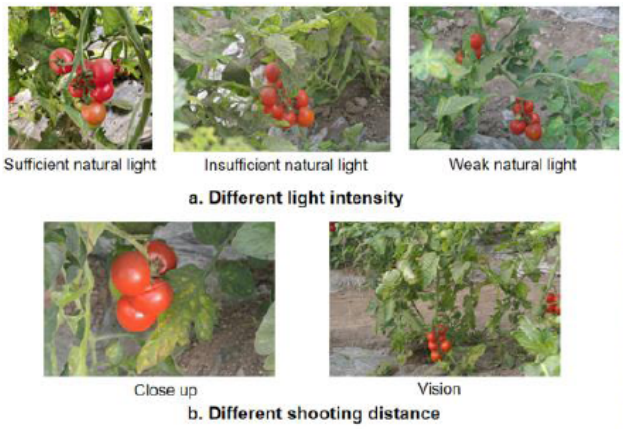
2. MATERIALS AND METHODS

2.1. Image Data Collection

The image data set used in this study are the pictures of big tomato fruits, which were collected in the tomato greenhouse of Chinese Academy of Agricultural Sciences, North3rd Ring West Road, Haidian District, Beijing. The tomato variety is Jiaxina. The images were taken from 3:00 pm to 4:30 pm on January 3, 2019 and from 2:00 pm to 4:00 pm on January 8, 2019. The image collection equipment is a high-resolution, high-definition Nikon J1 digital camera. The total number of original fruit images taken is 600. Figure **1** below shows some samples of original tomato datasets collected:

2.2. Data Annotation and Classification

According to the color comparison table of tomato maturity grade, the tomato growth and ripening process can be generally divided into four periods, namely, unripe period, half-ripe period, full-ripe period and over-ripe period [13]. Considering that the color thresholds of full-ripe period and over-ripe period are relatively close, it is difficult to distinguish in the labeling process,



(**a**) partial sample of original tomato datasets

(**b**) enlarged view of a single sample

Figure 1: samples of original tomato datasets collected.

therefore the two periods are combined as ripen period. Based on the HSV(Hue, Saturation, Value) color theory and the tomato maturity color comparison table, this paper adopts the self-defined threshold segmentation method of H-component - using the ratio of binary red pixels to tomato contour pixels to calculate the maturity level [14]. Finally, the ripening process of tomatoes can be roughly divided into three categories: tomatoes on the red surface are marked as "mature tomatoes", corresponding to the ripen period; tomatoes with a green surface are labeled as "immature tomatoes", corresponding to the unripe period; tomatoes with surface color between red and green are marked as "medium tomatoes", corresponding to half-ripe period, as shown in Figure **2**. The annotation tool is labelImg written by Qt in Python, which is designed by TzuTa Lin. After the data set is annotated by the software, the XML file with location and category can be obtained, which will be converted to the TXT format required later.

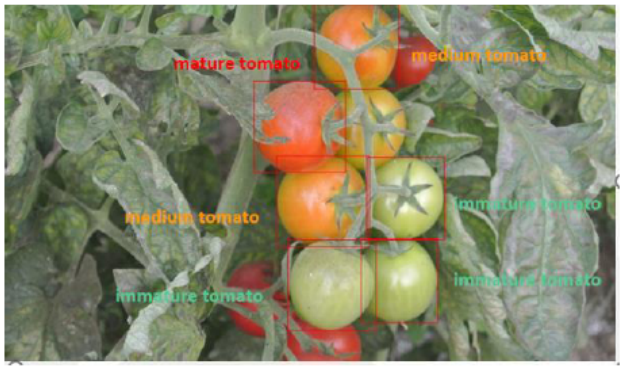


Figure 2: Effect diagram of tomato maturity division.

2.3. Data Enhancement

Data augmentation is a common way to expand the variability of the training data by artificially enlarging the data set using label-preserving transformations [15]. Generally speaking, it is to generate more data based on limited data, so that the generalization ability of the model obtained through the training set is stronger. Data enhancement can generally be divided into supervised and unsupervised methods. The difference between unsupervised data enhancement and supervised data enhancement lies in whether the augmentation methods are related to the data labels. In the actual process of training the model, some commonly used data enhancement methods include translation, flip, rotation, random color change, contrast, brightness enhancement, etc.

In order to further expand the data set to ensure the accuracy of training, this paper enhanced the initial 600 tomato images. The adopted data enhancement technologies include random HSV adjustment, Gaussian noise increase, rotation and translation, mirroring, etc. A total of 400 tomato datasets were added to improve the generalization performance and then were annotated. Finally, 1000 labeled tomato pictures were used as input samples for deep learning to extract features. Figure **3** shows some tomato fruit samples processed by the data enhancement method.

