Dynamic detection method for falling ears of maize harvester based on improved YOLO-V4

Ang Gao[1], Aijun Geng[1,2,3*], Zhilong Zhang[1], Ji Zhang[1], Xiaolong Hu[1], Ke Li[1]

(1. *College of Mechanical and Electrical Engineering, Shandong Agricultural University, Tai'an 271018, Shandong, China*;
2. *Shandong Provincial Engineering Laboratory of Agricultural Equipment Intelligence, Tai'an 271018, Shandong, China*;
3. *Shandong Provincial Key Laboratory of Horticultural Machineries and Equipment, Tai'an 271018, Shandong, China*)

Abstract: Traditional maize ear harvesters mainly rely on manual identification of fallen maize ears, which cannot realize real-time detection of ear falling. The improved You Only Look Once-V4 (YOLO-V4) algorithm was combined with the channel pruning algorithm to detect the dropped ears of maize harvesters. *K*-means clustering algorithm was used to obtain a prior box matching the size of the dropped ears, which improves the Intersection Over Union (IOU). Compare the effect of different activation functions on the accuracy of the YOLO-V4 model, and use the Mish activation function as the activation function of this model. Improve the calculation of the regression positioning loss function, and use the CEIOU loss function to balance the accuracy of each category. Use improved Adam optimization function and multi-stage learning optimization technology to improve the accuracy of the YOLO-V4 model. The channel pruning algorithm was used to compress the model and distillation technology was used in the fine-tuning of the model. The final model size was only 10.77% before compression, and the test set mean Average Precision (mAP) was 93.14%. The detection speed was 112 fps, which can meet the need for real-time detection of maize harvester ears in the field. This study can provide a technical reference for the detection of the ear loss rate of intelligent maize harvesters.

Keywords: maize ear detection, YOLO-V4, channel pruning algorithm, real-time detection
DOI: 10.25165/j.ijabe.20221503.6660

Citation: Gao A, Geng A J, Zhang Z L, Zhang J, Hu X L, Li K. Dynamic detection method for falling ears of maize harvester based on improved YOLO-V4. Int J Agric & Biol Eng, 2022; 15(3): 22–32.

1 Introduction

Maize ears would drop when the maize harvester is working. Excessive maize ear loss directly affects the maize harvest quality, and the maize ear loss rate is an important indicator to measure harvest quality. At present, the detection of fallen maize ears is labor-intensive and subjective by manual identification, and when too many fallen maize ears are found, the harvester has been working for a long time and has lost its timeliness. Real-time detection of fallen maize ears can determine the ear loss rate in real-time. When the ear loss rate is too high, the driver will be notified to stop and check immediately, so as to avoid greater ear loss. Therefore, it is necessary that real-time detection of falling ears by deep learning technology when harvesting maize.

Deep learning is playing a crucial role in precision agriculture to improve crop yields[1]. Many scholars have done a lot of research with excellent results. For example, Tian et al.[2] used the YOLO-V3 algorithm to recognize apples during different growth periods. The model has a resolution of 3000×3000 pixels, and the average detection time is 0.304 s per frame which meets the real-time detection requirements. Lyu et al.[3] combined the advantages of ear detection based on deep learning and photogrammetry based on consumer UAV, proposed a deep learning model based on Mask R-CNN to detect the number of rice ears in complex scenes of paddy field. Scores, precision, recall, Average Precision (AP), and F1-score of the Mask R-CNN are 82.46%, 80.60%, 79.46%, and 79.66%, respectively. In the study of Yang et al.[4], a method that first segments object pests in two color spaces using the Prewitt operator in I component of the hue-saturation-intensity (HSI) color space and the Canny operator in the B component of the Lab color space was proposed, the segmented results for the two-color spaces were summed and achieved 91.57% segmentation accuracy.

For field crop maize, the target detection research is performed in sowing, field management, harvesting, and various segments. For example, Pang et al.[5] used an improved deep neural network to detect early maize rows and adopt the new MaxArea Mask Scoring RCNN algorithm. The crop rows could be segmented in each image, and the accuracy of estimating the emergence rate was 95.8%. Monhollen et al.[6] built a machine vision image system, which used Fast R-CNN target detection algorithm to detect the falling maize grain from maize harvest for grain loss analysis, which could be used to detect the loss in a larger sampling area and save labor. Ni et al.[7] proposed an automatic maize screening machine based on double-sided nuclear images, and embedded a deep CNN algorithm in the machine. The accuracy of maize kernel prediction in the laboratory reached 98.2%. Although the above researches are based on deep learning on maize, the detection of lost maize ears based on deep learning has not been

Received date: 2021-04-07 **Accepted date:** 2021-11-23
Biographies: Ang Gao, PhD candidate, research interest: intelligent agricultural equipment and technology, Email: 2019110103@sdau.edu.cn; **Zhilong Zhang**, PhD, Lecturer, research interest: modern agricultural equipment and computer measurement and control, Email: sdauzzl@163.com; **Ji Zhang**, PhD, Associate Professor, research interest: modern agricultural equipment, Email: sdauzhangji@163.com; **Xiaolong Hu**, Master, research interest: intelligent agricultural equipment and technology, Email: 2020120119@sdau.edu.cn; **Ke Li**, Master, research interest: intelligent agricultural equipment and technology, Email: 2020120104@sdau.edu.cn.
***Corresponding author: Aijun Geng**, PhD, Associate Professor, research interest: modern agricultural machinery design and theory. Shandong Agricultural University, Tai'an 271018, Shandong, China. Tel: +86-5388242500, Email: gengaj@sdau.edu.cn.

reported.

However, the high predictive performance of large models is often at the expense of high storage and computational costs[8], which is impractical for application to low memory and low energy-consuming edge devices. But the actual application can often only use edge equipment, so scholars have done many studies on deep model compression to reduce the size of the model and speed up the operation. For example, Wu et al.[9] proposed a YOLO-V4 deep learning algorithm based on channel pruning to detect apple blossoms in the natural environment in real-time accurately. This method performed channel on the trained YOLO-V4 model. After pruning, the number of model parameters was reduced by 96.74%, the model size was reduced by 231.5 MB, and the recognition accuracy was almost unchanged. Run et al.[10] used real-time mango monitoring by the YOLO pruning network and peeled off one subnetwork in a large-scale detection network using generalized attributional pruning monitoring method to achieve real-time accurate detection of mango in order to meet the real-time demand of low-power processors for mobile devices. Fountsop et al.[11] applied model pruning and quantification in LeNet5, VGG16, and AlecNet for plant seedling classification and validated on the Flavia dataset, showing that the model size was compressed 38-fold without considerable loss of accuracy. Although all of the above studies applied deep learning model compression to agricultural scenarios, the detection of lost maize from maize harvest on the deep learning pruning model has rarely been reported.

