Improved YOLOv5s-based lightweight detection method for tobacco leaves in complex environments

Yihao Liu[1], Bojin Chen[1], Hengzhi Fan[1], Erdeng Ma[2], Delun Li[3], Xin Wang[1*], Du Chen[4*]

(1. *College of Engineering, China Agricultural University, Beijing 100083, China*;
2. *Yunnan Academy of Tobacco Agricultural Sciences, Kunming 650021, China*;
3. *Guizhou Academy of Tobacco Science, Guiyang 550081, China*;
4. *State Key Laboratory of Intelligent Agricultural Power Equipment, Beijing 100083, China*)

Abstract: Complex environments featuring variable lighting and backgrounds similar in color to the target objects present challenges for the rapid and accurate detection of tobacco leaves, which is critical for the development of automated tobacco leaf harvesting robots. This study introduces a depth filtering approach to filter out complex regions based on distance information, thereby simplifying the detection task, and proposes a lightweight detection method based on an enhanced YOLOv5s model. Initially, the YOLOv5s backbone network is substituted with a more lightweight MobileNetV2 to reduce the model size. Subsequently, sparse model training combined with the scaling factor distribution rules of batch normalization layers is utilized to identify and eliminate inconsequential neural network channels. Finally, fine-tuning and knowledge distillation techniques are employed to achieve a model accuracy close to the YOLOv5s baseline. Experimental results indicate that the depth filtering method can improve the model's precision, recall, and mean Average Precision (mAP) by 11.2%, 29.6%, and 17.1%, respectively. The optimized lightweight model achieves a precision of 91.1%, a recall of 90.8%, and an mAP of 91.6%, with a memory footprint of only 1.4MB. It delivers a detection frame rate of 112 fps on desktop computers and 21 fps on mobile devices, which is approximately 3.5 and 4 times faster, respectively, compared to the baseline YOLOv5s tobacco leaf detection model. The precision, recall, and mAP experience a marginal decrease of 3.8, 1.6, and 2.8 percentage points, respectively, while the memory consumption is merely 10% of the pre-optimization amount. In summary, the proposed method enables the accurate detection of tobacco leaves against near-color backgrounds. Simultaneously, it achieves effective lightweighting of the model without compromising its performance, thereby providing technical support for deploying tobacco leaf detection on mobile platforms.

Keywords: harvesting, object detection, YOLOv5s, tobacco leaves, complex environments
DOI: 10.25165/j.ijabe.20251804.9034

Citation: Liu Y H, Chen B J, Fan H Z, Ma E D, Li D L, Wang X, et al. Improved YOLOv5s-based lightweight detection method for tobacco leaves in complex environments. Int J Agric & Biol Eng, 2025; 18(4): 229–238.

1 Introduction

China is a major tobacco producer, contributing to half of the global tobacco yield[1,2]. Tobacco is cultivated across most provinces in China, with leaves maturing from June onwards. Delayed harvesting of mature leaves increases the risk of diseases such as tobacco brown spot disease, which can cause leaf withering and result in economic losses[3-5]. Currently, tobacco harvesting in China relies mainly on manual labor, a process characterized by high labor costs and harsh working conditions. Developing automated tobacco leaf harvesting robots capable of replacing manual labor in this demanding process is essential for stimulating the economic viability of the tobacco industry[6]. Accurate detection of tobacco leaves serves as a prerequisite for subsequent maturity identification and picking point localization, forming a critical foundation for the technical advancement of tobacco leaf harvesting robots. However, as tobacco is cultivated in extensive field environments, the similarity in color between tobacco leaves and stems complicates the extraction of effective features. This complexity contributes to the challenges of tobacco leaf detection, increasing the likelihood of false detections and missed detections[7-9]. While traditional detection algorithms perform well, their adaptability to low-computational platforms is limited, affecting the real-time applicability[10-13]. Researching detection methodologies that perform robustly against similar-color backgrounds and enhancing real-time detection capabilities in field conditions are urgent priorities to advance the technology of tobacco leaf harvesting robots[14,15].

Most existing algorithms for tobacco leaf detection are deployed in phases other than harvesting, such as tobacco phenotypic analysis[16], post-harvest leaf grading[17,18], and the identification and control of diseases and pests[19-22]. Consequently, there is a necessity to develop detection methodologies that are specifically tailored for the harvesting phase. Unlike other growth phases such as phenotypic analysis or disease detection, harvesting-

Received date: 2024-04-28 **Accepted date:** 2025-07-01
Biographies: Yihao Liu, PhD candidate, research interest: agricultural intelligent equipment, Email: liuyihao@cau.edu.cn; **Bojin Chen**, Undergraduate Student, research interest: agricultural image recognition, Email: 2022307150816@cau.edu.cn; **Hengzhi Fan**, Undergraduate Student, research interest: agricultural image recognition, Email: 3178527701@qq.com; **Erdeng Ma**, PhD, Associate Researcher, research interest: soil agrochemicals and tobacco cultivation, Email: erdengma@163.com; **Delun Li**, MS, Agronomist, research interest: integration of agricultural machinery and agronomy, Email: lidelun007@163.com.
***Corresponding author: Xin Wang**, PhD, Associate Professor, research interest: agricultural intelligent equipment. College of Engineering, China Agricultural University, Beijing 100083, China. Tel: +86-13810138307, Email: wangxin117@cau.edu.cn; **Du Chen**, PhD, Professor of Engineering, research interest: agricultural robot technology. State Key Laboratory of Intelligent Agricultural Power Equipment, Beijing 100083, China. Tel: +86-13426060122, Email: tchendu@cau.edu.cn.

stage leaf detection must cope with dense leaf coverage, near-color background interference (e.g., stems and weeds), and real-time operational demands. These differences necessitate dedicated detection methods optimized for both accuracy and computational efficiency. Leaf detection during the harvest period involves identifying targets within similar-color backgrounds, a challenge increasingly addressed by various deep learning algorithms that have been designed for this purpose. Tobacco leaf detection during the harvest period falls within the realm of object detection in similar-color environments. With the advancements in deep learning, numerous algorithms have been adapted for detecting objects under such conditions. Wang et al.[23] developed a tea leaf picking point detection method employing Mask R-CNN, integrated with Resnet50 and RoIAlign technologies, achieving an average detection accuracy of 93.95%. Cardellicchio et al.[24] utilized a YOLOv5-based single-stage detector for identifying phenotypic features of similarly colored tomato plants, attaining substantial average detection accuracy. Ma et al.[25] introduced the YOLOv5-lotus method, effectively detecting mature lotus pods against similar-color backgrounds in a natural environment. The addition of a Coordinate Attention module enhanced the detection precision, facilitating automated lotus pod harvesting. Wang et al.[26] proposed the YOLOv5s-CFL model for real-time detection of millet peppers in similar-color backgrounds, thus aiding the development of millet pepper harvesting robots. Qiu et al.[27] developed a lightweight variant of the enhanced YOLOv5, tailored for detecting Foxtail Millet Ears in densely populated fields, achieving an average precision of 96.60%. Ho et al.[28] employed the Faster R-CNN framework to detect watermelons against similar backgrounds, contributing to watermelon picking methodologies. Li et al.[29] formulated a deep learning-based object detection algorithm using YOLOv4_tiny, targeting green peppers in similar-color environments, thereby supporting green pepper harvesting automation. Although existing studies on object detection in similar-color backgrounds have shown promising results, research specifically targeting leaf detection under such conditions remains limited. Moreover, the proposed models often suffer from large parameter sizes, high algorithmic complexity, and slow detection speeds, making them less suitable for real-time field applications.