2.4. Data Set Division

The data set of 1000 tomato fruit pictures after labeling is randomly divided into training set and verification set at the ratio of 9:1. Then 100 pictures (10% data set) were randomly selected as the test set. After that, the number of tomatoes with different maturity in each tomato fruit data set was counted in batch by using python. The number of labeled tomato fruits is shown in Table **1**:

Figure 3: Tomato fruit image with four data enhancement methods.
(Including rotation, mirroring, HSV change, noise processing)

Table 1: Number Statistics of Tomato Fruit Data Set after Division

	Mature Tomato Amount	Medium Tomato Amount	Immature Tomato Amount
Train set	1560	3423	692
Validation set	184	343	84
Test set	163	299	78

3. METHOD

3.1. Yolov5 Model

Yolov3 [16] was proposed in 2018. Its network structure mainly includes input, a new backbone feature extraction network (darknet-53), feature pyramid network (FPN, neck), and category prediction. Yolov3 is a representative of single-stage deep learning structure innovation. With the development of deep learning, after Yolov3 was proposed, Alexey *et al*. [16] proposed Yolov4 in 2020. Yolov4 uses advanced CSPDarknet53 for feature extraction and the SPP+PAN module to further enhance the expression ability of features [17]. In the same year, the improved Yolov5 was proposed.

On the basis of Yolov3, the Yolov5 algorithm adds three improvement measures to the input: mosaic data enhancement, adaptive anchor frame calculation, and adaptive picture scaling. Some network layers, such as focus structure, CSP structure, and FPN+PAN structure, were added between the backbone network and the final output layer. In the output layer, the anchor frame mechanism and the loss function were improved. Figure **4** shows the network structure flow of the Yolov5 algorithm.

Yolov5 algorithm includes four models versions: Yolov5s, Yolov5m, Yolov5l, and Yolov5x. The depth and the number of residuals of these four models increase, and the accuracy increases in turn, but the speed detection decreases in turn. The research object of this paper is to detect tomato fruits with three kinds of maturity in real-time. Therefore, the improved algorithm proposed in this paper is based on the Yolov5s model.

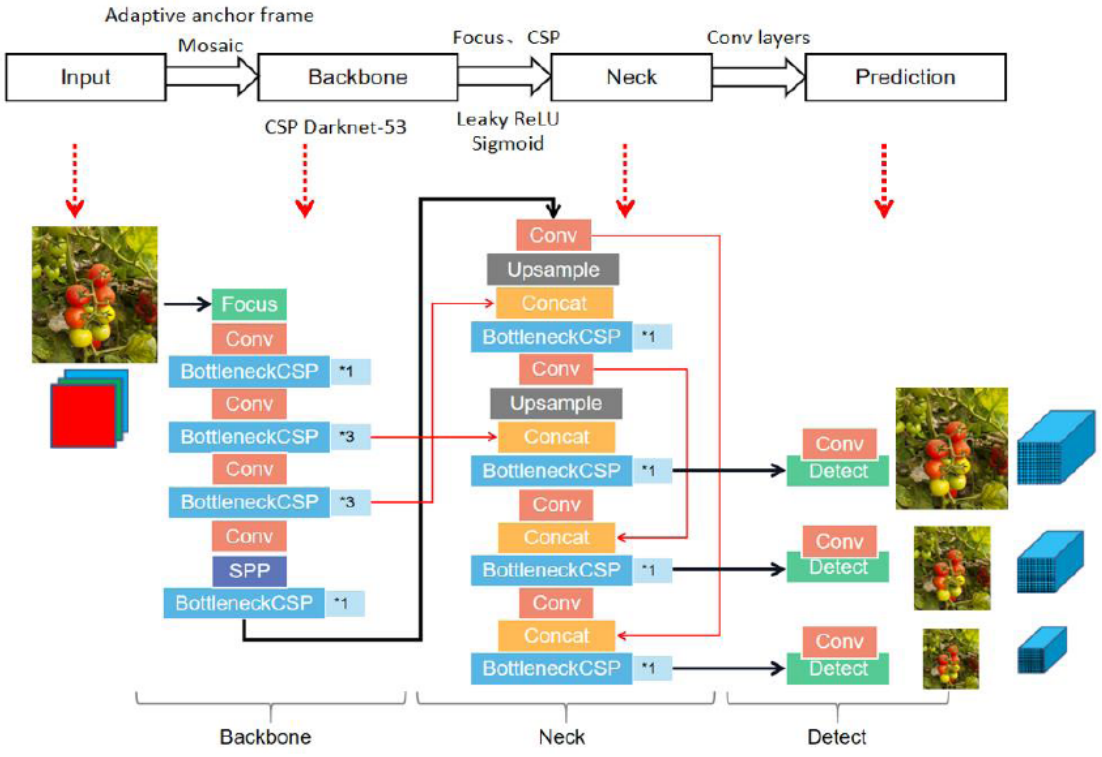


Figure 4: Model structure of Yolov5 algorithm.

3.2. Improved Yolov5s Model

This paper is devoted to realizing accurate and fast real-time detection of tomato fruits with different maturity, to meet the needs of actual agricultural production and life. Yolov5s model has a high recognition accuracy, but it has many parameters, large weight files, and high requirements on computer hardware. Therefore, the model needs to be compressed as much as possible to meet or improve the accuracy. By reducing the number and volume of the network weight parameters of the algorithm, it is convenient to improve the detection speed and better deploy it in the hardware equipment.

The lightweight network was developed in 2017 and has been extensively studied. The lightweight model was defined as a neural network model with small parameters and low memory requirements [18]. The core was to transform the network from both volume and speed while maintaining accuracy as much as possible. At present, the research on lightweight network mainly includes artificial design and automatic search based on neural network structure. Manually designed lightweight networks include: SqueezeNet, MobileNet, GhostNet, and their improved versions. The lightweight networks for neural network structure search include NasNet, MnasNet, etc. Among them, mobilenet series is the current mainstream lightweight model.