The objective of this study was to develop a detection method for maize ears falling after harvest based on the YOLO-V4 pruning model. Firstly, collect pictures of maize ears falling during the maize harvest to build a data set. After expanding the data set, the *K*-means algorithm is used to cluster the labeled maize samples to determine the appropriate aspect ratio of anchor, so as to improve the matching degree between a priori frame and feature layer; Then, the YOLO-V4 model is improved to calculate the regression positioning loss method to select the CEIOU function, and the extended IOU (EIOU)[12] function is improved to add the category weight. The optimizer of this model is improved, the adaptive coefficient calculation method is adopted for the search direction of the first momentum of the A Method for Stochastic Optimization (Adam)[13] optimizer, and the multi-stage learning optimization technology of the Adam optimizer and the stochastic gradient descent (SGD)[14] optimizer is adopted. Furthermore, the original YOLO-V4 model is pruned to reduce the model size and speed up the detection speed. Finally, the test set images are used for detection, and the result is that the pruning model is better than YOLO-V4 and V3 in this application. This method can realize the rapid and accurate detection of maize ear falling after harvest, meet the requirements of practical application, and provide a reference for the intelligent ear falling detection of maize harvester.

2 Materials and methods

2.1 Image acquisition and processing

The maize ear images were collected in October 2019 from the experimental field located in Shandong Agricultural University in Nanqiu village, Bianyuan Town, Feicheng City, twice under the conditions of suitable maize harvest and good weather. A smart phone with 12 million pixels was used for shooting to finish image acquisition. First, the images were collected from different angles and shooting distances of 0.3-0.5 m from the ground, and then a total of 1800 sample images were collected. Through the analysis of the data set, maize ears were divided into two categories: maize ears with skin and maize ears without skin. Some of the collected samples are shown in Figure 1. In order to assist the computer in processing the data set used in this paper, the collected images were uniformly scaled to 720×406 pixels, while the target area was labeled with labelImg annotation tool.

In order to enrich the image data set, reduce over-fitting, better extract maize image features and improve the generalization ability of the model, the data enhancement technology was used to expand the data set. The maize ear images were processed by enhanced contrast, horizontal flip, Gaussian noise, translation, and enhanced brightness. The results of data enhancement are shown in Figure 2. Finally, there were 6000 images in the expanded data set. After expansion, LabelImg labeling tool was used to label the target area, and the data set was made into VOC format. 80% of the images were used for the training of the YOLO-V4 maize ear detection model, and 20% of the images were used to test the detection effect of the model.

a. Maize ears without rind

b. Maize ears with rind

Figure 1 Images of dropped ear of maize

a. Original drawing b. Enhanced contrast c. Horizontal flip

d. Gaussian noise e. Translation f. Enhanced brightness

Figure 2 Data enhancement results of images of dropped ear maize

2.2 Detection method of maize ear based on YOLO-V4 pruning model

2.2.1 Technical route of maize ear detection model

Figure 3 shows the flowchart of the YOLO-V4 channel pruning based maize detection model proposed in this study. Firstly, the image data were obtained, and then the data preprocessing was carried out which included scaling and data enhancement of the images in the data set, making the data set into the format required by the maize ear detection model, while dividing the data set into a training set and a test set. Then the YOLO-V4 maize ear detection model was performed normal training on the preprocessed data, the initial weight and the number of training iterations were set, and the trained model after the model training was saved. The trained model was sparsely trained and then pruned. After the pruning was completed, the model was fine-tuned to restore it to model accuracy. The test set image was used to test and evaluate the completed pruning and fine-tuning model to complete the maize ear detection.

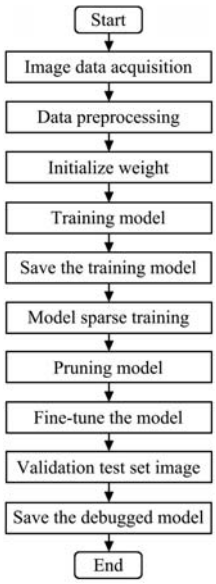


Figure 3 Flow chart of maize ear detection model

2.2.2 Maize ear detection algorithm based on YOLO-V4 network model

The YOLO-V4 network model was the fourth generation of the You Only Look Once (YOLO) series, and had higher accuracy and faster running speed than YOLO-V3[15]. Compared with the two-stage target detection Faster R-CNN, the detection speed was greatly improved[16]. The YOLO-V4 network enabled cheap 1080Ti or 2080Ti GPUs to train ultra-fast and accurate object detectors that changed the most advanced algorithms to make them more effective and more suitable for single GPU training. The current target detection model generally consisted of Input, Backbones, Neck, and Heads[15]. As shown in Figure 4, based on the YOLO-V4 maize ear detection network model structure diagram, the Backbones were Cross Stage Partial Darknet53 (CSPDarknet53)[17], Neck: Spatial Pyramid Pooling (SPP)[18], and Path Aggregation Network (PAN)[19], Head: YOLO-V3.

The backbone extraction network of YOLO-V4 was CSPDarknet53 which was the improvement of backbone extraction network YOLO-V3 Darknet53. The activation function of DarknetConv2D was changed from Leaky-ReLU[20] to Mish[21], the convolution block was changed from DarknetConv2D_ BN_Leaky to DarknetConv2D_BN_Mish, and the structure of Resblock_body was modified using the Cross Stage Partial net (CSPnet) structure.

YOLO-V4 used the SPP structure and the PAN structure in the feature pyramid. The SPP structure performed three DarknetConv2D_BN_Leaky convolutions on the last feature layer of CSPDarknet53 and used four different scales of maximum pooling for processing, which increased the field of perception and facilitated the separation of most notable contextual features. In YOLO-V4, the PAN structure was used on the main three effective feature layers to promote the flow of information. The PAN structure was characterized by repeated feature extraction through bottom-up path enhancement, and accurate low-level positioning signals to enhance the entire feature Hierarchy, thereby shortening the information path between low-level and top-level features[22].

YOLO-V4 used the probe head of YOLO-V3 as a multi-feature layer to detect the target. Three feature layers were respectively the middle layer, the middle and lower layer, and the bottom layer. It was extracted by a 3×3 convolutional layer and adjusted to the required number of channels by 1×1 convolution. The number of output channels was $3K+15$, where 3 represented the three sizes of anchor boxes set for each layer; K represented the number of categories; 5 could be divided into 4+1, which were 4

parameters of the target box and 1 parameter to judge whether there was an object in the box[23].

