High-efficiency tea shoot detection method via a compressed deep learning model

Yatao Li[1], Leiying He[1,2], Jiangming Jia[1,2], Jianneng Chen[1,2], Jun Lyu[3], Chuanyu Wu[1,2*]

(1. *Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou 310018, China*;
2. *Key Laboratory of Transplanting Equipment and Technology of Zhejiang Province, Hangzhou 310018, China*;
3. School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: Achieving high-efficiency and accurate detection of tea shoots in fields is essential for tea robotic plucking. A real-time tea shoot detection method using the channel and layer pruned YOLOv3-SPP deep learning algorithm was proposed in this study. First, tea shoot images were collected and data augmentation was performed to increase sample diversity, and then a spatial pyramid pooling module was added to the YOLOv3 model to detect tea shoots. To simplify the tea shoot detection model and improve the detection speed, the channel pruning algorithm and the layer pruning algorithm were used to compress the model. Finally, the model was fine-tuned to restore its accuracy, and achieve the fast and accurate detection of tea shoots. The test results demonstrated that the number of parameters, model size, and inference time of the tea shoot detection model after compression reduced by 96.82%, 96.81%, and 59.62%, respectively, whereas the mean average precision of the model was only 0.40% lower than that of the original model. In the field test, the compressed model was deployed on a Jetson Xavier NX to conduct the detection of tea shoots. The experimental results demonstrated that the detection speed of the compressed model was 15.9 fps, which was 3.18 times that of the original model. All the results indicate that the proposed method could be deployed on tea harvesting robots with low computing power to achieve high efficiency and accurate detection.

Keywords: deep learning, tea shoot detection, model compression, high-efficiency
DOI: 10.25165/j.ijabe.20221503.6896

Citation: Li Y T, He L Y, Jia J M, Chen J N, Lyu J, Wu C Y. High-efficiency tea shoot detection method via a compressed deep learning model. Int J Agric & Biol Eng, 2022; 15(3): 159–166.

1 Introduction

According to the Food and Agriculture Organization of the United Nations (FAO), world tea production (black, green, instant, and others) reached 5.73 million t and tea consumption reached 5.5 million t in 2016[1]. The traditional manual tea plucking method is not only inefficient but also requires a great deal of labor, which leads to a high-cost problem. The research and development of automated tea harvesting machines is an effective approach to solving such a problem[2,3]. However, to harvest tea, current machines usually use blades to cut the top leaves. Thus, the quality of the harvested tea is low; hence, these machines cannot replace the traditional manual harvesting approach. Additionally, there is an urgent need to design a high-quality tea harvesting robot that selectively plucks tea leaves to completely replace manual harvesting methods[4,5]. The accurate and rapid detection of tea shoots in fields is a primary and difficult task.

In fields, the detection of tea shoots is challenging, and a great deal of research on the automatic detection of tea shoots has been conducted by scholars. Tang et al.[6] performed image segmentation on tea shoot images against a complex background by combining super green features and an improved OSTU algorithm to guide the identification of the tea shoots by tea harvesting robots. Zhang et al.[7] used an image process algorithm combined with Bayesian discrimination to achieve the identification of fresh tea leaves and harvest status, thereby providing a basis for the automated management of tea gardens. Karunasena et al.[8] presented a new method for tea shoot detection using a cascade classifier, which carried out the detection of tea buds by combining histogram of oriented gradient features and support vector machine classification. Zhang et al.[9] proposed a method based on an improved watershed algorithm for the identification and segmentation of tea sprouts and used piecewise linear transformation to enhance the differentiation degree of old tea sprouts and the segmentation accuracy. However, most of the tea images in these studies were obtained under the conditions of a simple background or constant light. The tea garden environment has a complex background and large lighting changes. Hence, the method for identifying tea shoots based only on color and shape features is inappropriate in fields[10]. With the rapid development of deep learning technology, increasing numbers of deep learning algorithms are being used for the target recognition and detection tasks of agricultural robots in unstructured environments[11]. Yang et al.[12] used the improved you only look once (YOLO) network to train a tea shoot detection model, and the accuracy of the trained model was over 90% for the verification set. Chen et al.[13] used a faster region-based convolutional neural network to detect tea shoots in a tea garden, and achieved a precision of 79%

Received date: 2021-07-15 **Accepted date:** 2021-10-18
Biographies: Yatao Li, PhD candidate, research interest: agricultural robot vision, Email: 201910501019@mails.zstu.edu.cn; **Leiying He**, PhD, Associate Professor, research interest: agricultural robot vision, Email: hlying@zstu.edu.cn; **Jiangming Jia**, PhD, research interest: intelligent agricultural equipment, Email: jarky@zstu.edu.cn; **Jun Lyu**, Postgraduate, research interest: image identification, Email: lv_jun@zstu.edu.cn; **Jianneng Chen**, PhD, Professor, research interest: intelligent agricultural equipment, Email: jiannengchen@zstu.edu.cn.
***Corresponding author: Chuanyu Wu**, PhD, Professor, research interest: intelligent agricultural equipment and robotics. Faculty of Mechanical Engineering & Automation, Zhejiang Sci-Tech University, Hangzhou 310018, China. Tel: +86-13666698922, Email: cywu@zstu.edu.cn.

and a recall of 90%.

Because of the powerful feature extraction ability and robustness of the convolutional neural network, deep learning-based methods have achieved good performance in the field of object detection[14,15]. However, in the process of extracting features using deep learning (GoogleNet[16] and EfficientDet[17]), the number of network layers continues to deepen, and the number of model parameters continues to increase, which makes deep learning-based approaches time-consuming and computationally expensive. This series of reasons makes such approaches difficult to deploy on small mobile terminals, which seriously affects the deployment and application of a tea shoot detection model on small mobile terminals and restricts the development of tea harvesting robots, to some extent. To facilitate the implementation of deep learning-based methods in agricultural and industrial fields, the model needs to be compressed before it is deployed on a low computing power platform[18,19]. Reducing the number of model parameters, reducing the size of the model, and improving the real-time performance of detection while maintaining the accuracy of the model have become the research focus. Han et al.[20] proposed a weight pruning method that deletes unimportant weights to learn only the important connections. It greatly reduces the number of model parameters without reducing the accuracy of the model. Zhang et al.[21] proposed a channel pruning method that enforces the channel-level sparsity of convolutional layers by imposing L1 regularization on channel scaling factors, and prunes less informative feature channels so that the model can achieve the effect of real-time target detection using an unmanned aerial vehicle. Wu et al.[22] used the channel pruning method to prune the apple flower detection model, and achieved the fast and accurate detection of apple flowers in a natural environment. Wang et al.[23] used the channel pruning method to compress the apple detection model, which provides a reference for the development of portable mobile fruit thinning terminals. Xu et al.[24] proposed a layer pruning method that uses a fusible residual convolutional block (ResConv), which converts the convolutional layers of the network into a ResConv with a layer scaling factor. It greatly reduces network parameters while ensuring accuracy.