To address the challenge of similar-color background interference in tobacco leaf detection and to effectively lightweight the model while maintaining its performance, this study firstly utilizes depth information to filter out complex backgrounds in images. A lightweight tobacco leaf detection algorithm based on YOLOv5s is proposed, which significantly reduces the model size by substituting the YOLOv5s backbone network. Channel pruning is employed to remove non-essential channels from the model, followed by fine-tuning and knowledge distillation to restore the model's accuracy to near pre-pruning levels. This approach provides technical support for the rapid and accurate detection capabilities required by tobacco leaf harvesting robots.

2 Materials and methods

2.1 Data collection and processing

This study utilized the Intel Realsense D405 depth camera for tobacco leaf image collection. Images were captured in August 2023 at the World Tobacco Variety Park located at Zhangjiacun Road, Chengjiang City, Yuxi City, Yunnan Province, China (24°39′N, 102°52′E). The tobacco variety used for the study was Yunyan 87. The collected color images have a resolution of 1280×720 pixels and are stored in the PNG format, and depth images are saved in the NPY format. The imaging conditions encompassed diverse environments, including sunny and cloudy weather, different times of day such as morning, noon, and evening, as well as various lighting angles, including direct and backlighting. The capture process mimicked the robot picking routine, maintaining a consistent distance while acquiring images from various overhead angles, as illustrated in Figure 1. The dataset encompasses a wide range of maturity levels, postures, lighting conditions, and backgrounds, resulting in a total of 1020 tobacco leaf images. Figure 2 displays a selection of the captured images.

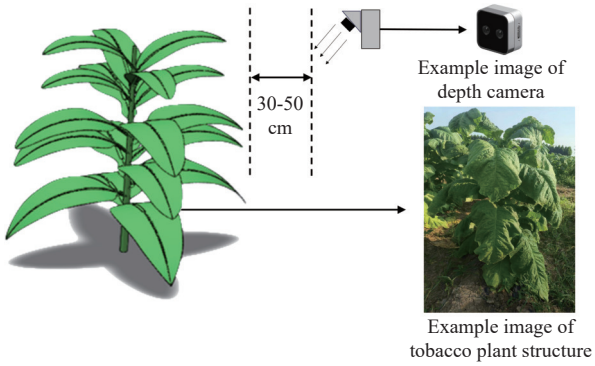


Figure 1 Schematic diagram of the acquisition process of images under different conditions

a. Morning b. Noon c. Nightfall

d. Back light e. Forward light f. Cloudy day

Figure 2 Example of tobacco leaf images

In this study, a camera-off-hand setup was adopted, where the depth camera is mounted in a fixed position on the harvesting platform to ensure stable imaging. This method is also potentially transferable to camera-in-hand configurations, as image acquisition typically occurs before manipulator movement begins.

The presence of similar-color backgrounds in tobacco leaves, such as weeds and incomplete leaves, tends to induce false detections and missed detections in the detection model. Furthermore, tobacco leaf harvesting robots require the identification of the nearest tobacco leaf targets during operation. To address this, the study utilizes depth information for background filtration. Specifically, in the RGB images, pixels with depth values greater than 60 cm are reassigned an RGB value of (128, 128, 128), effectively mitigating the impact of similar-color backgrounds on tobacco leaf detection. The images before and after processing, as well as the colored depth image, are shown in Figure 3.

a. Before depth filtering b. After depth filtering c. Depth image (colored)

Figure 3 Comparison before and after depth filtering

The harvesting environment for tobacco leaves differs significantly from typical greenhouse settings, as depicted in Figures 4a and 4b for tomato and pepper harvesting, where the fruits are readily distinguishable from their background. In contrast, the dense growth and large leaves of tobacco, alongside near-color backgrounds such as weeds and partial leaves, increase the risk of both false and missed detections in detection models, as illustrated in Figure 4c. Moreover, tobacco leaf harvesting robots are required to identify the nearest leaf targets accurately.

In the process of constructing the tobacco leaf dataset, images are firstly preprocessed through depth filtering, and then Labelimg software is used to mark the area of the target tobacco leaf in the image with a rectangular frame to generate a dataset that can be used for YOLOv5s training. In order to ensure the generalization performance and robustness of the model, the data enhancement used included randomly adding noise, cropping, rotation, and other operations, expanding it to 2380 images, and creating a dataset in VOC format. Among them, the category label of tobacco leaves is set to tobacco, and the labeled rectangle fits the outline of the tobacco leaves. Then it was divided into a training set (1666 pictures), a verification set (476 pictures), and a test set (238 pictures) according to the ratio of 7:2:1.

a. Tomato harvest environment b. Pepper harvesting environment c. Tobacco harvesting environment

Figure 4 Comparison of the growing environments of different crops

2.2 Algorithmic improvements

YOLO is a single-stage object detection network, conceptualizing detection as a regression problem. This end-to-end network significantly enhances detection speed. The YOLO series has evolved from YOLOv1, with YOLOv5 being one of the widely adopted versions. YOLOv1, while foundational, suffers from poor detection performance and inaccurate localization[30]. YOLOv2 improved both performance and speed, yet struggled in a complex environment[31]. YOLOv3 marked advancements in speed and accuracy over its predecessors, but underperformed in scenarios with significant occlusion[32]. YOLOv4 enhanced feature extraction diversity and robustness through network modifications, yet faced issues with inaccurate bounding boxes and low recall rates, alongside increased algorithmic complexity[33]. Subsequent versions post YOLOv5 have seen improvements in accuracy, but with more complex implementations and slower detection speeds compared to YOLOv5[34]. YOLOv5 stands out for its optimized architecture, offering rapid and accurate object detection. Hence, this study employs YOLOv5s, recognized for its superior overall performance within the YOLOv5 variants.

The YOLOv5s architecture comprises an input layer, a backbone feature extraction network, a neck network, and a prediction head. Within the fully trained YOLOv5s object detection model, the backbone layer accounts for the majority of computational resources and parameters. To effectively reduce the model's memory footprint and enhance field detection speed, it is essential to modify the backbone. Replacing the backbone with a lightweight network serves as an efficient strategy to decrease the number of parameters and computational load while maintaining network performance. Further model lightening is achieved through channel pruning, followed by the restoration of detection performance via knowledge distillation. Such adaptations are pivotal in optimizing the model's structure and enhancing its deployment efficiency.