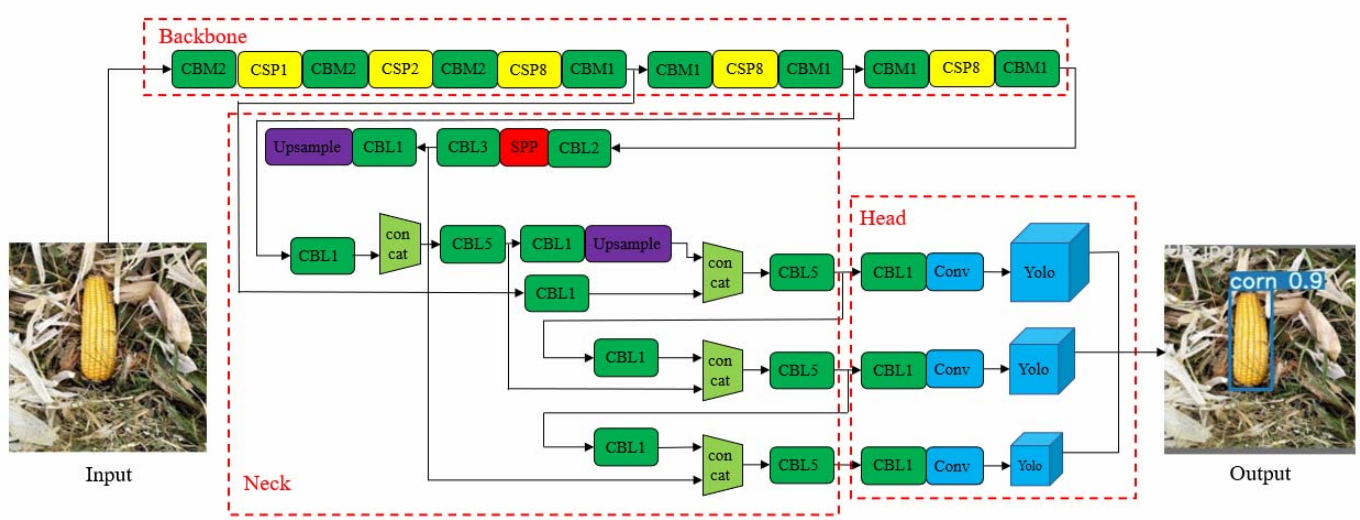


Figure 4 Target detection structure diagram

Compared with YOLO-V3, YOLO-V4 improved the loss of Bouding Box (BBox) region by using Complete Intersection Over Union (CIOU)[24] instead of Mean Square Error (MSE) as the regression function of the box. CIOU considers not only the center distance of the two detection frames, but also the scale information of the overlapping area and the aspect ratio, which enabled the rectangular BBox to achieve better convergence in the regression problem, and the penalty term could be defined as the follows:

$$R_{\mathrm{CIOU}} = \frac{\rho^2(b,b^{gt})}{c^2} + \alpha v \tag{1}$$

where, ρ is Euclidean distance between two center points; b is the center point of prediction box; b^{gt} is the center point of the real frame; c is Euclidean distance between two center points; v measures the consistency of aspect ratio; α is a positive trade-off parameter.

$$\alpha = \frac{v}{(1-\mathrm{IOU})+v} \tag{2}$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{3}$$

The loss function could be defined as:

$$L_{\mathrm{CIOU}} = 1 - \mathrm{IOU} + \frac{\rho^2(b,b^{gt})}{c^2} + \alpha v \tag{4}$$

where, IOU is the ratio of intersection and union of prediction frame and real frame; w^{gt} is the width of the real box; h^{gt} is the height of the real box; w is the width of prediction box; h is the height of prediction box.

2.2.3 Improvement of maize ear detection model based on YOLO-V4

(1) Regional suggestion network based on *K*-means algorithm

Since the data type and the number of the self-built maize ear data set were very different from the MSCOCO data set used in the original model test, if the original anchor box aspect ratio was continued to be used, the YOLO detection head would calculate the intersection and compare the IOU screening the accuracy of BBoxdecreases, which affected the detection performance. Therefore, the *K*-means[25] algorithm was used to cluster the aspect ratio of the anchor box, and find the most suitable anchor box aspect ratio to improve the adaptability of the model.

First, multiple clustering was performed on the aspect ratio value of the maize ear position frame marked in the self-built maize ear data set with the *K* value between 2-10, and the elbow method was used to estimate the best *K* value, which was, the most obvious change in the slope of the curve was the best *K* value, as shown in Figure 5. It could be seen that the change was the most obvious when *K*=6, and finally the *K* value was selected as 6 for cluster analysis, and the result was 6 cluster centers. Finally, it was determined that the aspect ratio of the anchor was 0.8, 1.6, 2.0, 3.0, 3.8, 4.9, and the size would not be changed.

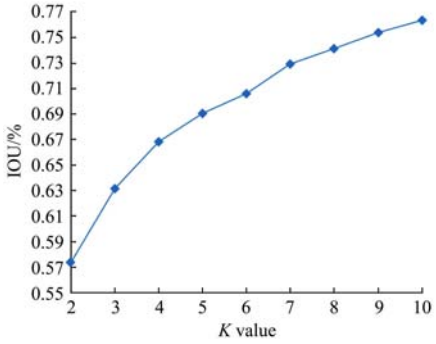


Figure 5 Change curve of average IOU and *K* value

(2) Improvement of activation function, BBox regression loss, and optimization function of YOLO-V4 model

The role of activation function was to introduce nonlinearity into the network model and strengthen the learning ability of the neural network. A good activation function could make the gradient propagate more effectively without too much additional computational cost. In order to select the activation function suitable for the YOLO-V4 model in this study, Mish function[21], Leaky ReLU function[20], and Swish function[26] were used as the comparison test of the activation function of the YOLO-V4 model in this study.

A better positioning regression loss was set to solve the problem of inaccurate regression of different shapes of objects, thereby the faster convergence and the better performance were achieved. For the regression positioning loss YOLO-V4 used CIOU, the CIOU loss function would produce unreasonable updates when updating the width and height of the prediction box, and the calculation of "v" in the equation was too complicated, so the EIOU[27] loss function was introduced. EIOU included IOU Loss, distance loss, and aspect ratio loss, an equation similar to distance loss was used to describe the aspect ratio loss, which was defined as Equation (5).

$$L_{\text{EIOU}} = L_{\text{IOU}} + L_{\text{distance}} + La_{\text{spect}} = 1 - \text{IOU} + \frac{\rho^2(A,B)}{C} + \frac{\rho^2(w^A,w^B)}{C_W^2} + \frac{\rho^2(h^A,h^B)}{C_h^2} \quad (5)$$

where, C_W is the width of the minimum closing box of two bounding boxes; C_h is the height of the minimum closing box of two bounding boxes.

Through preliminary test data, the experimental data are shown in Figure 6. The recognition effect of maize ears with skins was not very satisfactory, which was lower than that of maize ears without skins. By analyzing the images of maize ears with skins in the data set, the maize with skins could be recognized. The color of the ears was not much different from the background color, and some of the skins were similar to the maize ears with the skin, which increased the difficulty of detecting the maize ears with the skin. Therefore, an improved method of CEIOU was proposed. CEIOU added weight to the category of maize ears with skin. In order to improve the accuracy of its detection, the equation was as follows:

$$L_{\text{CEIOU}} = A_{cls} L_{\text{EIOU}} \quad (6)$$

where, A_{cls} refers to the weights represented by different categories. The ear category of maize with skin is set to 1.3, and the category of maize ear without skin is set to 1.0.