Although the aforementioned research can effectively reduce the size of the model, it is applied infrequently in industrial and agricultural fields. Moreover, model compression still needs improvement in actual applications. The aim of this study was to achieve the real-time and accurate detection of tea shoots in fields to facilitate the deployment of the detection model on tea harvesting robots. The study includes the following parts: 1) a tea shoot dataset was established and expanded, and a deep learning algorithm was used for tea shoot detection; 2) the trained model was compressed on the basis of the channel pruning and layer pruning algorithms; 3) the compressed model was deployed on a Jetson Xavier NX edge box to detect tea shoots in fields to verify the effectiveness of the proposed algorithm.

2 Materials and methods

2.1 Overview of the method

The overview of the technical route of this method is shown in Figure 1. First, tea shoot images in a tea garden were acquired using a digital camera, and the labeled data were expanded to establish a tea detection dataset. Second, the YOLOv3 network was used to detect tea shoots and the spatial pyramid pooling (SPP) module was added to achieve high-precision detection. Third, sparse training, channel pruning, layer pruning, and model fine-tuning were performed on the trained model to achieve model compression while maintaining the accuracy of the model. Finally, the compressed model was deployed on a Jetson Xavier NX to conduct tea shoot detection experiments in fields.

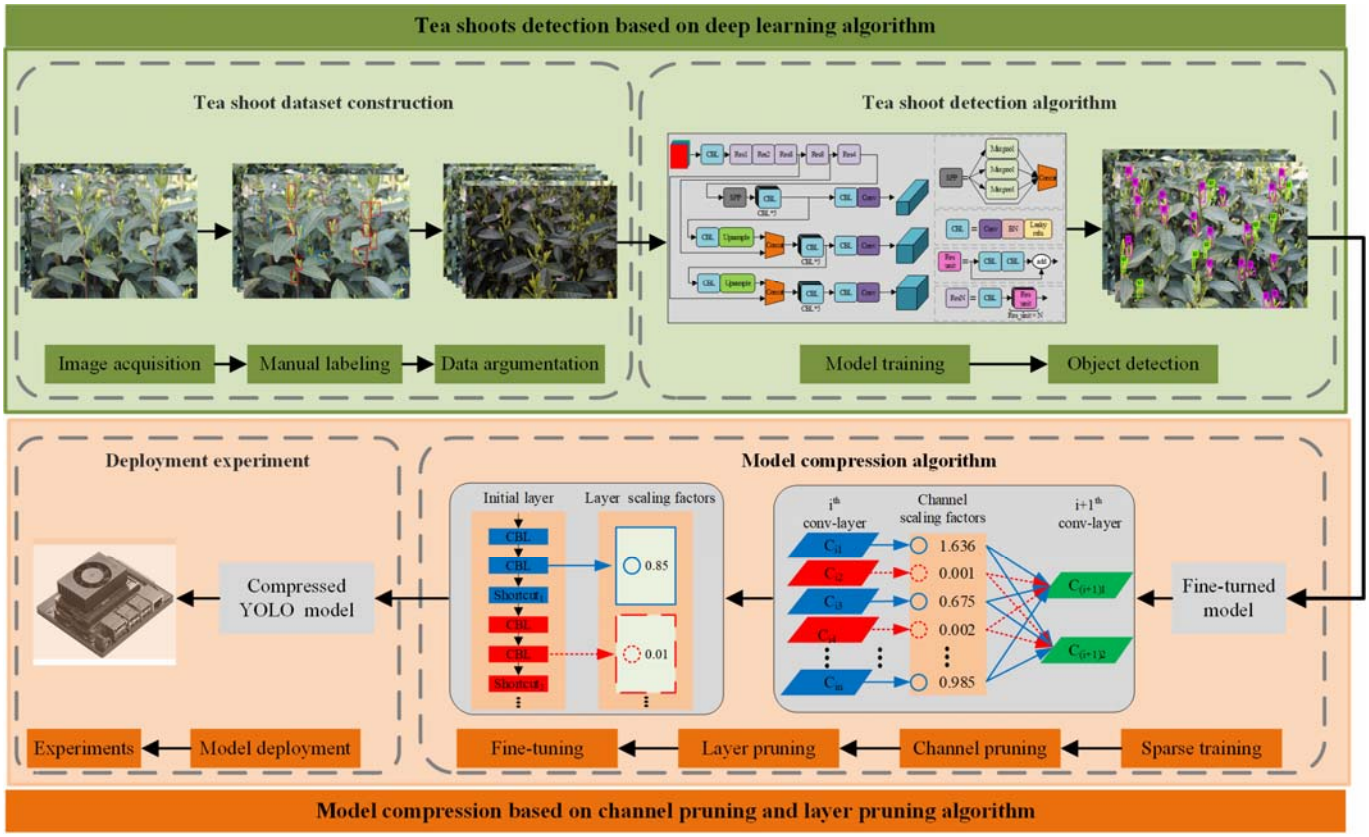


Figure 1 Overall technical route of the proposed tea shoot detection algorithm

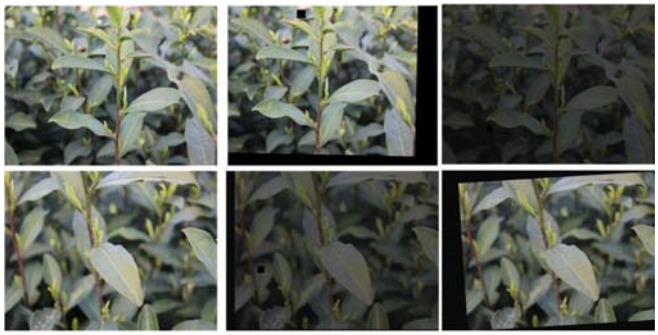
2.2 Image acquisition and dataset construction