Optimized Yolov5 model combined with the mobilenetv3 [19] lightweight network was adopted in this paper for tomato fruit recognition. The optimization operation mainly includes the following aspects: adding the deep separable convolution structure, the linear bottleneck inverse residual structure, and the lightweight attention mechanism structure, completing the change of the configuration file and the function call of the training file. Figure **5** shows the brief flow of optimization operation, and Figure **6** shows the specific network structure of optimizing Yolov5:

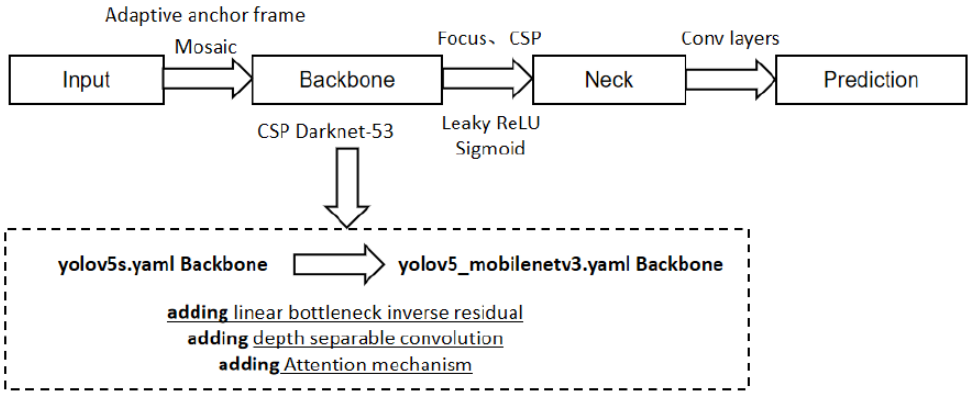


Figure 5: operation flow diagram of optimizing Yolov5s.

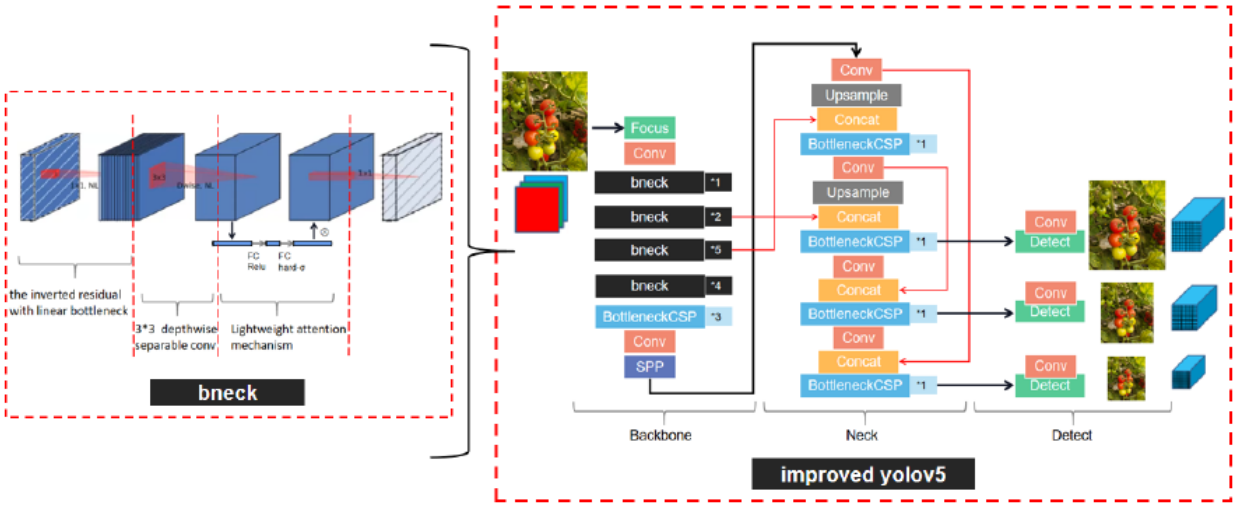


Figure 6: The network structure of improved Yolov5s.

3.2.1. Deep Separable Convolution Structure

Reducing the network overhead can significantly improve the speed of model recognition, and eliminating the convolution calculation can effectively reduce the time-consuming network overhead. Deep separable convolution [20] is a structure for optimizing the convolution part, which splits the ordinary convolution into a deep convolution and a point-by-point convolution, and transform the original multiplication convolution operation into an addition operation, thus greatly reducing the convolution parameter amount and calculation amount.

It is assumed that the size of the convolution kernel is DK×DK×M. The number of channels is M and the number is N. Each convolution kernel needs DW×DH operations. The calculation amount of deep separable convolution consists of deep convolution and point by point convolution: assuming that the calculation amount of standard convolution is Q1 and that of deep separable convolution is Q2, the following formulas 1, 2 and 3 can be obtained:

$$Q1 = DK * DK * M * N * DW * DH \tag{1}$$

$$Q2 = DK * DK * M * DF * DF + DF * DF * N * M \tag{2}$$

$$Q2 / Q1 = (1/N + 1/DK^2) \tag{3}$$

From the analysis of formula (3), $Q2/Q1 << 1$, which means $Q2 << Q1$. With the same number of weight parameters, compared with the standard convolution operation, the calculation amount of deep separable convolution is greatly reduced, even several times. Thus, the structure achieves the purpose of improving the network operation speed.