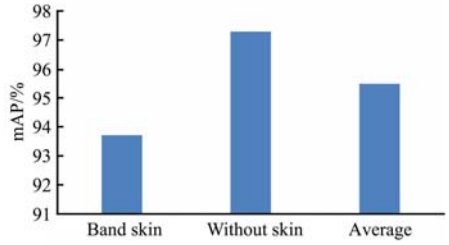


Figure 6 mAP of two categories

In the process of model training, each forward propagation would get the loss value of output value and real value. Generally, the optimization function was used to find the local minimum loss value. By calculating the gradient of error function relative to the weight parameter, the weight parameter was updated in the opposite direction of the loss function gradient to optimize the model. The Adam optimization function was advanced and more computationally efficient. It could automatically update the neural network weights iteratively. The Adam optimization function formula was as followings, assuming that the objective function $f_t(\theta)$ was a random function of the parameter θ in the t iteration, the optimization process of the YOLO model was to find a suitable θ to make $f_t(\theta)$ the minimum value, with the help of small batch gradient method of the sample function[27]. The equation was as follows:

$$g_t = \nabla_\theta f_t(\theta) \quad (7)$$

where, g_t represents the gradient of $f_t(\theta)$ with respect to θ, that is, the partial derivative vector of $f_t(\theta)$ with respect to θ under the number of iterations t.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (8)$$

$$v_t = \beta_2 m_{t-1} + (1-\beta_2) g_t^2 \quad (9)$$

where, m_t is the exponential moving mean; v_t is the square gradient, and β_1, $\beta_2 \in [0,1)$ represent the decay rate of the exponential moving mean.

The first-order moment estimation of the gradient was used for the moving mean itself. However, when these moving mean values were initialized, especially when the initial time and decay rate were very small, the deviation of the moment estimation tended to be 0, so the deviation shall be corrected to some extent:

$$\hat{m} = \frac{m_t}{1-\beta_1^t} \quad (10)$$

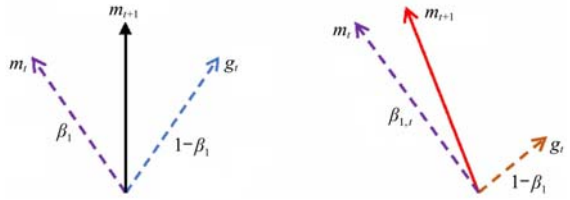
$$\hat{v} = \frac{v_t}{1-\beta_2^t} \quad (11)$$

For each iteration, the parameter value would be updated once. The formula is as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}} + \varepsilon} \hat{m} \quad (12)$$

where, η represents the learning rate; ε is a parameter constant.

In the Adam-based optimizer, the next search direction was determined by the first momentum m_t of the current gradient. If an undesirable gradient pointed in a direction away from the global optimum, the direction of the first momentum became far away from the approximate optimum, which made its search capabilities seriously deteriorated[28]. Figure 7a shows how the first momentum of the ideal state was distorted by the undesirable gradient, Figure 7b shows the non-ideal state, the first momentum was not distorted by the desired gradient, and the next search direction would deviate from the optimal solution.



a. Ideal state b. Non-ideal state

Figure 7 Influence of the outlier gradient value on the direction of the first momentum[28]

For this reason, when calculating the gradient of the first momentum, the difference between m_t and g_t was checked. The ratio of β_1 according to the degree of difference was adjusted between them, so that the force of g_t in m_{t-1} was minimized in the next iteration of calculation[29]. This mechanism is defined as:

$$m_t = \beta_a m_{t-1} + (1-\beta_a) g_t \quad (13)$$

where, β_a is the adaptive coefficient, defined as:

$$\beta_a \propto |m_t - g_t| \quad (14)$$

A determined the proportion of their differences accumulated by B. The calculation formula is as follows:

$$\beta_a = \frac{H_{t-1}}{H_{t-1} + h_t} \quad (15)$$

where, h_t represents the similarity between g_{t1} and m_{t2}, measured by the following equation:

$$h_t = 2d\left(d + \sum_{j=1}^{d} \frac{(g_{t,j} - m_{t-1})^2}{v_{t-1,j} + \varepsilon} \right)^{-1} \quad (16)$$

where, m_{t-1} and v_{t-1} are the first and second momentum calculated in the previous step, namely t_1.

The research results showed that in a complex solution space, a hybrid compensation method combining multiple strategies could significantly improve the search for approximate optimal solutions[30]. After experimentation, it was found that when the SGD optimization algorithm was used alone, the learning rate was too large and the algorithm was difficult to converge, and the learning rate was too small, which would cause the algorithm to converge very slowly; when the Adam optimization algorithm was used alone, the final training, the result was often worse than using SDG alone, but the advantage was that it had a self-applicable learning rate and the algorithm converged quickly. Therefore, this article combined the advantages of the two

optimization algorithms and proposed a multi-stage optimization algorithm that adapted to this data set. The Adam optimization algorithm was used for the first 200 rounds, and the optimization algorithm for the next 100 rounds was SGD in the second paragraph. The initial learning rate was set to 0.0001 and the momentum parameter was set to 0.9 in all trails. After each generation, the learning rate was reduced to the original 0.9.

2.2.4 Compression of maize ear detection model based on YOLO-V4

The practical application scenario of maize ear detection was that a maize harvester, it could only run on embedded devices for the desktop computer was too large to be installed and used, while the amount of computation required for the trained depth model was too large for embedded devices. The model needed to be compressed to minimize the amount of storage and reasoning calculations occupied by the model while ensuring a small loss of accuracy. The essence of the channel pruning algorithm was to eliminate unimportant channels and their associated input-output relationships by identifying network channels[8]. Therefore, the maize ear detection model of YOLO-V4 network model was used for network pruning and knowledge distillation was used in fine-tuned pruning network.

As shown in Figure 8, the output channels convoluted by different layers were sparsely regularized on the left, and the Batch Normalization layer was sparsely trained to get a set of weights. The channels with smaller weights (yellow) in the output channel and the neurons with smaller contributions (red) were clipped. The pruned network retained the higher weight channels (blue) as shown on the right side of Figure 8. The training objective function of the model pruning method is

$$L(w)=\sum_{(x,y)}C(Net(x;W),y)+\lambda\sum_{\gamma}g(\gamma) \tag{17}$$

where, (x, y) is the training input and output; γ is the scaling factor; W is the trainable weight, the first term is the normal training of the corresponding convolutional network; $g()$ function is the punishment of the sparse scaling factor; λ is the balance factor of the two terms.