All images were acquired at the Academy of Agricultural Sciences, Hangzhou City, Zhejiang Province, China, in April 2020. The variety of the acquired tea was Longjing 43 (LJ-43), which is a famous high-quality tea. A scene of the tea garden is shown in Figure 2a, the tea tree row spacing was approximately 1.5 m and the tree height was approximately 1.2 m. Figure 2b shows a self-built image acquisition mobile platform, which comprised an industrial camera and mobile platform. The camera was installed under the mobile platform approximately 1.5 m from the ground, and the installation angle of the optical axis of the camera was 45°-60° to the horizontal plane to reduce the mutual occlusion of the tea shoots. Additionally, to increase the diversity of sample images in different illumination conditions, images were collected during different weather conditions (sunny, and cloudy) from 7:00 am to 5:00 pm. Finally, a total of 6248 images were collected in the tea garden with a resolution of 1280 pixels×720 pixels; the examples are shown in Figure 2c.

a. Experimental tea garden used in this study b. Self-built mobile platform for image acquisition c. Tea shoot images collected in different weather and time

Figure 2 Tea in the tea garden

To achieve better detection results, tea shoots in images were labeled using two categories: LF (captured from the front view) and LS (captured from the side view), as in our previous study[10]. What's more, considering that the complex field environment could easily cause part of the tea shoot to be occluded, an object whose occlusion area was greater than 75% was not labeled. Finally, a total of 90 753 tea shoots were labeled from a dataset containing 6248 images. Then all the annotated images were divided into three sets: a training set containing 3748 images, a validation set containing 1250 images, and a test set containing 1250 images. The training images were randomly obtained from the independent and uniform sampling of the entire dataset. All images were unequal, thereby ensuring the generalization ability of the tea shoot dataset and reliability of the later evaluation standards. Because of varying illumination conditions, different growth angles, and the complex crown structure of tea trees in the field, whether the neural network could process the images collected in different environments depends on the integrity of the training dataset. Hence, the trained images were expanded to enhance the richness of the experimental dataset, which included adding noise, changing the brightness, simulating occlusion, and performing the affine transformation. Then these processes were randomly combined for data augmentation, as shown in Figure 3. The complete tea shoot dataset of LJ-43 is listed in Table 1.



a. Original image b. Expanded images randomly combined, for example, by changing the brightness, adding noise, and rotating the image

Figure 3 Data augmentation example images

Table 1 Complete tea shoot dataset

Datasets	Original training set	Expanded training set	Valid set	Test set
LJ-43	3748 images	14 992 images	1250 images	1250 images

2.3 Tea shoot detection based on YOLO

Compared with the two-stage detector (e.g., faster region-based convolutional neural network and R-FCN), YOLO, as a one-stage detector, has a fast detection speed, so it has been widely used in agriculture and industry. The YOLOv3 network evolved from the YOLOv1 and YOLOv2 networks[25], which is simpler than YOLOv4 and YOLOv5, and its detection accuracy meets the needs of tea shoot detection, so it is adopted in this study. It uses multi-scale prediction to detect bounding boxes on different scales, which makes YOLOv3 more effective for detecting small targets than previous networks. Because of the dense growth of tea shoots in a tea garden, the size of tea shoots at different distances varies greatly. To integrate the local features and global features of tea shoots, the SPP module was added by referring to the idea of a spatial pyramid[26].

The training platform included a workstation with an Intel i7-9800x (3.80 GHz) eight-core CPU, four NVIDIA RTX2080Ti (1620 MHz) GPUs, and 64 GB of memory running on the Ubuntu 16.04 system. The hyperparameters for model training are listed in Table 2. The image input size was 416×416 pixels and the initial learning rate of weights was 2.5×10^{-4}. To speed up the training process and prevent overfitting, the momentum parameter was chosen to be 0.90, and the weight attenuation coefficient was 0.0005. The total training iterations of the model was 100 000, and the learning rate decreased to 2.5×10^{-5} after 80 000 iterations and 2.5×10^{-6} after 90 000 iterations.

Table 2 Hyperparameters for model training

Parameters	Value
Input size/pixels	416×416
Intimal learning rate	2.5×10^{-4}
Momentum	0.9
Weight attenuation coefficient	0.0005
Iterations	100 000

2.4 Model compression

Because of the deep level of the YOLOv3-SPP network and

a large number of model parameters, the real-time performance of tea shoot detection is still insufficient on mobile terminals. To reduce the size of the model and facilitate deployment on a low computing power platform, such as Raspberry Pi, the ARM platform, or Jetson Xavier NX, it is essential to compress the model[27]. Generally, there are three steps for model compression: sparse training, model pruning, and model fine-tuning.

2.4.1 Sparse training

The first task of model compression is to make the trained tea shoot detection model sparse. There are many sparse training approaches, and the most widely used approach, L1 regularization based on the L1 norm was adopted in this study. As a result of adding the L1 regularization term using the γ coefficient of the batch normalization (BN) layer to the loss function of the model, the γ coefficient of the BN layer becomes sparse, hence, the model is adjusted in the direction of the sparse structure. The optimizing objective of sparse training is given by