3.2.2. Linear Bottleneck Inverse Residual Structure

Linear bottlenecks and inverse residuals [21] were added from mobilenetv2 and form an efficient basic module structure. First, in order to increase the number of channels and obtain more features, the dimension upgrading part is added at the beginning of the network architecture. Secondly, in order to prevent the damage characteristics of relu, a shortcut was introduced, and the last relu was removed and changed to linear. That is, when the step size was 1, the 1 * 1 convolution dimension was first raised, and then the depth convolution was performed to extract features. Then, the dimension was reduced by point by point convolution of linear. Finally, the input and output were added to form a residual structure. Figure **7** below shows the structure comparison before and after the linear bottleneck inverse residual structure was added:

In summary, on the one hand, the linear bottleneck inverse residual structure was raised to the high-dimensional space to obtain more features; On the other hand, linear point by point convolution reduces the dimension to ensure that the features were not destroyed; At the same time, 3 * 3 convolution was

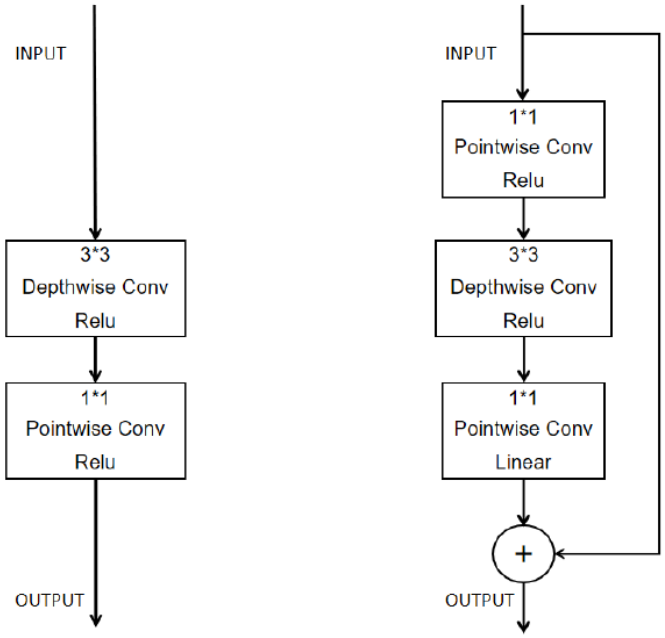


Figure 7: Comparison before and after the increase of linear bottleneck inverse residual structure.

performed in a low dimensional space, which can reduce the amount of calculation.

3.2.3. Lightweight Attention Mechanism

The SENET module (squeeze and networks, SENET) [22] is a kind of visual attention mechanism network, wherein a new feature re-calibration strategy, illustrating the importance of each feature channel, is automatically obtained through learning, and then useful features are promoted, and unimportant features are suppressed accordingly.

The attention mechanism SENET is mainly divided into two parts. One is the squeeze part, which compresses the spatial resolution of the previous layer to 1x1 through adaptive avgpool2d. The second part is the exception part, which changes the compressed part linearly. Then relu and linear ensures that the number of channels is consistent with the number of channels in the next layer. After that sigmoid function is used to activate it. The obtained layer is multiplied point by point directly with the layer convoluted by the original input layer. The advantage is that some features will be highlighted and some features will be suppressed. As a feature extraction method, attention mechanism can be applied in many network structures.

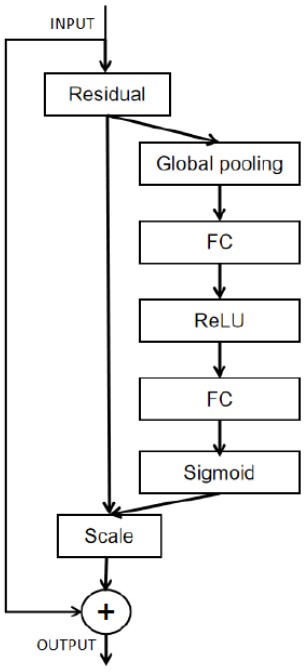


Figure 8: Network structure of attention mechanism of mobilenetv3.

SENET module is added to mobilenetv3. Its function is to adjust the weight of each channel with some adjustments being made. First, average pooling was used to change each channel into a value, and then the output of channel weight was obtained after passing through two full connection layers. The second full connection layer used the hard sigmoid activation function. Then the weight of the channel was multiplied back to the original feature matrix to obtain a new feature matrix. The number of channels of the second linear layer was changed to one fourth of the number of channels of the next layer. The h-swish activation function was also used in the network structure to replace the swish function, which reduced the amount of calculation and improved the performance. The network structure is shown in Figure **8**.

3.3. Evaluation Index

In order to compare different target detection algorithms in multiple directions, a series of recognized evaluation indicators have been proposed for detection results, such as intersection over Union (IOU), frames per second, precision, recall, average precision (AP), mean average precision (MAP), etc.