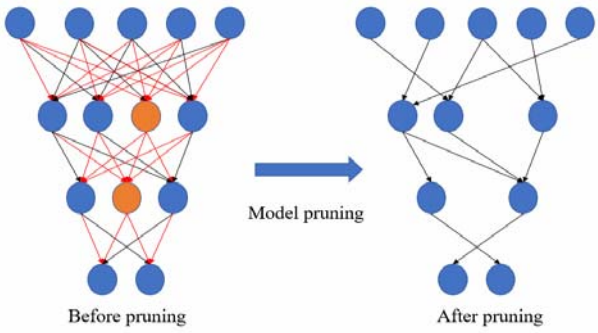


Figure 8 Schematic diagram of neural network pruning

Knowledge distillation[31] is a way for teachers to guide students in model transfer training. Its structure is shown in Figure 9. The purpose was to use high-precision large models to guide small model training to improve its accuracy. In this study, the model before pruning was used as the teacher model, and the model after pruning was used as the student model. In the training, boxloss and classloss were distinguished, and students did not directly learn from teachers. Students, teachers, and GT found the distance of L2 respectively, and added a loss of student and GT when the student was greater than the teacher.

The main steps of compression of the maize ear detection model based on YOLO-V4 are as follows:

1) Sparse training. A scale factor was introduced to each channel and it was multiplied by the output of the channel, a sparsity penalty term L1 was added to each convolutional layer backpropagation process, which was used to constrain the scale factor of the BN layer of the maize ear measurement model. The model structure was sparse, and the global scale attenuation method was adopted. The scale attenuation was 100 times when epochs were performed 0.6 iterations.

2) Channel pruning. After the sparse training was completed, the importance of the channel was determined according to the size of the scale factor, and the channel was pruned according to different pruning rates.

3) Fine-tune the trimmed model. In order to avoid excessive loss of model accuracy after pruning, it was necessary to perform secondary training and fine-tuning, and use knowledge distillation in the fine-tuning to help the model accuracy rise. The main parameter settings of the maize ear detection model compression are listed in Table 1.

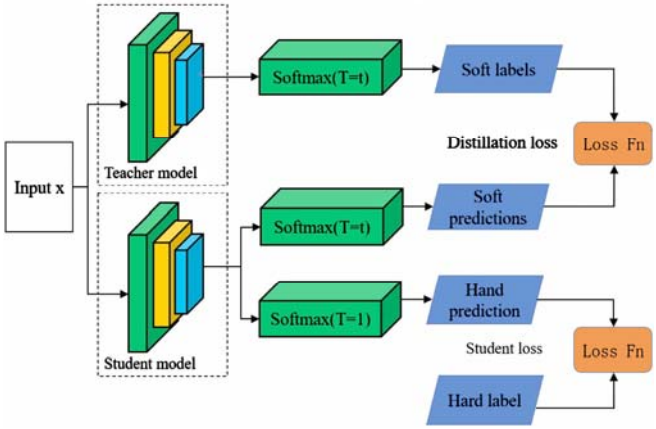


Figure 9 Model of network slimming

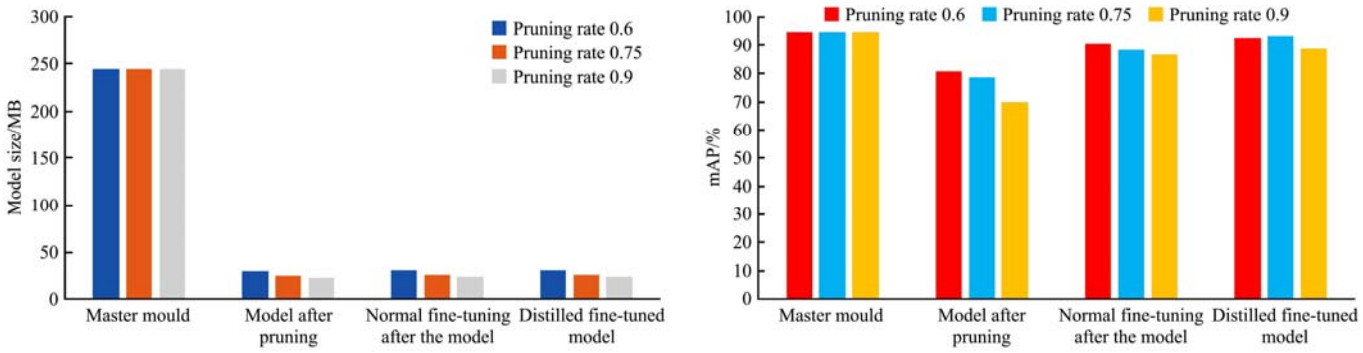
Table 1 Model compression parameter settings

Stage	Parameters	Value
Sparse training	Learning rate	0.0001
	Batch_size	6
	Epochs	400
	Scale sparse rate	0.005
	Sparse factor	0.001
Model fine-tuning	Learning rate	0.0010
	Batch_size	16
	Pruning rate	0.75
	Epochs	200

After sparse training, different pruning rates were selected to perform channel pruning and knowledge distillation strategy fine-tuning model for the maize ear detection model of YOLO-V4. A large number of experiments showed that different pruning rates had different effects on the compression and accuracy of the model. The experiment chose three pruning rates of 0.60, 0.75, and 0.90 to perform channel pruning on the model, and the changes in the size and accuracy of the model are shown in Figure 10. It could be seen that the method with a pruning rate of 0.90 had the highest compression rate of the model, and the method with a pruning rate of 0.60 had the highest average accuracy rate of the model after pruning. In the fine tuning, the accuracy of the model after knowledge distillation was improved compared with that without knowledge distillation. Through the above comparative test results, the reliability of this method could be proved.

To consider the combined effects of three factors: the size of

the model after pruning, the average accuracy, and the test time of a single image, the final pruning rate was set to 0.75, the knowledge distillation strategy fine-tuned model size was 26.3 MB, and the average progress of the test set was 93.14%. As shown in Figure 11, the channel changes in each layer of YOLO-V4 maize ear detection model after pruning, red was the number of channels without pruning, and green was the number of channels remaining after pruning. It was obvious that the number of each channel was decreasing after pruning. After 50 layers of channel layer, the number of channels in each layer was greatly reduced, so the maize ear detection model of YOLO-V4 was compressed after using channel pruning.