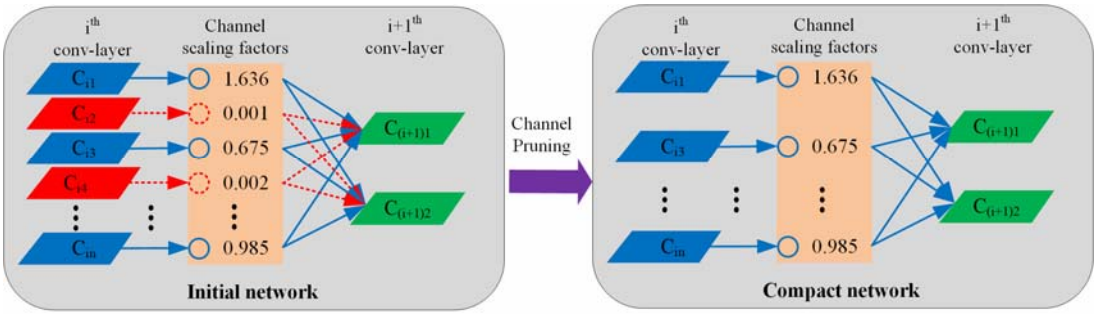
$$\text{Loss} = \sum_{(x,y)} l(h(x,w),y) + \lambda \sum_{i=0}^{L} |m_i| \quad (1)$$

where, x and y denote the training input and target, respectively; w denotes the trainable weights; l denotes the training loss of the network output and target; $|m_i|$ denotes the absolute value of the i-th layer scaling factor; L denotes the total number of layers in the network; λ denotes the sparsity rate, which is used to balance the two sum terms.

The γ coefficient is updated accordingly in the reverse transfer process of sparse training, and the value of a large number of γ coefficients tends to 0. This process can be regarded as feature selection occurring in the intermediate layers of deep networks, where only channels with non-negligible scaling factors are chosen. After sparse training, the accuracy of the model is likely to decrease and the loss value is likely to increase. Fortunately, the accuracy of the model can be restored by fine-tuning the model later.

2.4.2 Model pruning

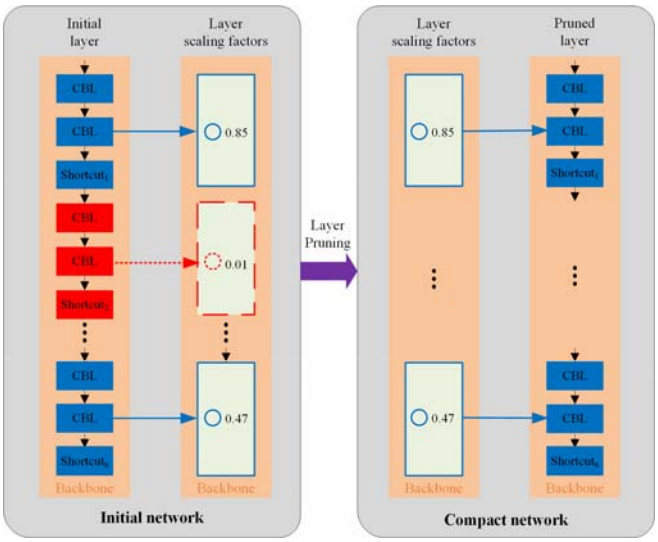
The model pruning operation on the sparse model is mainly divided into two parts: channel pruning and layer pruning. The principle of the channel pruning algorithm is shown in Figure 4. The contribution score of the channel to the network is evaluated using the γ coefficient of the BN layer. Then the channels with a high contribution score are retained according to the pruning ratio and the distribution of the γ coefficient, and channels with a low contribution score are deleted. When connecting the channels between the layers, the neurons of the channels with lower contributions do not participate in the connection, and a simplified model that takes up less storage space is generated.



Note: Conv means Convolution; blue solid line means channel with a high contribution score; red dotted line means channel with a low contribution score.

Figure 4 Principles of channel pruning

To further compress the model, layer pruning is performed on the basis of channel pruning. The layer pruning algorithm is derived from the channel pruning algorithm and pruning is mainly performed on the shortcut layer. The principles of layer pruning are shown in Figure 5. CBL is a module comprising a convolution (Conv) layer, BN layer, and leaky rectified linear unit (leaky-ReLU) activation function. The γ coefficient of the CBL module connected by each shortcut is evaluated according to the contribution score of the layer to the network. Then high contribution layers are retained and low contribution layers are deleted according to the pruning rate and the order of the γ mean. To ensure the integrity of the structure of the pruned model, the previous two CBL modules are also pruned accordingly when layer pruning is performed on the shortcut layer; hence, there are three layers (two CBLs and one shortcut) for pruning for each shortcut layer.



Note: blue solid line means high contribution layer and red dotted line means low contribution layer.

Figure 5 Principles of layer pruning

2.4.3 Model fine-tuning

After the model is made sparse and pruned, although the number of parameters is greatly reduced and the model size is effectively reduced, the accuracy of the model also drops significantly. To overcome the problem of the excessive accuracy loss of the model after pruning, it is necessary to train the data again on the pruned model to fine-tune the model.

Model fine-tuning needs to use the pruned network to readjust the pruned weights of the model, and it needs to set the compressed model as a pre-training model and perform a new round of training. Because the model has undergone channel pruning and layer pruning, the parameters, and size of the model have been greatly reduced. Hence, the time consumption of fine-tuning the pruned model is greatly reduced. After fine-tuning, the γ coefficient of

the model is redistributed and the accuracy of the detection model can be effectively restored.

3 Results and analysis

In this study, two evaluation indices were used to evaluate the performance of the tea shoot detection network: mean AP (mAP) and recall. Intersection over Union (IoU)≥0.5 indicates a true case; IoU<0.5 indicates a false positive case; IoU=0 indicates a false negative case. IoU is calculated as,

$$\mathrm{IoU}=\frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \tag{2}$$

where, S_1 is the area of the detected bounding box; S_2 is the area of the real bounding box. The value range of the mAP and Recall is [0, 1], and calculated as the follows:

$$\mathrm{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{3}$$

$$\mathrm{mAP}=\frac{\sum_1^c \mathrm{AP}_c}{c}\times 100\% \tag{4}$$

$$\mathrm{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\times 100\% \tag{5}$$

where, TP, FP, and FN are the number of true positive cases, false positive cases, and false negative cases, respectively; c is the number of detection categories and AP is the average precision.