In actual deep learning detection, precision and recall was used to jointly evaluate the model. F1 score can be regarded as a weighted average of model accuracy and recall, and is also an indicator of model evaluation. F1 = 2 * (precision * recall) / (precision + recall). The larger the F1 value, the better the model. And AP is expressed as the integral value of the accuracy rate to the recall rate of a certain category on the accuracy rate recall curve; MAP is expressed as the integral of the accuracy rate to the recall rate on the accuracy recall rate curve of the whole data set, which is equal to the mean value of all target AP. Therefore, AP or MAP are often used to evaluate the quality of the model.

At the same time, confusion matrix was used for error classification between categories, so as to visually check whether there are specific categories confused with each other. The structure of the confusion matrix is shown in Table **2**:

Table: 2 Confusion Matrix

	Positive	Negative
True	TP	TN
False	FP	FN

In the confusion matrix: if the label is a positive sample, and the number of positive samples classified is true, the sample is referred to as TP; if the label is a positive sample, and the number of negative samples

classified is false, the sample is referred to as FN; if the label is a negative sample, and the number of positive samples classified is false, the sample is referred to as FP; if the label is a negative sample, and the number of negative samples classified is false, the sample is referred to as TN; These four parameters are the parameters that constitute the accuracy rate and the recall rate. See the following formulas 4, 5 and 6 for details:

$$Precision = TP/(TP+FP) \tag{4}$$

$$Recall = TP/(TP+FN) \tag{5}$$

$$AP = \int_0^1 P(R)dR \tag{6}$$

In the formula, P (R) represents the PR curve function of accuracy and recall. It can be seen from the analysis that when comparing the detection results, the recognition accuracy of the model can be compared using the average accuracy rate. The speed of the model can be compared by using the time of identifying a picture or a frame of video, so as to evaluate the quality of the model.

3.4. Fruit Detection Software Development

Intel real sense d435i camera, which is developed by Intel Corporation(Santa Clara, California, USA), was used to provide depth image and RGB image. Pyrealsene2 module and the OpenCV library were utilized to develop software for real-time and convenient visualization of processing results [23].

Figure 9: basic structure of Intel real sense d435i.

The main structure of Intel Real Sense D435i is shown in Figure **9**: It integrates two IR Stereo Cameras, an IR Projector and a Color Camera. Intel Real Sense D435i is used to combine two-dimensional information and depth information of tomatoes with different maturity, for tomato recognition and positioning. Figure **10** shows the flow chart of the application of Intel real sense D435i in tomato robot picking.

4. RESULTS AND ANALYSIS

4.1. Experimental Preparation and Model Setting

Combined with 1000 labeled tomato fruit pictures, we began to use Yolov3, Yolov5, and improved Yolov5 models to train the data sets. The operating platform of the training test is a desktop computer with Ubuntu 18.04 operating system. Its configuration environment is Intel (R) Xeon (R) CPU, 32 GB memory, and the graphics card is GeForce GTX 1080ti model and 11 GB video memory. The programming language used is Python 3.8, and the deep learning framework built is Pytorch 1.8. After the training, the experimental operating platform for connecting the depth camera to

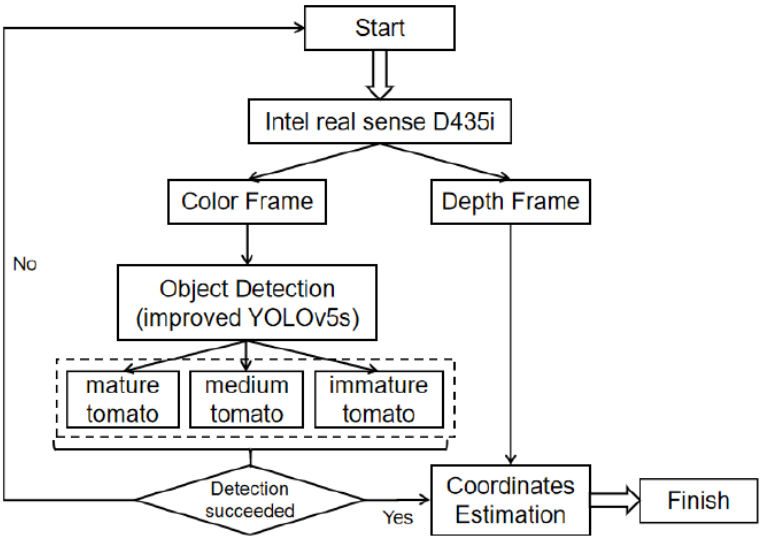


Figure 10: the flow chart of the application of Intel real sense d435i in tomato robot picking.

check the recognition effect is the notebook computer with windows 10, Intel (R) Xeon (R) CPU, 8 GB memory, and the desktop computer with Linux operating system was used to train. The training rounds (epochs) were 300. After the network structure was built and the configuration file was modified, the training and testing of the model were started.

4.2. Comparison of Traditional Image Processing, Yolov3 and Yolov5s Results

The main process of traditional image processing methods for tomato recognition includes image graying and binarization, edge extraction and iterative random circle. Table **3** below shows the recognition of mature tomatoes obtained by traditional image processing methods. Taking mature tomatoes as an example, it can be seen that the accuracy of traditional image processing methods is less than 90%, which is relatively low. Therefore, the later part focuses on the analysis and comparison of the improvement effect of deep learning.