a. Model size comparison diagram b. Average precision comparison chart

Figure 10 Comparison diagram of model size and average precision

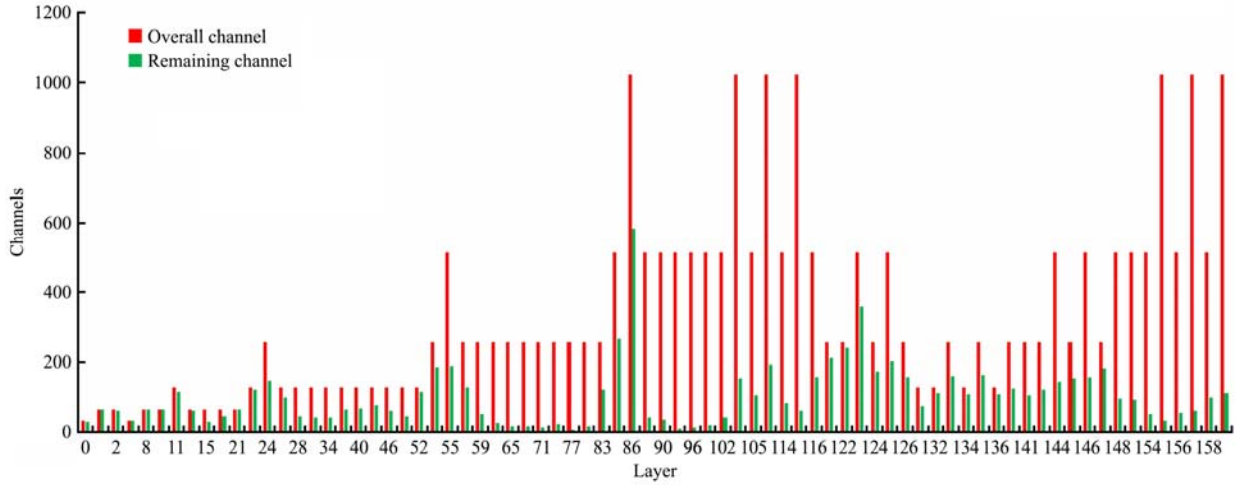


Figure 11 Model channel parameters before and after pruning

2.5 Test environment and evaluative index

In this study, the hardware test environment processing platform was a desktop computer, the processor was Intel Pentium G4560, the main frequency was 3.5 GHz, and the GPU was GeForce GTX 1060 8 G. The software test environment was Ubuntu (18.04) Linux system, the machine learning library was Pytorch 1.5.2, and the parallel computing architecture was CUDA 10.2.

In order to analyze and evaluate the performance of the training model in this study, Recall, Precision, F_1 score and mAP were calculated. The indexes were defined as follows:

$$P = \frac{T_P}{T_P + F_P} \times 100\% \quad (18)$$

$$R = \frac{T_P}{T_P + F_N} \times 100\% \quad (19)$$

$$\mathrm{AP} = \int_0^1 P(R)\mathrm{d}R \quad (20)$$

$$F_1 = \frac{2 \times P \times R}{P + R} \times 100\% \quad (21)$$

$$\mathrm{mAP} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \times 100\% \quad (22)$$

where, P is precision rate; R is recall rate; T_P is the number of positive samples that were correctly predicted; F_P is the number of sub-samples predicted to be positive samples; F_N is the number of positive samples that were predicted to be negative samples; F_1 is measurement of average precision P and recall rate R, %; AP is average precision; mAP is the average value of the average precision.

3 Results and discussion

3.1 Different model results and analysis

In the case of the same hardware environment and software environment, the existing advanced target detection model with the YOLO-V4 model was compared. The benchmark data set PASCAL VOC2007 was used in the test data set, because the optimal hyperparameters of each model were different. Finally, the optimal detection results of each model were selected as shown in Table 2. It could be seen from Table 2 that in the two-level target detection algorithm Faster R-CNN[32], different feature extraction networks had different detection accuracy. Using ResNet101[33] as the feature extraction network Faster R-CNN was higher than the average detection accuracy of using ResNet50[33] as the feature extraction network, which showed that the ResNet network after using the residual network had a higher accuracy of feature extraction with the deepening of the number of layers. In the first-level target detection algorithm, the average detection accuracy of YOLO-V4-EIOU was 2.1% higher than the average

detection accuracy of YOLO-V3, and 1.2% higher than the average detection accuracy of YOLO-V4, indicating that YOLO-V4-EIOU was superior to YOLO-V3 and V4 in recognition accuracy. At the same time, comparing the whole test results, the detection speed of the first-level target detection algorithm was significantly higher than that of the second-level target detection algorithm. It showed that the first-level target detection algorithm directly converted the detection problem into a regression problem, which greatly improved the detection speed, and used various. The technique of improving the accuracy made the detection accuracy slightly higher than that of the secondary target detection algorithm. Among the comparison models, the YOLO-V4-EIOU model performed the best, with a frame rate of 33% and a total average detection accuracy mAP of 77.2%.

Table 2 Comparison test results of different recognition models

Model	Backbone	Detection speed/fps	mAP/%
Faster R-CNN	ResNet50	6	73.5
Faster R-CNN	ResNet101	5	74.7
YOLO-V3	Darknet-53	28	75.1
YOLO-V4	CSPDarknet-53	32	76.0
YOLO-V4-EIOU	CSPDarknet-53	33	77.2

3.2 Results and analysis of improved activation functions, BBox regression loss and optimization functions

In the comparison test of different activation functions of the YOLO-V4 model, the performances of Mish, Leaky ReLU, and Swish functions were different. The results are shown in Table 3. It could be seen from the test that different activation functions had different effects on the YOLO-V4 model. When the Leaky ReLU function was the activation function, the scores of R, P, F_1, and mAP were the lowest in the experiment. The four evaluation indexes of Mish function and Swish function were similar, but the total average recognition accuracy of mish function as activation function was 95.6%, the total F1 was 91.1%, and the total mAP was 95.6%.

Table 3 Experimental results were compared with different activation functions

Activation function	Grain type	R/%	P/%	F1/%	mAP/%
Mish	Band skin	88.3	95.7	91.9	93.9
	Without skin	83.7	97.9	90.2	97.3
	All	86.0	96.8	91.1	95.6
Leaky ReLU	Band skin	83.4	90.6	86.8	91.7
	Without skin	81.6	88.0	84.0	93.1
	All	82.5	89.3	85.8	92.4
Swish	Band skin	95.1	87.4	86.2	94.3
	Without skin	81.7	84.0	82.8	93.1
	All	83.4	85.7	84.5	93.7

This study selected three calculation IOU variants such as DIOU, CIOU, EIOU, and EIOU's improved CEIOU for comparative experiments. The results are listed in Table 4. The recall rate, precision, F1 score, and average precision were quantitatively evaluated. The performance of different calculated IOU variants was different. In the recall rate, YOLO-V4-CEIOU had the highest score of 94.1% for maize ears with skin, and the lowest score of YOLO-V4-CIOU was 83.7% for maize ears without skin, with accuracy. YOLO-V4-CIOU performed best, and F_1 score was YOLO-V4-CIOU performed best. Using the CEIOU model to identify maize ears with skin mAP had increased by 1.4% compared with the EIOU model for identifying ears of maize with skin. At the same time, the average accuracy of maize ears with skin using the CEIOU model differed by only 0.3% from the mAP without maize skin. It demonstrated that the increased weight for the maize ear category with the skin improved its detection accuracy and balanced the recognition accuracy of the two categories.