2.3 MobileNetV2

MobileNetV2, a lightweight neural network architecture, achieves low computational complexity and reduced parameter size while delivering exceptional performance[35]. As a part of the MobileNet series, it is tailored for efficient image processing in resource-constrained environments, such as mobile devices and

embedded systems. Consequently, MobileNetV2 is utilized in this study to modify the backbone of YOLOv5s for optimized performance.

In the MobileNet series, depthwise separable convolution is introduced, segmenting the convolution process into depthwise and point-wise convolutions to decrease computational demand. Unlike standard convolutions where each kernel processes all channels of the input feature map simultaneously (Figure 5), depthwise separable convolution efficiently reduces the complexity. Assuming the input image has a dimension of N, with a convolution kernel size of L_k, and output dimensions of $L_w \times L_k \times M$, the total parameter count $X_{\text{parameter}}$ and computational load $X_{\text{calculation}}$ can be represented as follows:

$$X_{\text{parameter}} = L_k \times L_k \times N \times M \tag{1}$$

$$X_{\text{calculation}} = L_k \times L_k \times N \times M \times L_w \times L_k \tag{2}$$

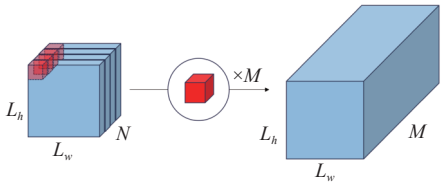


Figure 5 Standard convolutions

Depthwise separable convolution comprises two stages: depthwise and pointwise convolutions. Initially, each input channel undergoes an independent convolution to create feature maps across multiple channels, known as depthwise convolution. Subsequently, 1×1 convolutions are applied to linearly combine these feature maps, resulting in the final output feature map, termed pointwise convolution. This process is illustrated in Figure 6. The total number of parameters and computational load in depthwise separable convolution can be quantified as:

$$X_{\text{parameter}} = L_k \times L_k \times N + N \times M \tag{3}$$

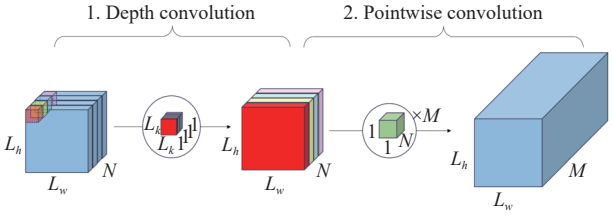


Figure 6 Depthwise separable convolution

$$X_{\text{calculation}} = L_k \times L_k \times N \times L_w \times L_k + N \times M \times L_w \times L_k \tag{4}$$

Consequently, the number of parameters and computational effort required for depthwise separable convolution is only $\left(\frac{1}{M} + \frac{1}{L_k^2}\right)$ of that for standard convolution. This results in a significant reduction in parameter and computational complexity for depthwise separable convolution, enabling a substantial increase in the model's execution speed.

The essence of the MobileNetV2 architecture lies in its series of inverted residual modules, each composed of three layers: an expansion layer, a depthwise separable convolution layer, and a projection layer. The expansion layer employs 1×1 convolutions to upsample low-dimensional inputs to a higher dimension, followed by the depthwise separable convolution layer which performs spatial filtering on these high-dimensional features. Subsequently, the projection layer utilizes 1×1 convolutions to project the features back to a lower dimension.

Incorporating MobileNetV2 as the backbone network for YOLOv5s leverages its efficient feature extraction and compact model size. MobileNetV2's layered architecture efficiently captures a wide range of image features, conveying them to other components of YOLOv5s for object detection. The integration of MobileNetV2 into YOLOv5s enhances the real-time performance on embedded devices, simultaneously reducing the number of the model's parameters and computational resources while ensuring high accuracy. The network's architecture is depicted in Figure 7.

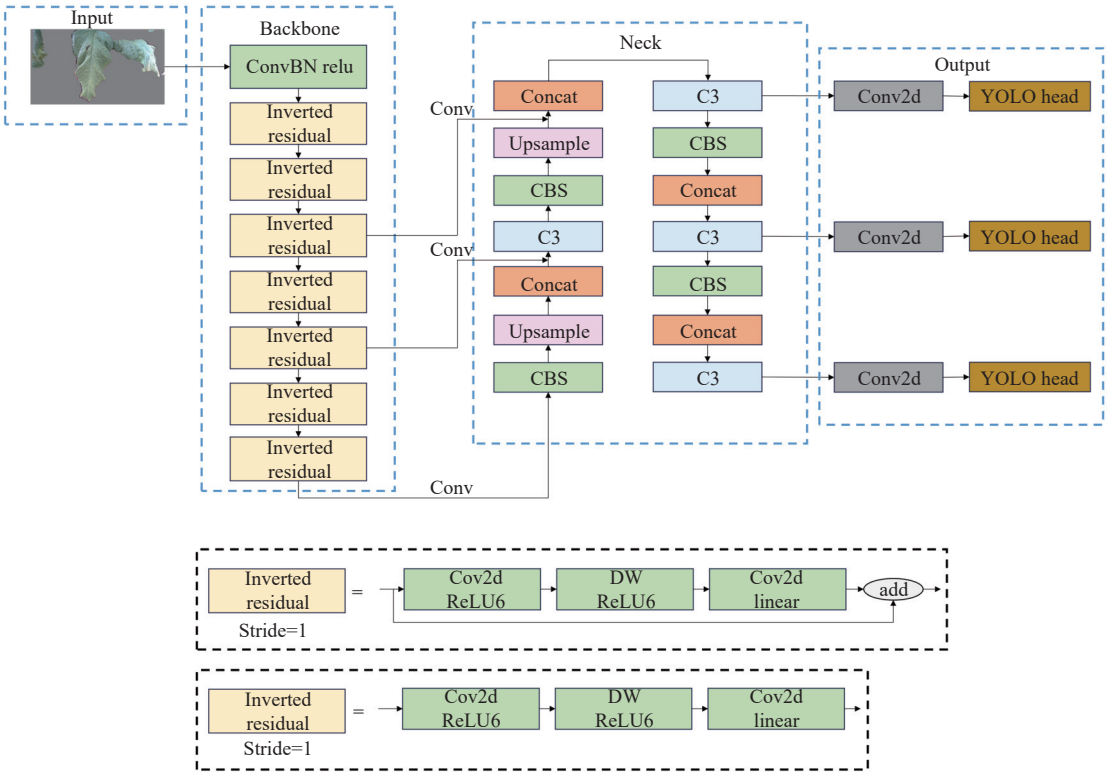


Figure 7 Structure schematic of YOLOv5s-MobileNetV2 algorithm

2.4 Channel pruning and knowledge distillation

Although the YOLOv5s model with a modified backbone can accurately detect tobacco leaves, its size remains slightly large. To further reduce the complexity and enhance the efficiency of the detection model, channel pruning algorithms are employed to further optimize the YOLOv5s tobacco leaf detection model with the altered backbone[36].