3.1 Evaluation of data augmentation

To assess the data augmentation effect on the tea shoot detection model, data before and after augmentation were trained. The final results are shown in Table 3. The results showed that the recall and mAP increased after augmentation, which demonstrates that data augmentation effectively expanded the richness of the samples, thereby improving the generalization ability and robustness of the tea shoot detection model. Image processing operations such as a random combination of affine transformation (translation and rotation), changing the brightness, adding noise, and adding cutouts were chosen to augment the data because these operations restore actual changes in the field environment as much as possible. For example, the affine transformation was used to simulate the change of the camera position and angle; the brightness was changed to simulate the change of sunlight; the noise was added because noise is unavoidable when a camera collects images in fields; cutout was used because the pastoral phenomenon is inevitably blocked by some tea shoots in fields. A random combination of these operations can increase the amount of data, but the amount of data was only tripled in this study, which was because the amount of data already met the demand for the detection accuracy of tea shoots, and it is easy to cause the overfitting phenomenon if too much data is expanded.

Table 3 Performance of data augmentation

Parameters	Before augmentation	After augmentation
mAP/%	76.66	90.01
Recall/%	80.55	83.99

3.2 Evaluation of model pruning

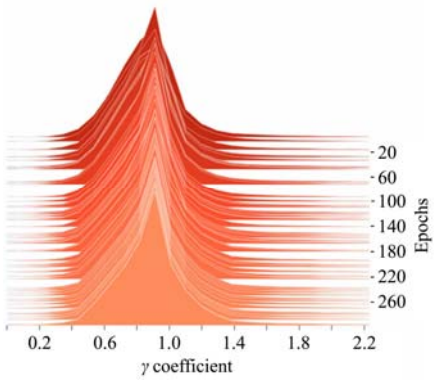
3.2.1 Results of sparse training

To determine the appropriate sparsity rate of the tea shoot detection model, different sparsity rate coefficients were selected for the experiments. The experimental results are listed in Table 4. The results showed that the mAP of the model decreased severely after λ=0.003, so the sparse rate was selected as 0.003 in this study.

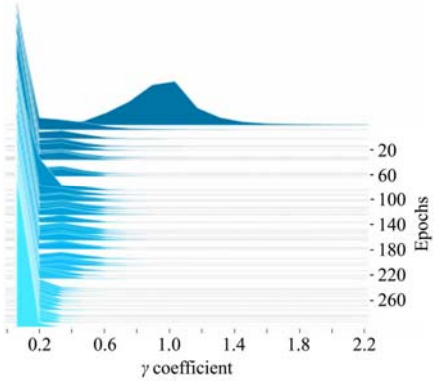
The γ coefficient distribution of the original tea shoot detection model is shown in Figure 6a, which was approximately a normal distribution as a whole. Then the sparse rate of 0.003 was selected for the sparse training of the tea shoot detection model. The sparse training results are shown in Figure 6b. The results showed that the γ coefficient distribution centers of all the BN layers gradually moved closer to 0 from the original 0.9, and the distribution became concentrated, which indicates that the model gradually became sparse, which verifies the effectiveness of sparse training. After 226 iterations of sparse training, the γ coefficient gradually stabilized, which indicates that sparse training was saturated.

Table 4 Performance of the different sparse rates

Sparse rate (λ)	mAP/%
0	90.01
0.001	88.79
0.002	73.53
0.003	86.59
0.004	70.91
0.005	66.58



a. Distribution of the γ coefficient before sparse training



b. Distribution of the γ coefficient after sparse training

Figure 6 Sparse training of the tea shoot detection model

3.2.2 Results of model pruning

After sparse training, channel pruning and layer pruning algorithms were used to prune the model network. First, channel pruning was performed on the sparse tea shoot detection model. To determine the appropriate channel pruning coefficient, different pruning coefficients were selected for the experiments, with a step size of 0.05. The experimental results are shown in Figure 7. The results showed that the mAP of the model dropped sharply and the parameter change rate became small after r=0.85, so the channel pruning coefficient was selected as 0.85 in this study. Finally, the model channel changes after the channel pruning algorithm are shown in Figure 8. A total of 22 033 channels in 70 layers were pruned, the minimum pruning channel for each layer

was 13, and the maximum number of pruning channels was 911. Moreover, Figure 8 shows that the channels in most convolutional layers were greatly reduced, which demonstrates the effectiveness of the channel pruning algorithm.

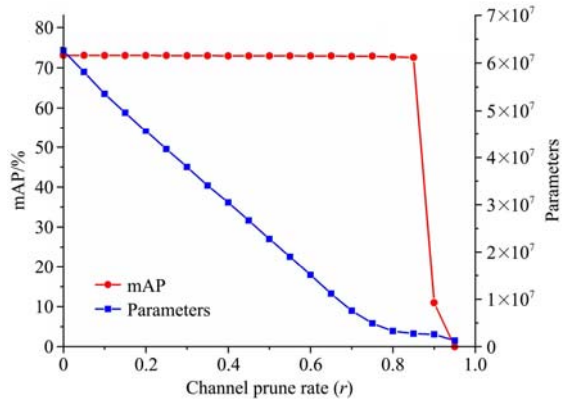


Figure 7 Influence of different channels' pruning coefficients on the tea shoot detection model

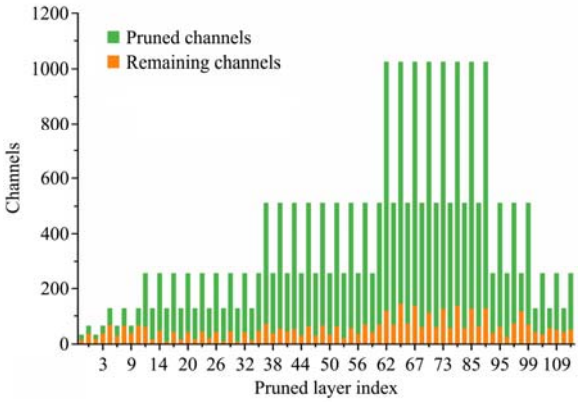


Figure 8 Changes in the number of channels of the layer processed by channel pruning

Channel pruning greatly reduced the parameters and size of the model. To better compress the model, layer pruning was performed on the model after channel pruning. The original model had a total of 23 shortcut layers for pruning. Different numbers of layers were selected for the layer pruning experiments to determine the appropriate number of layers for pruning. The experimental results are shown in Figure 9. The results showed that after 13 shortcut layers were pruned, the parameter quantity of the model steadily became smaller, and the mAP dropped sharply when 15 shortcut layers were pruned. To ensure that the accuracy did not drop too severely (<20%), the number of pruning layers was selected as 16 in the study, and the index number of the pruned shortcut layer at this time was [68, 74, 65, 71, 61, 55, 52, 49, 40, 43, 58, 46, 18, 15, 30, 27], where the shortcut layer with the first index number was pruned first because of its low contribution to the model. The remaining shortcut layers ([21, 24, 33, 36, 8, 11, 4]) and the corresponding CBL layers were retained.