Table 4 Experimental results of different IOU

Band skin	Grain type	R/%	P/%	F1/%	mAP/%
YOLO-V4-DIOU	Band skin	90.7	90.2	90.4	91.9
	Without skin	89.5	87.2	88.3	93.1
	All	90.1	88.7	89.4	92.5
YOLO-V4-CIOU	Band skin	88.3	95.7	91.9	93.9
	Without skin	83.7	97.9	90.2	97.3
	All	86.0	96.8	91.1	95.6
YOLO-V4-EIOU	Band skin	94.2	89.9	92.1	95.3
	Without skin	92.0	86.9	89.4	97.2
	All	93.1	88.4	90.7	96.1
YOLO-V4-CEIOU	Band skin	93.4	89.4	91.4	96.8
	Without skin	94.1	89.8	92.2	96.5
	All	93.7	89.6	91.8	96.8

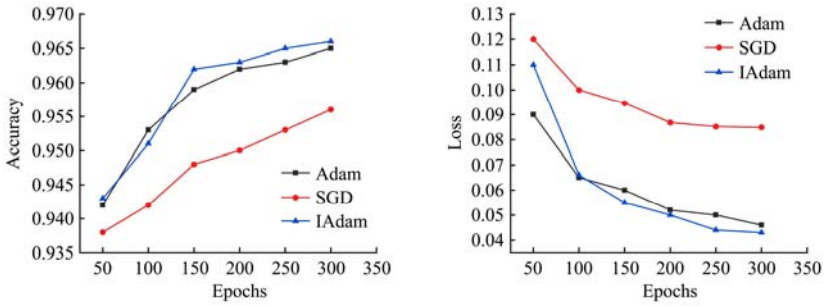
In this study, The IAdam optimizer was used to verify its performance after improvement. As shown in Table 5, the accuracy of the YOLO-V4 model of the three optimizers of SGD, Adam, and IAdam in 300 epochs was compared under different learning rates. It could be seen that the accuracy of the model was different under different learning rates. On the whole, the SGD optimizer could not obtain a good accuracy rate when the learning rate was small, and the IAdam accuracy rate should exceed the Adam optimizer when the learning rate was appropriate. The accuracy of IAdam had achieved good results under different learning rates. The learning rate was $1e^{-4}$, and the highest accuracy was 96.8%, which showed the superiority of the improved optimizer.

At the same time, a qualitative analysis of the optimizer's accuracy and training loss was conducted in different epochs. The accuracy of the Adam, SGD, and IAdam optimizers under different iterations of the YOLO-V4 model is shown in Figure 12a. In 300 epochs, the verification accuracy had reached the highest level, and the verification accuracy of IAdam was higher than that of Adam and SGD. Figure 12b shows the loss curves of Adam, SGD, and IAdam optimizers under different iterations of the YOLO-V4 model. It could be seen that the three optimizers were all less than 0.1 in the later stage of training loss. The Adam optimizer converged quickly, and with the IAdam optimizer a smaller loss was achieved.

Table 5 Experimental results of optimizer with different learning rates

Optimizer	Learning rate									
	$1e^{-1}$	$1e^{-2}$	$1e^{-3}$	$1e^{-4}$	$1e^{-5}$	$1e^{-6}$	$1e^{-7}$	$1e^{-8}$	$1e^{-9}$	$1e^{-10}$
Adam Optimizers	96.5	96.5	96.5	96.5	95.4	95.4	95.4	95.1	95.1	95.1
SGD	95.6	95.6	95.6	94.5	94.5	94.5	93.8	93.8	93.8	93.8
IAdam	96.4	96.6	96.8	96.8	96.6	96.6	96.6	96.4	96.1	96.1

Note: The activation function of both groups was mish.



a. Validation accuracy b. Training loss

Figure 12 Comparison of different optimizer accuracy validation with loss

3.3 Experimental results and analysis of compression model

In the maize ear detection model of YOLO-V4, the activation function was Mish function, the optimization function was IAdam function, the pruning rate was 0.75, and the final model size was 26.3 MB. In order to further illustrate the reliability of the model after pruning, this study conducted a comparative test between YOLO-V3, YOLO-V4, YOLO-V4-tiny and YOLO-V4 pruning models, and R, P, F_1 and mAP were quantitatively evaluated. The test results were shown in Table 6. The size of this model was 26.3 MB, the total F_1 was 91.4%, and the total mAP was 93.14%. The mAP of maize ears without skin was 3.4% higher than that of maize ears with skin, and the frame rate was 112 fps, which was 3.5 times higher than that of YOLO-V4 model, and the total mAP was only 1.3% different. In the comparison test, the minimum value of the YOLO-V4-tiny model in the model size category was 22.5 MB, but the scores of P, F1 and mAP in the test were the lowest in comparison. The smallest frame speed category was the YOLO-V3 model, which was 28 frames, and could not meet the use requirements of embedded devices. Although the frame rate of this model was not as high as that of the YOLO-V4-tiny model, the total F_1 was 2.6 percentage points higher and the total mAP was 1.44 percentage points higher than that of the YOLO-V4-tiny model. The detection speed of 112 fps of this model could meet the requirements of embedded devices.

The maize ear recognition effect diagrams are shown in Figure 13. The four different models of YOLO-V3, YOLO-V4, YOLO-V4-tiny and this model could reach a recognition accuracy of more than 0.7 for a single maize ear. As far as the overall recognition accuracy is concerned, the recognition accuracy of YOLO-V4 was the highest and missing ear detection in YOLO-V4 did not happen, but when there were two maize ears in a picture, both YOLO-V3 and YOLOV4-tiny had missed maize ear detection. Although the recognition accuracy of this model decreased when there were two maize ears in a picture, there was no missed detection, which further illustrated the effectiveness of this model.

Table 6 Comparison test results of model before and after compression

Model	Model size /MB	Frame rate/s	Grain type	R/%	P/%	F_1/%	mAP/%
YOLO-V3	235	28	Band skin	71.5	96.4	82.1	92.5
			Without skin	67.3	98.6	80.0	94.9
			All	69.4	97.5	80.1	93.7
YOLO-V4-CIOU	244	32	Band skin	88.3	95.7	91.9	93.9
			Without skin	83.7	97.9	90.2	97.3
			All	86.0	96.8	91.1	95.6
YOLO-V4-tiny	22.5	250	Band skin	55.5	97.5	70.7	91.2
			Without skin	81.0	98.4	88.8	92.2
			All	68.1	97.9	79.8	91.7
YOLO-V4-pruning	26.3	112	Band skin	86.6	97.5	91.7	92.6
			Without skin	86.8	97.8	92.0	96.0
			All	86.7	97.7	91.9	94.3

a. YOLO-V4

b. YOLO-V3

c. YOLO V4-tiny

d. Proposed model of this study

Figure 13 Effects of different models of maize ear recognition

4 Conclusions

In this study, a method for detecting fallen ears of maize based on the YOLO-V4 pruning model was proposed. The existing classic target detection methods were comprehensively discussed and comparative experiments were made to analyze the advantages and disadvantages of the models. The *K*-means algorithm was used to cluster the proportions of anchor frames. The anchor frames suitable for this data set were selected. Secondly, the performance of different activation functions in the model was compared and Mish activation function was selected to optimize the Mish activation function. The CEIOU function was improved in EIOU function, which added weight to the category of peeled maize ear and balanced the recognition accuracy of the two categories. The optimizer of this model was improved, the multi-stage learning optimization technology of Adam optimizer combined with SGD optimizer was adopted, and the adaptive coefficient calculation method for the search direction of the first momentum of Adam optimizer was adopted, so that the YOLO-V4 maize detection model could achieve the best speed and accuracy.