2.4.1 Sparse training

In the method of channel pruning, it is a prerequisite to conduct sparse training on the batch normalization (BN) layers within the network model. In models trained without sparsity constraints, the scaling factors γ of the batch normalization (BN) layers tend to follow a normal distribution centered around 1 as training progresses. During sparse training, by incorporating an L1 regularization constraint on the BN layer's scaling factors into the loss function, the parameters are sparsified, driving the scaling factors' distribution closer to zero. The loss function's computation process is formulated as follows:

$$L=\sum_{(x,y)} l(f(x,W),y)+\lambda\sum_{\gamma\in\Gamma} g(\gamma) \tag{5}$$

In the equation, (x,y) denotes the training inputs and targets, while W represents the weights used in standard training. The first summation term signifies the loss function for standard training. The second summation term pertains to the sparsification of the BN-layer scaling factors (denoted as γ) through L1 regularization. Here, $g(\gamma)=|\gamma|$ serves as a penalty term for sparse training, and λ is the balancing factor between standard and sparse training, also known as the sparsity regularization coefficient.

During the sparse training process, balancing model accuracy with the degree of sparsity is crucial, and this balance hinges on the sparsity regularization coefficient λ. A larger value of λ accelerates the approach of the BN layer's scaling factors towards zero, potentially reducing the model's average recognition accuracy. Conversely, a smaller λ value results in a slower rate of approach to zero for the scaling factors, leading to more stable convergence of the model's average recognition accuracy. Thus, careful selection of the sparsity regularization coefficient is essential in designing an effective sparsification strategy to optimize the performance balance.

2.4.2 Channel pruning and model fine-tuning

Post sparse training, model pruning is executed by eliminating less significant channels within the convolutional layers. This pruning is informed by the distribution of the scaling factors in the BN layer. The factors are ordered by mean value, and channels approaching zero are removed. Deep networks primarily involve multiplications and additions across layers, and channels with near-zero scaling factors contribute minimally to the model. Thus, their removal aids in network compression, as illustrated in Figure 8. Models subjected to pruning typically experience some accuracy degradation. To mitigate this loss and adjust to the altered network architecture, fine-tuning is essential. This step entails re-loading the dataset and associated configuration files to either restore or enhance the network's performance.

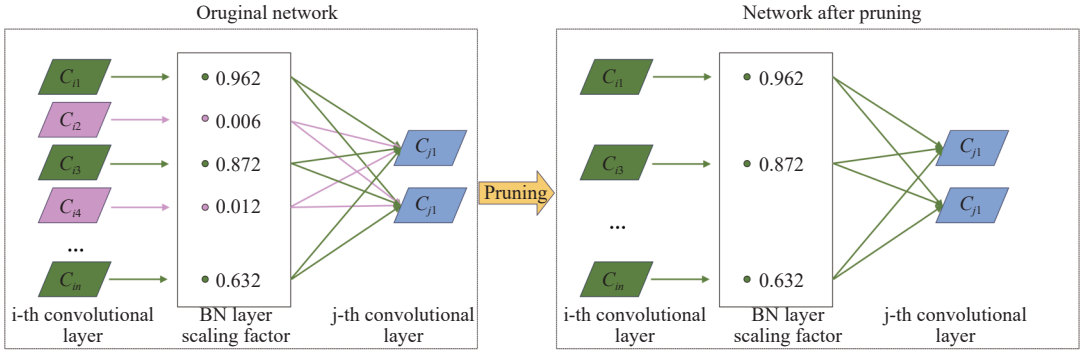


Figure 8 Channel pruning

2.4.3 Knowledge distillation

Knowledge distillation is a technique for model compression, fundamentally involving the transfer of knowledge from a large, complex 'teacher model' to a smaller, simpler 'student model'[37]. This approach enables smaller models to achieve performance comparable to larger models while requiring fewer computational resources. In the knowledge distillation process, the student model learns not only the correct outputs for actual labels but also emulates the outputs of the teacher model, as depicted in Figure 9.

In this study, the tobacco leaf detection model with a replaced backbone is selected as the teacher network for the knowledge distillation experiment, while the fine-tuned pruned model serves as the student network. Through knowledge distillation, the soft feature information from the intermediate layers learned by the teacher network is imparted to the student network, thereby enhancing its recognition accuracy.

2.5 Test platform

The computational setup used for the experiments consisted of a computer running the Windows 10 operating system, equipped with an Intel Core I5-13490F CPU, Nvidia Geforce RTX 3080 Ti GPU, and 32GB of RAM. The software environment included Pytorch 1.10.0, Cuda 11.3, cuDNN 8.2.0, and Python 3.8.

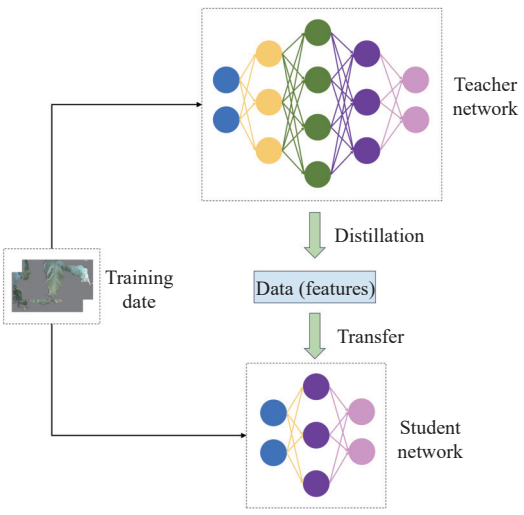


Figure 9 Knowledge distillation

2.6 Evaluation indicators

To holistically assess the impact of lightweight model processing and data modification on detection performance, three model recognition metrics, two computational performance metrics, and model memory usage were chosen for evaluation. The recognition metrics include precision (P), recall (R), and mean average precision (mAP) at an IoU threshold of 0.5. The computational metrics comprise the number of parameters and the volume of floating-point operations. Precision, recall, and mAP are mathematically expressed as follows:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{6}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

$$\mathrm{mAP@0.5} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{AP}_i \tag{8}$$

In these definitions, TP (True Positive) denotes the count of positive samples correctly identified as positive; FP (False Positive) refers to the count of negative samples incorrectly identified as positive; FN (False Negative) represents the count of negative samples incorrectly identified as negative; N signifies the total number of categories; and AP_i denotes the average precision for the ith category at an IoU threshold of 0.5.

Post knowledge distillation, to further examine the effect of lightweight processing on the model, the frame rates detected on the same desktop configuration and on a mobile platform (NVIDIA Jetson Orin NX 16 G) are also included as computational performance metrics.

3 Results and discussion

3.1 The impact of depth filtering on detection results

To validate the effectiveness of depth filtering in simplifying the complex background of tobacco leaf images for enhanced detection, two datasets were utilized: one comprising original tobacco leaf images and the other consisting of images post depth filtering. Both datasets were trained using the YOLOv5s model. The mAP variation during training for each dataset is depicted in Figure 10, with the training spanning 100 epochs. The results of the training are presented in Table 1.