Table 3: Accuracy Statistics of Mature Tomatoes using Traditional Image Processing Methods

Method	Number of Fruits	Number of Fruits Detected	Accuracy (%)
Traditional image processing methods	184	157	85.3%

Figure **11** shows the confusion matrix and PR curve of tomato fruit training with Yolov3; Figure **12** shows the confusion matrix and PR curve of tomato fruit training with Yolov5s. When IOU is 0.5, the AP of all pictures of tomatoes of each maturity is obtained first, and then the $MAP_{0.5}$ is obtained by averaging all classes. In terms of detection accuracy, the $MAP_{0.5}$ obtained by Yolov3 training is 0.988 and obtained by Yolov5 training is 0.985.

In terms of detection speed, the average frame rate of the video detected by the CPU in the Yolov3 model is 0.926f·s^{-1} and by GPU is 45f·s^{-1}. The average frame rate of video detected by CPU in Yolov5 model is 1.299f·s^{-1} and by GPU is 96f·s^{-1}. After the training

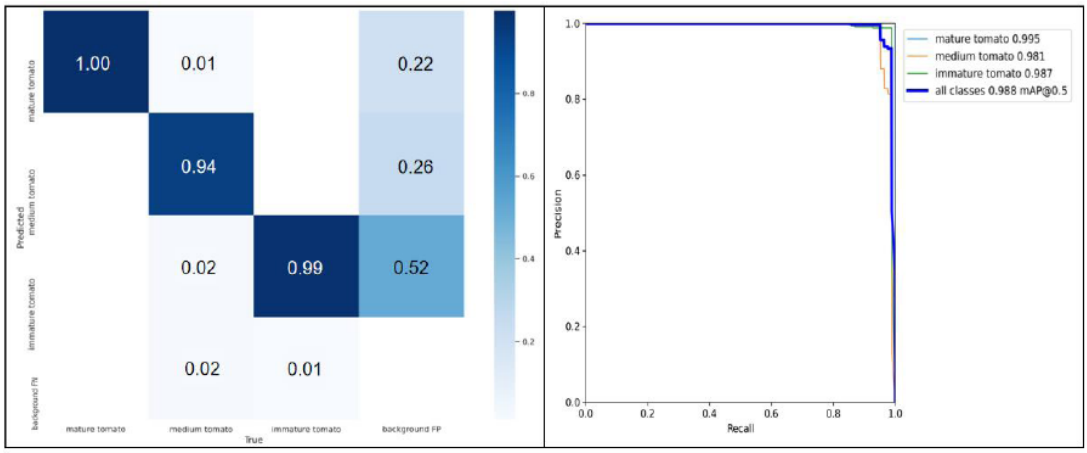


Figure 11: Yolov3 confusion matrix (left) and PR curve (right).

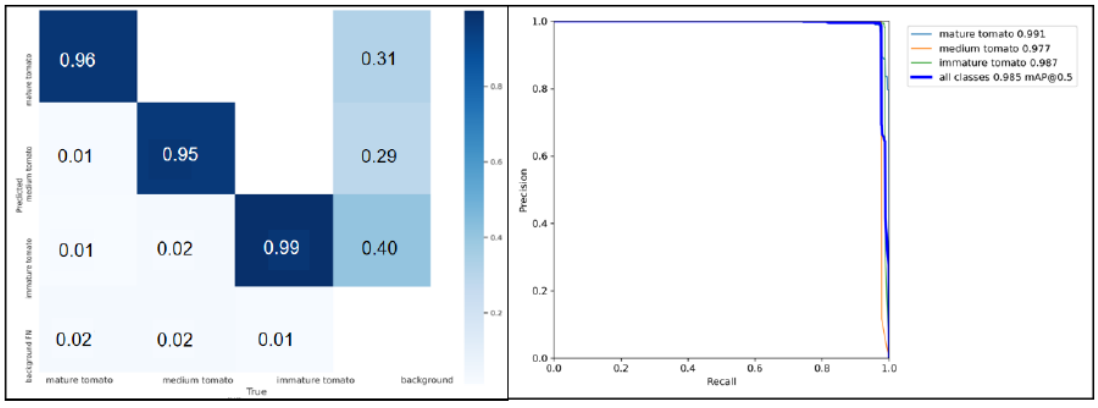


Figure 12: Yolov5s confusion matrix (left) and PR curve (right).

Table 4: Comparison of Yolov3 and Yolov5 Training and Testing Results

Method	R/%	F1/%	$MAP_{0.5}$/%	Recognition speed(CPU)/f·s^{-1}	Recognition speed(GPU)/f·s^{-1}	Model memory/MB
Yolov3	97.6	98.5	98.8	0.926	45	123.5
Yolov5s	97.6	98.3	98.5	1.299	96	14.5

rounds were more than 200, the loss values both decreased to below 0.2. It can be seen that the training accuracy of the two methods is high. Finally, import the trained .pt weight file into the image display of the depth camera to check the detection speed and recognition effect.

Table **4** shows the comparison of Yolov3 and Yolov5 training and testing results. Through the comparison of deep learning methods, it can be seen that the training accuracy of Yolov3 and Yolov5 are both high, and the MAP values both reach more than 98%. In contrast, when GPU tests the recognition speed of the model, Yolov5s recognizes 40 frames more than Yolov3 per second. The memory of Yolov5s model is also relatively small by more than 100MB.