3.2.3 Results of fine-tuning

Although the parameters and size of the tea shoot detection model after channel pruning and layer pruning were reduced, the accuracy of the model was also greatly reduced. To restore the accuracy of tea shoot detection, the model needed to be fine-tuned. Table 5 lists the changes in the main parameters of the model during the fine-tuning process. It can be found that the total parameters, model size, and inference time have substantially not changed after fine-tuning, whereas the mAP of the tea shoot detection model was essentially restored to that of the original model, which effectively and quickly detected tea shoots.

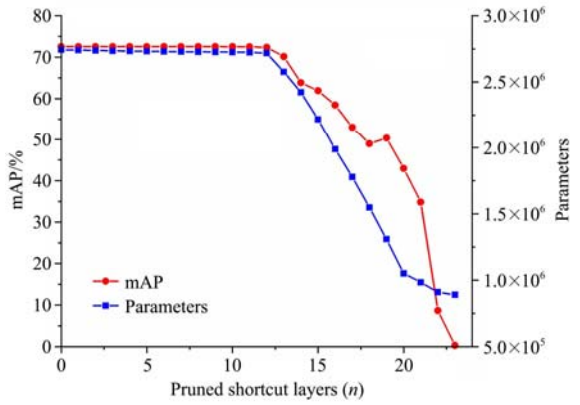


Figure 9 Changes in the mAP and parameters of the model processed by layer pruning

Table 5 Performance of model fine-tuning

Parameters	Before fine-turning	After fine-turning
Total parameters	1 989 221	1 989 221
mAP/%	58.41	89.61
Model Size/MB	8	8
Inference/s	0.0041	0.0042

3.2.4 Evaluation of the final model compression

The changes in model parameters, mAP, model size, and inference time after sparse training, channel pruning, layer pruning, and model fine-tuning are listed in Table 6. The results showed that the number of model parameters, model size, and inference time after compression was reduced by 96.82%, 96.81%, and 59.62%, respectively, while the mAP of the model was only reduced by 0.40%. All the results indicate that it is feasible to use this method to achieve the rapid and accurate detection of tea shoots.

Table 6 Performance of the model compression (λ=0.003, r=0.85, n=16)

Parameters	Initial model	After sparse	After pruning channels	After pruning layers	Final model
Total parameters	62 578 719	62 578 719	2 743 669	1 989 221	1 989 221
mAP/%	90.01	86.59	86.09	71.95	89.61
Model size/MB	250.5	250.5	11	8	8
Inference/s	0.0105	0.0095	0.0068	0.0041	0.0042

3.3 Comparison of tea shoot detection models

To evaluate the effectiveness of the proposed method for the detection of tender tea shoots, three object detection algorithms of the YOLO network were compared in this study: YOLOv3-tiny, YOLOv3, and YOLOv3-SPP. They all belong to the one-stage network, and all have a fast detection speed, which ensures the high detection efficiency of tea harvesting robots. The augmented training set of tea shoots was adopted to train the detection models on the basis of the four algorithms, and then the test set was used to evaluate the performance of the different detection algorithms on the server. The test results are listed in Table 7.

Table 7 Tea shoot detection results for different YOLO models

Model	Recall/%	mAP/%	Model size/MB	Detection speed/fps
YOLOv3-tiny	73.58	65.08	34.7	98.7
YOLOv3	83.75	85.97	246.3	45.2
YOLOv3-spp	83.99	90.01	250.5	47.8
Proposed method in this study	83.59	89.61	8.0	62.5

The test results showed that the recall of the four object detection algorithms were 73.58%, 83.75%, 83.99%, and 84.59%, respectively; the mAPs were 65.08%, 85.97%, 90.01%, and 89.61%, respectively; the model sizes were 34.7 MB, 246.3 MB, 250.5 MB, and 8.0 MB, respectively; and the detection speeds were 98.7 fps, 45.2 fps, 47.8 fps, and 62.5 fps., respectively. An analysis of the test results shows that the proposed method had a smaller model size for tea shoots than the other three comparison algorithms. In terms of the detection speed, although YOLOv3-tiny was faster than the proposed method, the mAP of the proposed method was 24.53% higher than that of YOLOv3-tiny, and the detection speed of the proposed method still met the real-time requirements. Moreover, the model size of the proposed method was small and could be deployed easily on small mobile terminals. The recall of YOLOV3-SPP was 0.40% higher than that of the proposed method, but the model size of the proposed method was 96.82% smaller than that of YOLOV3-SPP. Moreover, the mAPs of the two algorithms were not substantially different, which indicates that the proposed method was more cost-effective than YOLOv3-SPP.

4 Experiments and discussion

4.1 Model deployment experiments

To verify the effectiveness of the proposed method, the compressed tea shoot detection model was deployed on a small mobile terminal, that is, a Jetson Xavier NX (NVIDIA Inc, City, CA, USA), with 384 NVIDIA Volta CUDA cores and 48 tensor cores, which only provides 21TOPS of AI computing power. A RealSense L515 camera (Intel Inc, City, CA, USA) was used to capture RGB images up to 1280×720 pixels at 30 fps, which communicated with the Jetson Xavier NX via USB. The experimental system and detection results are shown in Figure 10. The uncompressed model and compressed model were deployed on the Jetson Xavier NX for the tea shoot detection experiments. The results showed that the size of the model after compression reduced from 250.5 MB to 8.0 MB, which resulted in an increase in the detection speed from 5.0 fps to 15.9 fps, which was 3.18 times that of the original model. Hence, the proposed model compression algorithm was effective, and the compressed model met the detection requirements of the field mobile terminal.