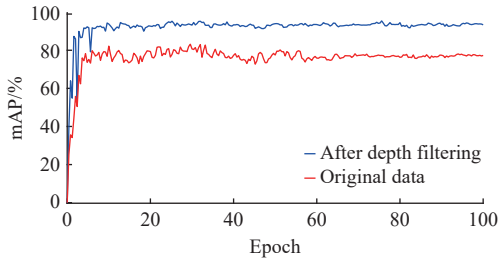


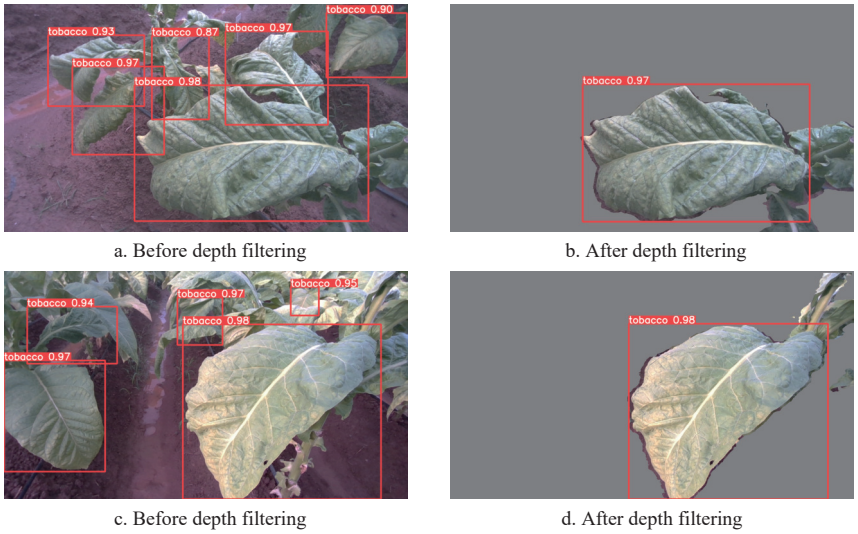
Figure 10 mAP change curves of different datasets

Table 1 Training results on different datasets

Training data name	P/%	R/%	mAP/%
Original data	83.7	62.8	77.3
Depth-filtered data	94.9	92.4	94.4

As indicated by the figure and table, training with depth-filtered tobacco leaf images resulted in an average precision of 94.4%. In contrast, the original dataset achieved only 77.3% average precision after training, primarily due to background interference from distant tobacco leaves, which negatively affected the detection performance. Depth filtering of the images effectively filters out the complex background, thereby enhancing the detection of complete tobacco leaves. A comparison of the detection results before and after depth filtering is illustrated in Figure 11.

As indicated in Table 1, training with depth-filtered tobacco leaf images results in a model with higher accuracy, recall rate, and mAP by 11.2, 29.6, and 17.1 percentage points, respectively, compared to using original images. Thus, employing depth filtering to remove the complex background from the original images proves more effective for tobacco leaf detection.



a. Before depth filtering b. After depth filtering

c. Before depth filtering d. After depth filtering

Figure 11 Detection results before and after depth filtering

3.2 Effects of model lightweighting

3.2.1 Comparing models with and without backbone replacement

Utilizing depth information to filter out the complex background of tobacco leaves has simplified the task of tobacco leaf detection. To effectively reduce the memory footprint of the YOLOv5s model and accelerate its detection speed in field conditions, the backbone of the model was replaced with MobileNetV2. The performance of the model post replacement is presented in Table 2.

As listed in Table 2, the precision, recall, and mAP of the

YOLOv5s-MobileNetV2 model are 2.6, 0.9, and 2.3 percentage points lower, respectively, than those of the YOLOv5s model. However, the size of the YOLOv5s-MobileNetV2 model is only 23.5% of the YOLOv5s, representing a significant step towards the lightweighting of the detection model.

Table 2 Performance of the model before and after backbone replacement

Model name	*P*/%	*R*/%	mAP/%	Number of parameters	GFLOPS	Model size/MB
YOLOv5s	94.9	92.4	94.4	7.2×10^6	16.2	13.6
YOLOv5s-MobileNetV2	92.3	91.5	92.1	1.4×10^6	2.5	3.2

3.2.2 Effects of model pruning

To avoid confusion, it is noted that λ represents the sparsity regularization coefficient controlling the strength of L1 penalty, while γ denotes the scaling factors in BN layers used for pruning decisions.

After the backbone replacement, channel pruning was conducted on the YOLOv5s-MobileNetV2 model. To maintain robust detection performance while facilitating channel pruning, comparative experiments were performed using different sparse regularization coefficients λ. Variations in λ resulted in corresponding changes in the weights and mAP of the model's BN layer. These changes were visualized using the TensorBoard module in the PyTorch framework. The distribution changes of the BN layer's scaling factors (γ) for different λ values are illustrated in Figure 12.

Figure 12a depicts the BN layer's weight changes with λ set to 0.0003 after 100 epochs of training. After 100 epochs of sparse training, the distribution of BN-layer scaling factors (γ) remained largely unchanged, with none approaching zero, suggesting an overly small λ value of 0.0003. Figure 12c, with λ set to 0.05, shows significant screening of channels with BN layer scaling factors nearing zero within the initial 10 epochs of sparse training, potentially impacting model accuracy, indicating an excessively high λ value of 0.05. As shown in Figure 12b, setting λ at 0.004 and after 100 epochs of sparse training, the variation in scaling factors nearing zero in the BN layer stabilized, indicating that the model is adequately prepared for the pruning experiment.

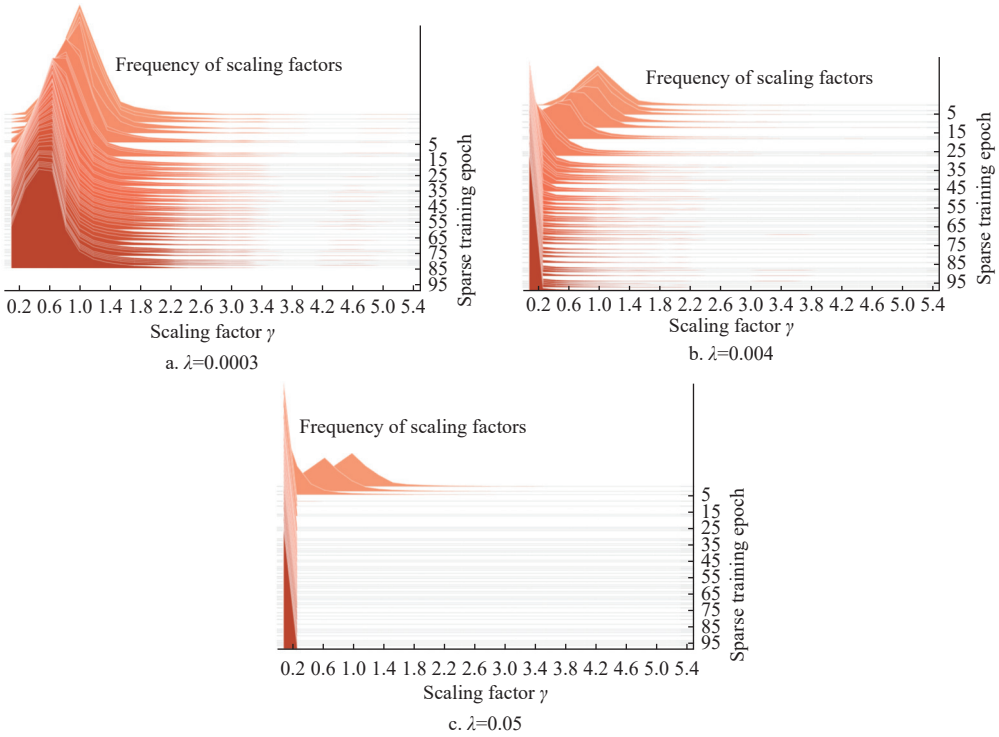


Figure 12 Distribution of BN layer scaling factors after different sparse trainings