4.3. Improved Yolov5 VS Yolov5

Combined with 1000 labeled tomato fruit pictures, the optimized Yolov5 model with a lightweight structure is used for training and testing. Table **5** shows the detection number and AP of Yolov5s and improved Yolov5s for tomato fruits with multiple growth periods.

It can be seen from the table that the recognition accuracy of the lightweight improved YOLOv5s is roughly the same as that of the original YOLOv5s. The

Table 5: Detection Number and AP of Multi-Growth Period Tomato Fruit by Yolov5s and Improved Yolov5s

Multiple Growth Periods	Mature Tomato Amount		Medium Tomato Amount		Immature Tomato Amount	
Detection algorithm	Yolov5s	Improved Yolov5s	Yolov5s	Improved Yolov5s	Yolov5s	Improved Yolov5s
Number of actual data set	**184**	**184**	**343**	**343**	**84**	**84**
Number of detected data set	**182**	**182**	**335**	**334**	**83**	**83**
AP	**0.991**	**0.989**	**0.977**	**0.975**	**0.987**	**0.985**
$MAP_{0.5}$ (Yolov5s)	**0.985**					
$MAP_{0.5}$ (improved Yolov5s)	**0.983**					

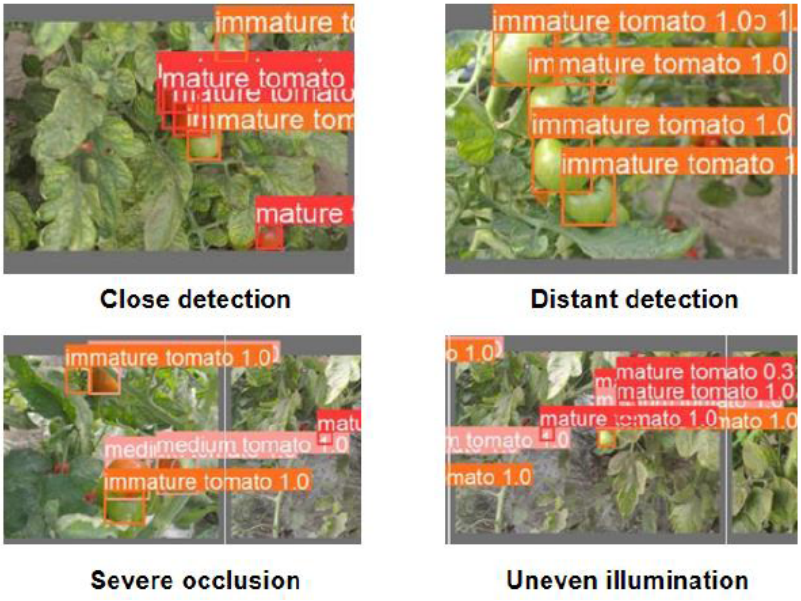


Figure 13: Verification of recognition effect of optimized Yolov5.

Table 6: Comparison of Improved Yolov5s, Yolov3 and Yolov5s Results

Method	R/%	F1/%	$MAP_{0.5}$/%	Recognition Speed(CPU)/f·s^{-1}	Recognition Speed(GPU)/f·s^{-1}	Model Memory/MB
Yolov3	97.6	98.5	98.8	0.926	45	123.5
Yolov5s	97.6	98.3	98.5	1.299	96	14.5
Improved Yolov5s	97.5	98.1	98.3	2.174	189	12.3

slight decrease in the accuracy of the improved YOLOv5 is mainly reflected in the detection of medium tomatoes. Figure **13** shows the model recognition effect of the optimized Yolov5 model by Intel real sense d435i visual processor.

The comparison with the results obtained by Yolov3, Yolov5, and improved Yolov5s is shown in Table **6**. In the table, model memory refers to the size of the model. It can be seen that although the recognition accuracy of the improved Yolov5 is slightly reduced, the $MAP_{0.5}$ remains above 98%, and the overall GPU recognition speed is increased by 90 frames per second. At the same time, the memory ratio of the model is further reduced. In a sense, the Yolov5s optimization model combined with the lightweight structure of mobilenetv3 greatly improves the speed of real-time recognition on the premise of ensuring accuracy. It is of great significance for rapid and accurate identification of tomatoes with different maturity.

5. CONCLUSION

In this work, an improved Yolov5s model is proposed to accurately and quickly identify multi growth period tomato fruits. Improved Yolov5 adopts the network structure of lightweight mobilenetv3, which embeds deep separable convolution, residual structure and attention mechanism. On the premise of ensuring the accuracy of more than 98%, the memory of the model is greatly reduced and the recognition speed is improved.

The main contributions of this paper are as follows: Yolov3 and Yolov5s have high recognition accuracy but slow detection speed for multi-growth period tomato fruit. Improved Yolov5 introduces a deep separable convolution structure and a linear bottleneck inverse residual structure, which greatly reduces the amount of convolution parameters and computation in the network overhead. The insertion of the attention mechanism highlights features and improves accuracy. Finally, this method effectively improves recognition speed and accuracy.

Limitations of the detection scheme in this paper: The optimized recognition algorithm has been verified by using the depth camera and software system, but it has not been integrated into the actual picking robot. Therefore, the robot picking will be further studied in the future.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (32171893), the National Natural Science Foundation of China (31971786) and the 2115 Talent Development Program of China Agricultural University.