The maize ear detection model of YOLO-V4 was sparsely trained, pruned and fine-tuned, and the distillation knowledge was used in the process of fine tuning. Finally, the compressed model size after pruning and knowledge distillation was only 10.77% of the original model. The model was 10.77%, the accuracy rate in the test set was 93.14%, and the detection speed was 112 fps. The result proved that the speed of the maize falling ear target detection method based on the YOLO-V4 pruning model and the detection accuracy rate met the requirements.

In this study, the method for detecting maize falling ears based on the YOLO-V4 pruning model was proposed which could achieve the accuracy and the speed of practical application through training and learning and test bench testing, so further research could be done on the basis of this research. In the future, the YOLO-V4 pruning model would be transplanted to embedded applications like jeston nano, and installed on the maize harvester to improve its practical application value.

Acknowledgements

This work was financially supported by the Shandong Provincial Key Science and Technology Innovation Engineering Project (Grant No. 2018CXGC0217) and the 13th Five-Year National Key Research and Development Program (Grant No. 2018YFD0300606).

[References]

[1] Jin X B, Yu X H, Wang X Y, Bai Y T, Su T L, Kong J L. Deep learning predictor for sustainable precision agriculture based on internet of things system. Sustainability, 2020; 12(4): 1–18.

[2] Tian Y, Yang G, Wang Z, Wang H, Li E, Liang Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Computers and Electronics in Agriculture, 2019; 157: 417–426.

[3] Lyu S X, Noguchi N, Ospina R, Kishima Y. Development of phenotyping system using low altitude UAV imagery and deep learning. Int J Agric & Biol Eng, 2021; 14(1): 207–215.

[4] Yang Z K, Li W Y, Li M, Yang X T. Automatic greenhouse pest recognition based on multiple color space features. Int J Agric & Biol Eng, 2021; 14(2): 188–195.

[5] Pang Y, Shi Y, Gao S, Jiang F, Veeranampalayam-Sivakumar A N, Thompson L, et al. Improved crop row detection with deep neural network for early-season maize stand count in UAV imagery. Computers and Electronics in Agriculture, 2020; 178: 105766. doi: 10.1016/j.compag.2020.105766.

[6] Monhollen N S, Shinners K J, Friede J C, Rocha E M, Luck B L. In-field machine vision system for identifying corn kernel losses. Computers and Electronics in Agriculture, 2020; 174: 105496. doi: 10.1016/j.compag.2020.105496.

[7] Ni C, Wang D, Vinson R, Holmes M, Tao Y. Automatic inspection machine for maize kernels based on deep convolutional neural networks. Biosystems Engineering, 2019; 178: 131–144.

[8] Yeom S K, Seegerer P, Lapuschkin S, Binder A, Samek W. Pruning by explaining: a novel criterion for deep neural network pruning. Pattern Recognition, 2021; 12: 107899. doi: 10.1016/j.patcog.2021.107899.

[9] 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(4): 105742. doi: 10.1016/j.compag.2020.105742.

[10] Run S, Li, T X, Yamaguchi Y. An attribution-based pruning method for real-time mango detection with YOLO network. Computers and Electronics in Agriculture, 2020; 169: 105214. doi: 10.1016/j.compag.2020.105214.

[11] Fountsop A N, Fendji, J, Atemkeng M. Deep learning models compression for agricultural plants. Applied Sciences, 2020; 10(19): 6866. doi: 10.3390/app10196866.

[12] Peng H Y, Yu S Q. A systematic IOU-related method: Beyond simplified regression for better localization. IEEE Transactions on Image Processing, 2021; 30: 5032–5044.

[13] Kingma D, Ba J. Adam: a method for stochastic optimization. Computer Science. 3rd International Conference for Learning Representations, San Diego, 2015; arXiv. doi: 10.48550/arXiv.1412.6980.

[14] Ketkar N. Stochastic gradient descent. In Deep learning with Python. Apress, Berkeley, CA, 2017; pp.113–132.

[15] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection. arXiv, 2020. doi: 10.48550/arXiv.2004.10934.

[16] Tian Y, Yang G, Wang Z, Wang H, Li E, Liang Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Computers and Electronics in Agriculture, 2019; 157: 417–426.

[17] Wang C-Y, Liao H-Y, Wu Y-H, Chen P-Y, Yeh I-H. CSPNet: A new backbone that can enhance learning capability of CNN. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020; pp.1571–1580. doi: 10.1109/CVPRW50498.2020.00203.

[18] He K, Zhang X, Ren S, Sun J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015; 37(19): 4–16.

[19] Shu L, Lu Q, Haifang Q, Jianping S, Jiaya J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018; pp.8759–8768. doi: 10.1109/CVPR.2018.00913.

[20] Zhang X, Zou Y, Shi W. Dilated convolution neural network with LeakyReLU for environmental sound classification. In: 2017 22nd International Conference on Digital Signal Processing (DSP). IEEE, 2017; pp.1–5. doi: 10.1109/ICDSP.2017.8096153.

[21] Misra D. Mish: A self regularized non-monotonic neural activation

function. 2019. arXiv: 1908.08681, 4.

[22] Liu S, Qi L, Qin H, Shi J, Jia J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: pp.8759–8768. doi: 10.1109/CVPR.2018.00913.

[23] Long X, Deng K, Wang G, Zhang Y, Dang Q, Gao Y, et al. PP-YOLO: An effective and efficient implementation of object detector. arXiv, 2020. doi: 10.48550/arXiv:2007.12099.

[24] Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D. Distance-IOU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020; 34(7): 12993–13000.

[25] Dhanachandra N, Manglem K, Chanu Y J. Image segmentation using *K*-means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science, 2015; 54: 764–771.

[26] Ramachandran P, Zoph B, Le Q V. Searching for activation functions. arXiv, 2017. doi: 10.48550/arXiv.1710.05941.

[27] Peng H, Yu S. A Systematic IOU-Related Method: Beyond Simplified Regression for Better Localization. IEEE Transactions on Image Processing, 2021; 30: 5032–5044.

[28] Kim K, Choi Y. HyAdam C: A new Adam-based hybrid optimization algorithm for Convolution Neural Networks. Sensors, 2021; 21(12): 4054. doi: 10.3390/s21124054.

[29] Iiboudo W, Kobayashi T, Sugimoto K. Robust stochastic gradient descent with student-t distribution based first-order momentum. IEEE Transactions on Neural Networks and Learning Systems, 2020; 99: 1–14.

[30] Wu Xu, Qi Z, Wang L J, Yang J, Xia X. Apple detection method based on light-YOLOV3 convolutional neural network. Transactions of the CSAM, 2020; 51(8): 17–25. (in Chinese)

[31] Chen G, Choi W, Yu X, Han T, Chandraker M. Learning efficient object detection models with knowledge distillation. In Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017; pp.742–751.

[32] Ren S, He K, Girshick, R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017; 39(6): 1137–1149.

[33] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016; pp.770–778.