Figure 13 illustrates the variations in average accuracy during sparse training at sparse regularization coefficients of 0.05 and 0.004. It is evident from the figure that the model accuracy with a coefficient of 0.004 significantly surpasses that with 0.05, resulting in a difference of 10.7 percentage points in final average accuracy.

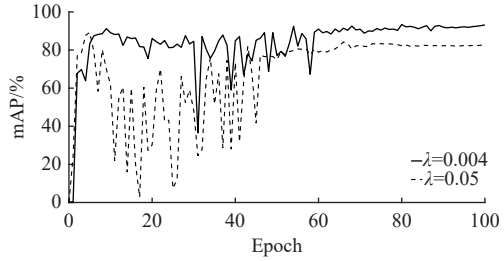


Figure 13 mAP change curves under different λ

The experimental results indicate that with the sparse regularization coefficient λ set at 0.004, the optimal pruning model was identified by iteratively adjusting the pruning rate. This model, preserving 25% of the original model's channels, was selected as the final configuration. The variation in the number of channels in the model before and after pruning is illustrated in Figure 14.

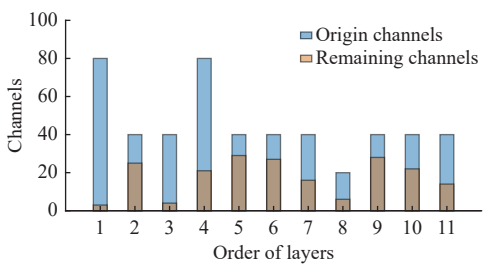


Figure 14 Changes of partial channels in the model

Following the pruning and subsequent fine-tuning training, the model, now designated as YOLOv5s-MobileNetV2-Pruned, exhibited a slight reduction in accuracy compared to its pre-pruning state. The detailed performance metrics of the model before and after pruning are depicted in Figure 15. Although the average precision decreased by 3.2 percentage points post pruning, the model's memory footprint experienced a significant reduction of 56 percentage points.

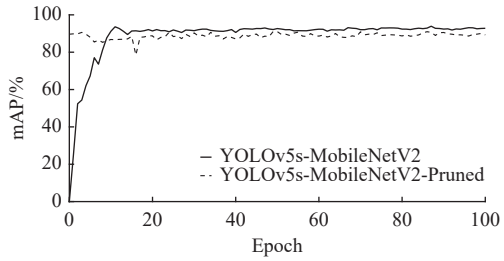


Figure 15 Comparison of the model's mAP before and after pruning

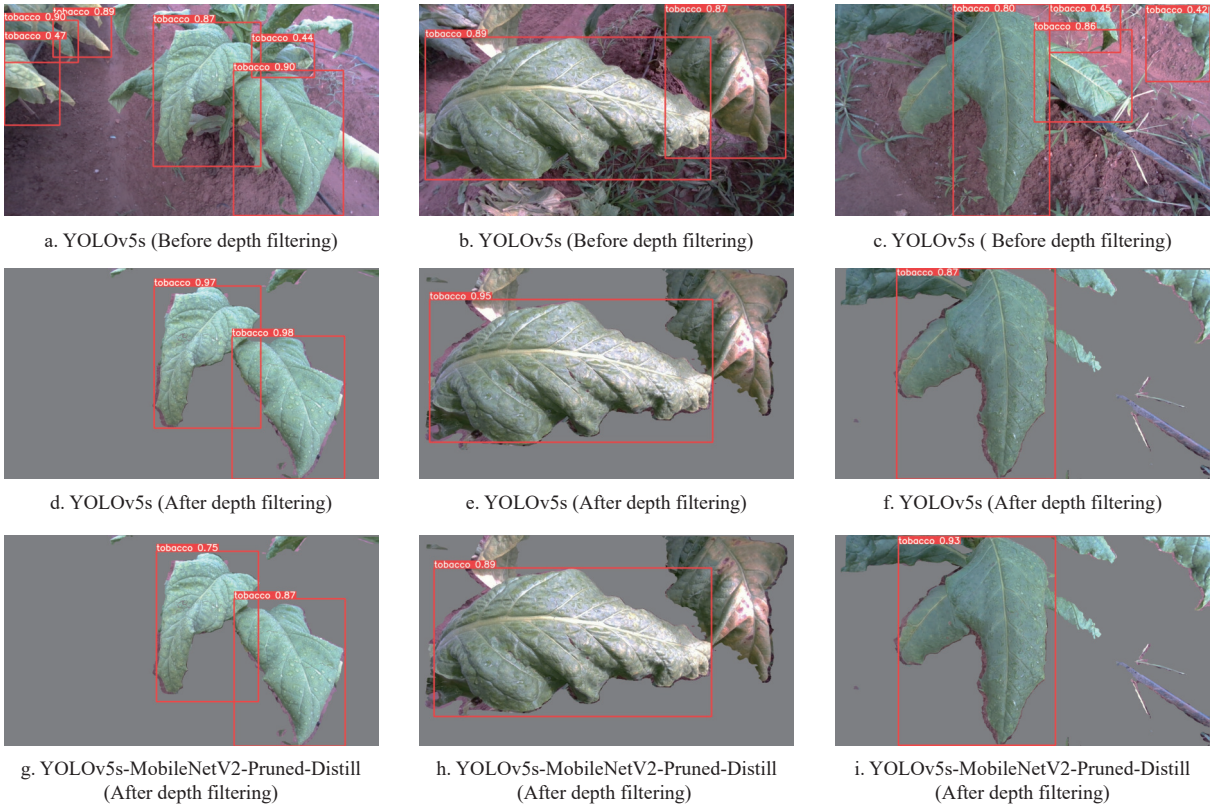
3.2.3 Effects of knowledge distillation

The model post knowledge distillation has been designated as YOLOv5s-MobileNetV2-Pruned-Distill. Various models were deployed on a mobile platform for comparative testing, with the results presented in Table 3.