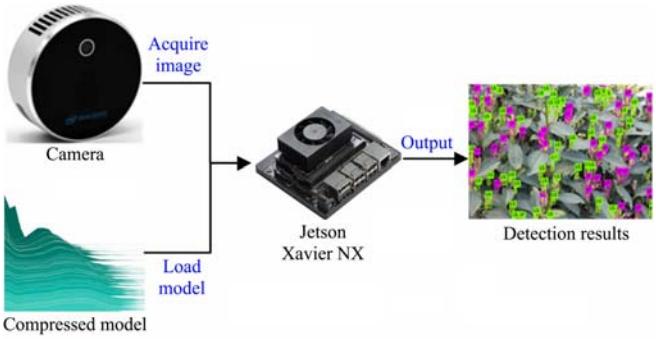
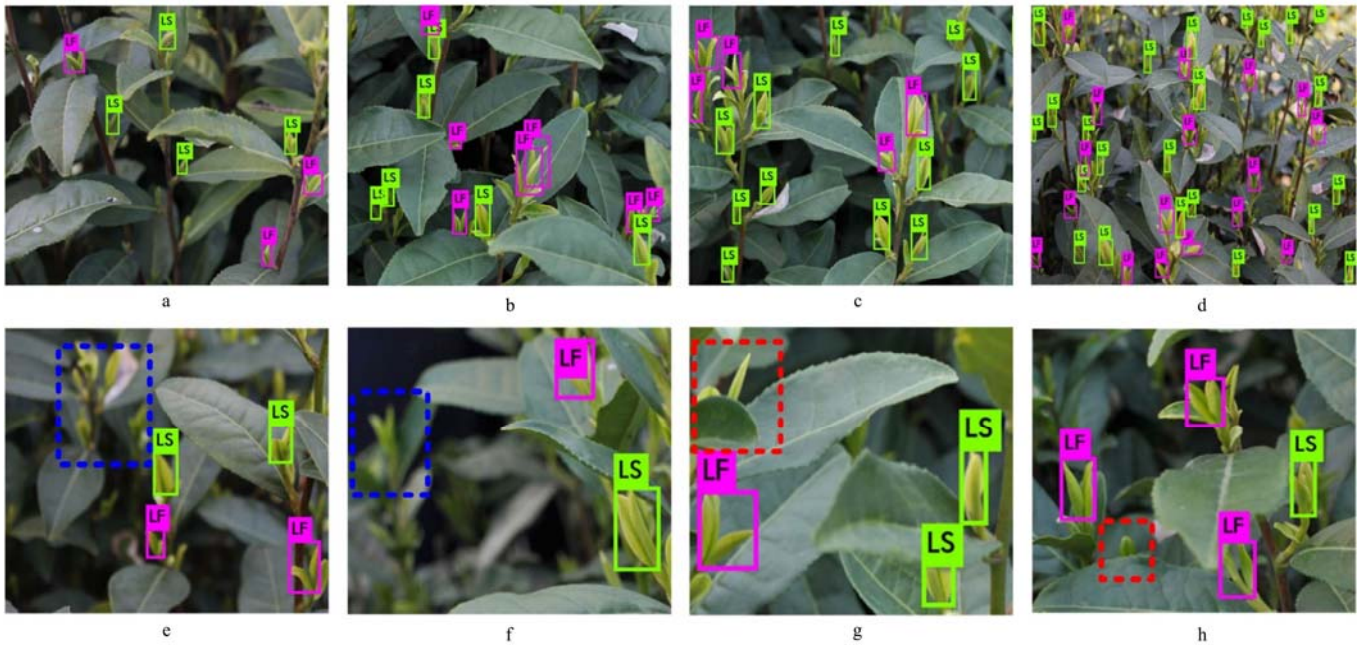


Figure 10 Field tea shoot detection experimental system and detection results

4.2 Discussion of the tea shoot detection experiments

The proposed tea shoot detection algorithm quickly and effectively detected the targets in the tea images with different growth densities and shooting distances (Figures 11a-11d), but there was still a small number of tea shoots that were not detected in actual field experiments. The main reasons for inaccurate detection are as follows:

1) Because of the different distances between the tea shoots and the camera in different positions, the image was inevitably blurred when the image was collected, which led to the difficulty of artificial labeling. To unify the labeling standards, the fuzzy tea shoots were not labeled. This may have resulted in the failure of the detection of partially blurred tea shoots, as shown by the blue dashed box in Figures 11e-11f.



a-d represent the detection results with different growth densities and shooting distances, e-f represent the missed detection results caused by blur, g-h represent the missed detection results caused by large occlusion

Note: LF means captured from the front view; LS means captured from the side view.

Figure 11 Detection results of the experiment in fields

2) Because of the dense growth of tea leaves and the complex crown structure in the actual field environment, it is inevitable that a large number of tea shoots would be blocked. This caused great difficulties in the task of detecting tea shoots and easily led to

detection failures. Image augmentation through cutout processing could have increased the detection success rate of partially occluded shoots. However, there were still a small number of shots blocked by a large area, which resulted in the tea shoot features learned by the deep network being lost or not detected correctly, as shown by the red dotted line in Figures 11g-11h.

3) The images of the tea shoot detection model dataset were collected in mid-March, so the established model had a high detection accuracy for tea images at that time. In early April, the image characteristics of the tea shoots changed slightly over time, so the accuracy of model detection declined. To solve this problem, transfer learning can be introduced to transfer the model and establish a tea shoot detection model for the current period, which would improve the detection results.

5 Conclusions

In this study, a high-efficiency tea shoot detection method that can be deployed on a low computing power platform was proposed on the basis of the YOLOv3-SPP network and a model compression algorithm. Experiments were conducted using the compressed model on the Jetson Xavier NX to verify the effectiveness of the proposed method. Three conclusions can be drawn, as follows.