REFERENCES

[1] Nicola S, Tibaldi G, Fontana E. Tomato production systems and their application to the tropics. [J]. Acta Horticulturae, 2009; 821: 27-33. https://doi.org/10.17660/ActaHortic.2009.821.1

[2] Fao. Faostat database [EB/OL]. http://www.fao.org/faostat/en/#data/QC.

[3] TX Zheng, MZ Jiang, MC Feng. Vision based target recognition and location for picking robot: a review[J]. Chinese J. Scientific Instrument, 2021; 09(42): 28-51.

[4] Arefi A, Motlagh A. Development of an expert system based on wavelet transform and artificial neural networks for the ripe tomato harvesting robot [J]. Australian Journal of Crop Science, 2013,5(7): 699-705.

[5] C Hu, X Liu, Z Pan, P LI. Automatic Detection of Single Ripe Tomato on Plant Combining Faster R-CNN and Intuitionistic Fuzzy Set [J]. IEEE Access, 2019; 7: 154683-154696. https://doi.org/10.1109/ACCESS.2019.2949343

[6] Payne A, Walsh K, Subedi P. Estimating mango crop yield using image analysis using fruit at 'stone hardening' stage and night time imaging.[J]. Computers and Electronics in Agriculture, 2014,100:160-167. https://doi.org/10.1016/j.compag.2013.11.011

[7] Sa I, Ge Z, Dayoub F. Deep Fruits: A Fruit Detection System Using Deep Neural Networks [J]. Sensors, 2016,16(8): 1222. https://doi.org/10.3390/s16081222

[8] WK Jia, YY Tian, R Luo, ZH Zhang, J Lian, YJ Zheng. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot [J]. Computers and Electronics in Agriculture, 2020,172. https://doi.org/10.1016/j.compag.2020.105380

[9] HW Kang, C Chen. Fast implementation of real-time fruit detection in apple orchards using deep learning[J]. Computers and Electronics in Agriculture, 2020,168 https://doi.org/10.1016/j.compag.2019.105108

[10] Koirala A, Walsh KB, Wang Z. Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'Mango YOLO' [J]. Precision Agriculture, 2019; 20(6): 1107-1135. https://doi.org/10.1007/s11119-019-09642-0

[11] G Liu, JC Nouaze, Mbouembe PL Touko, JH Kim. Yolo-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3 [J]. Sensors, 2020; 7(20): 2145. https://doi.org/10.3390/s20072145

[12] Gai RCNY. A detection algorithm for cherry fruits based on the improved YOLO-v4 model [J]. Neural Comput & Applic, 2021. https://doi.org/10.1007/s00521-021-06029-z

[13] Antihus Hernández Gómez, Guixian HU, Jun Wang, Annia García Pereira. Evaluation of tomato maturity by electronic nose [J]. Computers and Electronics in Agriculture, 2006; 54(1): 44-52. https://doi.org/10.1016/j.compag.2006.07.002

[14] S Lian, L Li, W Tan, L Tan. Research on Tomato Maturity Detection Based on Machine Vision [J]. The International Conference on Image, Vision and Intelligent Systems, 2022; 813. https://doi.org/10.1007/978-981-16-6963-7_60

[15] Bargoti S, Underwood JP. Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards[J]. Journal of Field Robotics, 2017; 34(6): 1039-1060. https://doi.org/10.1002/rob.21699

[16] Joseph Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement [J]. arXiv preprint arXiv, 2018;1804: 02767.

[17] A Bochkovskiy, CY Wang, H Liao. Yolov4: optimal speed and accuracy of object detection [J]. arXiv preprint arXiv, 2020,10934.

[18] Hongchun QU, Min Sun. A lightweight network for mummy berry disease recognition [J]. Smart Agricultural Technology, 2022; 100044(2). https://doi.org/10.1016/j.atech.2022.100044

[19] Howard A, Sandler M, Chu G, *et al*. Searching for MobileNetV3: IEEE/CVF International Conference on Computer Vision[C], Seoul, 2019. https://doi.org/10.1109/ICCV.2019.00140

[20] Sandler M, Howard A, Zhu M, *et al*. MobileNetV2: Inverted Residuals and Linear Bottlenecks: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)[C], Salt Lake City, UT, 2018. https://doi.org/10.1109/CVPR.2018.00474

[21] Wen C, Wen J, LI J, *et al*. Lightweight silkworm recognition based on Multi-scale feature fusion [J]. Computers and Electronics in Agriculture, 2022; 200: 107234. https://doi.org/10.1016/j.compag.2022.107234

[22] J Hu, L Shen, S Albanie, G Sun, E Wu. Squeeze-and-Excitation Networks [J]. IEEE Trans. Pattern Anal. Mach. 2022,42: 2011-2023. https://doi.org/10.1109/TPAMI.2019.2913372

[23] Andriyanov N. Intelligent System for Estimation of the Spatial Position of Apples Based on YOLOv3 and Real Sense Depth Camera D415 [J]. Symmetry, 2022; 14(1): 148. https://doi.org/10.3390/sym14010148

Received on 09-09-2022 Accepted on 16-11-2022 Published on 25-11-2022

DOI: https://doi.org/10.31875/2409-9694.2022.09.06