Table 3 reveals that following knowledge distillation, the model demonstrated improvements in precision, recall, and mAP by 3.2, 4.1, and 2.7 percentage points, respectively, while maintaining consistent memory usage and frame rate. When compared to YOLOv5s-MobileNetV2, YOLOv5s-MobileNetV2-Pruned-Distill showed slight reductions of 1.2, 0.7, and 0.5 percentage points in precision, recall, and mAP, respectively, along with a decrease in memory usage by 1.8 MB and an increase of 34 in desktop frame rate. Overall, the memory footprint of the optimized YOLOv5s model for leaf disease detection was significantly reduced by 90%, with desktop and mobile detection frame rates increasing nearly 3.5 and 4 times, respectively. To validate the effectiveness of the lightweight model, tobacco leaf images from the dataset were tested, with results presented in Figure 16.

Table 3 Performance of models

Model name	*P*/%	*R*/%	mAP/%	Model size/MB	Desktop frame rate	Mobile frame rate
YOLOv5s	94.9	92.4	94.4	13.6	32	5
YOLOv5s-MobileNetV2	92.3	91.5	92.1	3.2	78	14
YOLOv5s-MobileNetV2-Pruned	87.9	86.7	88.9	1.4	112	21
YOLOv5s-MobileNetV2-Pruned-Distill	91.1	90.8	91.6	1.4	112	21



a. YOLOv5s (Before depth filtering) b. YOLOv5s (Before depth filtering) c. YOLOv5s (Before depth filtering)

d. YOLOv5s (After depth filtering) e. YOLOv5s (After depth filtering) f. YOLOv5s (After depth filtering)

g. YOLOv5s-MobileNetV2-Pruned-Distill (After depth filtering) h. YOLOv5s-MobileNetV2-Pruned-Distill (After depth filtering) i. YOLOv5s-MobileNetV2-Pruned-Distill (After depth filtering)

Figure 16 Result of model validation

Figure 16 illustrates that depth information filtering effectively simplifies complex backgrounds, thereby reducing the complexity of detection tasks. In tobacco leaf detection, both the original and the lightweight models successfully identified all tobacco leaves. The marginally reduced confidence level in the lightweight model suggests that its detection performance is closely aligned with the original model.

4 Discussion

Tobacco leaf detection is challenging due to the color similarity between leaves and stems, leading to potential both false and missed detections that could impact the efficiency of harvesting equipment. To address this, a depth filtering method was introduced to enhance the model's detection accuracy by simplifying the complex background, improving from 77.3% to 94.4%. However, inappropriate depth filtering thresholds might render leaf information incomplete, as demonstrated with the upper right tobacco leaves in Figure 16b, which are fully detected in the original images, but are missed after depth filtering due to the loss of complete leaf information. Future adjustments to the depth

filtering threshold could allow for tailored background filtration based on specific conditions.

To tackle the large size of the YOLOv5s model and the limited computational capacity of harvesting equipment, a lightweighting strategy was developed. This involved compressing the model by replacing its backbone network and further reducing its size through channel pruning, with knowledge distillation applied to closely restore its pre-pruning accuracy. Experimental results indicate that this approach reduced the model size from 13.6 MB to 1.4 MB while minimally impacting accuracy by only 2.8%, ensuring precise tobacco leaf detection with lower computational demands. Dense growth and occasional leaf occlusion present additional challenges; slight occlusions allow for the differentiation of leaves as depicted in Figures 16a, 16d, and 16g, while severe occlusions, shown in Figures 16c, 16f, and 16i, hinder separate leaf detection. Incorporating an attention mechanism algorithm into the model is considered for future work to address severe occlusions. In addition to occlusion, variable illumination in field environments—such as shadows, strong sunlight, and backlighting—can lead to inconsistent color features and reduced detection accuracy. To mitigate this, future work will explore illumination-invariant preprocessing techniques such as adaptive gamma correction, histogram equalization, or high dynamic range (HDR) enhancement, in order to improve model robustness under diverse lighting conditions.

Unlike other fruit harvesting scenarios, tobacco leaf harvesting targets the leaves themselves, where inter-leaf interference is common, as illustrated in Figure 3. The depth filtering method and YOLOv5s-MobileNetV2-Pruned-Distill lightweight model proposed herein effectively detect leaf targets, offering valuable insights for future research on leaf detection in near-color backgrounds, including mulberry and mint leaves. Furthermore, tobacco leaf detection is crucial for identifying maturity levels and detecting pests and diseases, with future studies aimed at enhancing leaf maturity identification.

5 Conclusions

To address the challenges of low accuracy, limited real-time performance, and poor robustness in tobacco leaf detection under complex environments, this paper introduces a depth filtering method and a lightweight tobacco leaf detection approach based on an enhanced YOLOv5s model. The MobileNetV2 lightweight network is employed as the backbone, and channel pruning is utilized to eliminate unimportant channels, thereby reducing the model size. Knowledge distillation is then applied to restore the pruned model to its pre-pruning state. Based on the comparison of detection performance before and after depth filtering, as well as algorithm performance before and after lightweighting, the following conclusions were drawn:

1) Training with images post depth filtering, compared to original images, shows significant improvement. Depth filtering raises the model's precision, recall, and mAP by 11.2, 29.6, and 17.1 percentage points, respectively.

2) Replacing the detection network's backbone and applying channel pruning significantly reduces the model's memory usage and computational demand. The memory usage of the YOLOv5s-MobileNetV2-Pruned-Distill model is 90% of the YOLOv5s, and the frame rates on desktop and mobile platforms increase to 3.5 and 4 times their original values, respectively.

3) Combining knowledge distillation with backbone replacement and channel pruning effectively lightweights the model while minimizing the loss in accuracy. The YOLOv5s-MobileNetV2-Pruned-Distill model, post knowledge distillation, achieves precision, recall, and mAP of 91.1%, 90.8%, and 91.6%, respectively, fulfilling the requirements for tobacco leaf detection.

Acknowledgements

This work was supported by the Key Research and Development Program of China National Tobacco Corporation, titled "Key Technologies Research and Development for Intelligent Tobacco Leaf Harvesting" (Grant No. 110202301016). The authors also acknowledge the Beijing Jingwa Agricultural Science & Technology Innovation Center for providing a supportive environment and facilitating academic exchange throughout the course of this research.

[References]

[1] Tang Z X, Chen L L, Chen Z B, Fu Y L, Sun X L, Wang B B, et al. Climatic factors determine the yield and quality of Honghe flue-cured tobacco. Scientific Reports, 2020; 10: 19868.

[2] Shen X P, Zhang Y H, Tang Y M, Qin Y F, Liu N, Yi Z L. A study on the impact of digital tobacco logistics on tobacco supply chain performance: Taking the tobacco industry in Guangxi as an example. Industrial Management & Data Systems, 2022; 122(6): 1416–1452.

[3] Martins-da-Silva A S, Torales J, Becker R F V, Moura H F, Waisman Campos M, Fidalgo T M, et al. Tobacco growing and tobacco use. International Review of Psychiatry, 2022; 34(1): 51–58.