1) The trained tea shoot detection model was compressed using sparse training, channel pruning, layer pruning, and model fine-tuning, where a total of 22 033 channels and 16 shortcut layers were pruned during the channel pruning and layer pruning process. After compression, the number of model parameters, model size, and inference time were reduced by 96.82%, 96.81%, and 59.62%, respectively, whereas the mAP of the model was only reduced by 0.40%. All the results indicate that it is feasible to use this method to achieve the rapid and accurate detection of tea shoots.

2) The compressed tea shoot detection model was deployed on a Jetson Xavier NX with a detection speed of 15.9 fps, which was 3.18 times that of the original model, thereby demonstrating that the proposed method could be used for the detection task of a tea harvesting robot with low computing power in fields.

Acknowledgements

This work was financially supported by the China Agriculture Research System of the Ministry of Finance and the Ministry of Agriculture and Rural Affairs and the National Natural Science Foundation of China (Grant No. 51975537). The authors also acknowledge the Hangzhou Academy of Agricultural Sciences for their assistance in conducting field experiments.

[References]

[1] Food and Agriculture Organization of the United Nations (FAO). Current market situation and medium term outlook: FAO. Available: http://www.fao.org/3/BU642en/bu642en.pdf. Accessed on [2020-11-23].

[2] Han Y, Xiao R H, Song Y Z, Ding Q W. Design and evaluation of tea-plucking machine for improving quality of tea. Applied Engineering in Agriculture, 2019; 35(6): 979–986.

[3] Yang H, Chen L, Ma Z, Chen M, Zhong Y, Deng F, et al. Computer vision-based high-quality tea automatic plucking robot using Delta parallel manipulator. Computers and Electronics in Agriculture, 2021; 181: 105946. doi: 10.1016/j.compag.2020.105946.

[4] Motokura K, Takahashi M, Ewerton M, Peters J. Plucking motions for tea harvesting robots using probabilistic movement primitives. IEEE Robotics and Automation Letters, 2020; 5(2): 3275–3282.

[5] Yuan J. Research process analysis of robotics selective harvesting technologies. Transactions of the CSAM, 2020; 51(9): 1–17. (in Chinese)

[6] Tang Y, Han W, Hu A, Wang W. Design and experiment of intelligentized tea-plucking machine for human riding based on machine vision. Transactions of the CSAM, 2016; 47(7): 15–20. (in Chinese)

[7] Zhang L, Zhang H, Chen Y, Dai S, Li X, Kenji I, et al. Real-time monitoring of optimum timing for harvesting fresh tea leaves based on machine vision. Int J Agric & Biol Eng, 2019; 12(1): 6–9.

[8] Karunasena G, Priyankara H. Tea bud leaf identification by using machine learning and image processing techniques. International Journal of Scientific & Engineering Research, 2020; 11(8): 624–628.

[9] Zhang L, Zou L, Wu C, Jia J, Chen J. Method of famous tea sprout identification and segmentation based on improved watershed algorithm. Computers and Electronics in Agriculture, 2021; 184: 106108. doi: 10.1016/j.compag.2021.106108.

[10] Li Y, He L, Jia J, Lyu J, Chen J, Qiao X, et al. In-field tea shoot detection and 3D localization using an RGB-D camera. Computers and Electronics in Agriculture, 2021; 185: 106149. doi: 10.1016/j.compag.2021.106149.

[11] Kamilaris A, Prenafeta-Boldu F X. Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 2018; 147(1): 70–90.

[12] Yang H, Chen L, Chen M, Ma Z, Deng F, Li M, et al. Tender tea shoots recognition and positioning for picking robot using improved YOLO-V3 Model. IEEE Access, 2019; 7: 180998-181011.

[13] Chen Y-T, Chen S-F. Localizing plucking points of tea leaves using deep convolutional neural networks. Computers and Electronics in Agriculture, 2020; 171: 105298. doi: 10.1016/j.compag.220.105298.

[14] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015; 521(7553): 436–444.

[15] Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, et al. Deep learning for generic object detection: A survey. International Journal of Computer Vision, 2020; 128(2): 261–318.

[16] Szegedy C, Wei L, Yangqing J, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA: IEEE, 2015; pp.1–9. doi: 10.1109/CVPR.2015.7298594.

[17] Tan M X, Pang R, Le Q V. EfficientDet: Scalable and efficient object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, USA: IEEE, 2020; pp.10778–10787. doi: 10.1109/CVPR42600.2020.01079.

[18] Liu Z, Li J, Shen Z, Huang G, Yan S, Zhang C. Learning efficient convolutional networks through network slimming. In: 2017 IEEE International Conference on Computer Vision, Venice, Italy: IEEE, 2017; pp.2755–2763. doi: 10.1109/ICCV.2017.298.

[19] Molchanov P, Mallya A, Tyree S, Frosio I, Kautz J. Importance estimation for neural network pruning. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, USA: IEEE, 2019; pp.11256–11264. doi: 10.1109/CVPR.2019.01152.

[20] Han S, Pool J, Tran J, Dally W J. Learning both weights and connections for efficient neural networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada: MIT Press, 2015; pp.1135–1143.

[21] Zhang P Y, Zhong Y X, Li X Q. SlimYOLOv3: Narrower, faster and better for real-time UAV applications. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul: IEEE, 2019; pp.37–45. doi: 10.1109/ICCVW.2019.00011.

[22] Wu D, Lyu S, Jiang M, Song H. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. Computers and Electronics in Agriculture, 2020; 178: 105742. doi: 10.1016/j.compag.2020.105742.

[23] Wang D, He D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning. Biosystems Engineering, 2021; 210: 271–281.

[24] Xu P, Cao J, Shang F, Sun W, Li P. Layer pruning via fusible residual convolutional block for deep neural networks. arXiv. 2020; arXiv: 2011.14356v1. doi: 10.48550/arXiv.2011.14356.

[25] Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv, 2018; arXiv:1804.02767. doi: 10.48550/arXiv.1804.02767.

[26] He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA: IEEE, 2016; pp.770–778. doi: 10.1109/CVPR.2016.90.

[27] He Y, Zhang X, Sun J. Channel pruning for accelerating very deep neural networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy: IEEE, 2017; pp.1398–1406. doi: 10.1109/ICCV.2017.155.