[4] Xiang L G, Wang H C, Wang F, Cai L T, Li W H, Hsiang T, et al. Analysis of phyllosphere microorganisms and potential pathogens of tobacco leaves. Frontiers in Microbiology, 2022; 13: 843389.

[5] Xiao B G, Zhu J, Lu X P, Bai Y F, Li Y P. Analysis on genetic contribution of agronomic traits to total sugar in flue-cured tobacco (*Nicotiana tabacum* L.). Field Crops Research, 2007; 102(2): 98–103.

[6] Thakur A, Venu S, Gurusamy M. An extensive review on agricultural robots with a focus on their perception systems. Computers and Electronics in Agriculture, 2023; 212: 108146.

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

[8] Zaidi S S A, Ansari M S, Aslam A, Kanwal N, Asghar M, Lee B. A survey of modern deep learning based object detection models. Digital Signal Processing, 2022; 126: 103514.

[9] Tang Y C, Qiu J J, Zhang Y Q, Wu D X, Cao Y H, Zhao K X, et al. Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: A review. Precision Agriculture, 2023; 24: 1183–1219.

[10] Wu D H, Lv S C, Jiang M, Song H B. 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.

[11] 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.

[12] Zhao S L, Zhang S, Lu J M, Wang H, Feng Y, Shi C, et al. A lightweight dead fish detection method based on deformable convolution and YOLOV4. Computers and Electronics in Agriculture, 2022; 198: 107098.

[13] Cao S, Zhao D, Liu X Y, Sun Y P. Real-time robust detector for underwater live crabs based on deep learning. Computers and Electronics in Agriculture, 2020; 172: 105339.

[14] Tang Y C, Chen M Y, Wang C L, Luo L F, Li J H, Lian G P, et al. Recognition and localization methods for vision-based fruit picking robots: A review. Frontiers in Plant Science, 2020; 11: 510.

[15] Xu Z B, Huang X P, Huang Y, Sun H B, Wan F X. A real-time zanthoxylum target detection method for an intelligent picking robot under a complex background, based on an improved YOLOv5s architecture. Sensors, 2022; 22(2): 682.

[16] Zhang W Z, Wang Y F, Shen G C, Li C L, Li M, Guo Y C. Tobacco leaf segmentation based on improved MASK RCNN algorithm and SAM model. IEEE Access, 2023; 11: 103102–103114.

[17] Li G C, Zhen H J, Hao T M, Jiao F Y, Wang D J, Ni K P. Tobacco leaf and

tobacco stem identification location detection method based on YOLOV3 network. In: 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China: IEEE, 2021; 29–31. doi: 10.1109/ICAICA52286.2021.9497964

[18] Harjoko A, Prahara A, Supardi T W, Candradewi I, Pulungan R, Hartati S. Image processing approach for grading tobacco leaf based on color and quality. International Journal on Smart Sensing and Intelligent Systems, 2019; 12(1): 1–10.

[19] Zhu H Y, Cen H Y, Zhang C, He Y. Early detection and classification of tobacco leaves inoculated with tobacco mosaic virus based on hyperspectral imaging technique. In: 2016 ASABE Annual International Meeting, Michigan: American Society of Agricultural and Biological Engineers, 2016; 162460422. doi: 10.13031/aim.20162460422

[20] Lin H, Tse R, Tang S K, Qiang Z P, Ou J L, Pau G. Tobacco plant disease dataset. In: Fourteenth International Conference on Digital Image Processing (ICDIP 2022). Wuhan, China: SPIE, 2022. doi: 10.1117/12.2644288

[21] Zhu H Y, Chu B Q, Zhang C, Liu F, Jiang L J, He Y. Hyperspectral imaging for presymptomatic detection of tobacco disease with successive projections algorithm and machine-learning classifiers. Scientific Reports, 2017; 7(1): 4125.

[22] Tufail M, Iqbal J, Tiwana M I, Alam M S, Khan Z A, Khan M T. Identification of tobacco crop based on machine learning for a precision agricultural sprayer. IEEE access, 2021; 9: 23814–23825.

[23] Wang T, Zhang K M, Zhang W, Wang R Q, Wan S M, Rao Y, et al. Tea picking point detection and location based on Mask-RCNN. Information Processing in Agriculture, 2023; 10(2): 267–275.

[24] Cardellicchio A, Solimani F, Dimauro G, Petrozza A, Summerer S, Cellini F, et al. Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors. Computers and Electronics in Agriculture, 2023; 207: 107757.

[25] Ma J, Lu A, Chen C, Ma X D, Ma Q C. YOLOv5-lotus an efficient object detection method for lotus seedpod in a natural environment. Computers and Electronics in Agriculture, 2023; 206: 107635.

[26] Wang F, Sun Z X, Chen Y, Zheng H, Jiang J. Xiaomila green pepper target detection method under complex environment based on improved YOLOv5s. Agronomy, 2022; 12(6): 1477.

[27] Qiu S J, Li Y, Zhao H M, Li X B, Yuan X Y. Foxtail millet ear detection method based on attention mechanism and improved YOLOv5. Sensors, 2022; 22(21): 8206.

[28] Ho M J, Lin Y C, Hsu H C, Sun T Y. An Efficient recognition method for watermelon using faster R-CNN with post-processing. In: 2019 8th International Conference on Innovation, Communication and Engineering (ICICE), Zhengzhou, China: IEEE, 2019; pp.86–89. doi: 10.1109/ICICE49024.2019.9117374

[29] Li X, Pan J D, Xie F P, Zeng J P, Li Q, Huang X J, et al. Fast and accurate green pepper detection in complex backgrounds via an improved Yolov4-tiny model. Computers and Electronics in Agriculture, 2021; 191: 106503.

[30] Liu W Y, Ren G F, Yu R S, Guo S, Zhu J, Zhang L K, et al. Image-adaptive YOLO for object detection in adverse weather conditions. Proceedings of the AAAI Conference on Artificial Intelligence. 2022; 36(2): 1792–1800. doi: 10.1609/aaai.v36i2.20072

[31] Jiang F C, Zhang H Y, Feng C W, Chen Z. A closed-loop detection algorithm for indoor simultaneous localization and mapping based on you only look once v3. Traitement du Signal, 2022; 39(1): 109–117.

[32] Rahim U F, Mineno H. Data augmentation method for strawberry flower detection in non-structured environment using convolutional object detection networks. Journal of Agricultural and Crop Research, 2020; 8(11): 260–271.

[33] Liu Q L, Ye H X, Wang S M, Xu Z. YOLOv8-CB: Dense pedestrian detection algorithm based on in-vehicle camera. Electronics, 2024; 13(1): 236.

[34] Du H W, Zhu W Z, Peng K, Li W F. Improved high speed flame detection method based on YOLOv7. Open Journal of Applied Sciences, 2022; 12: 2004–2018.

[35] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA: IEEE, 2018; pp.4510–4520. doi: 10.1109/CVPR.2018.00474

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

[37] Gou J P, Yu B S, Maybank S J, Tao D C. Knowledge distillation: A survey. International Journal of Computer Vision, 2021; 129: 1789–1819.