Computing and Informatics, Vol. 44, 2025, 151–175, doi: 10.31577/cai_2025_1_151

RESEARCH ON DENSE DETECTION ALGORITHM FOR BROWN MUSHROOM BASED ON IMPROVED YOLOV7

Shanheng Tan, Shengjie Ma, Xinglu Ma*

Qingdao University of Science and Technology
Qingdao, Shandong Province, 266061, China
e-mail: shanheng988@gmail.com, msjqd2023@163.com, qdmxl@163.com

Weian Dong, Yangbo Sheng

Hohai University, Changzhou
Jiangsu Province, 213200, China
e-mail: 20221032@hhu.edu.cn

Chunxu Li*

Mechanical Engineering Department, Swansea University
UK, SA1 8EN
e-mail: chunxu.li@swansea.ac.uk

Abstract. In the complex environment of industrialized brown mushroom cultivation, a dense brown mushroom detection algorithm based on improved YOLOv7 is proposed to address the issues of low real-time detection accuracy and speed, and the high false detection rate of picking robots in densely grown brown mushroom clusters. To prevent network degradation, improve the detection accuracy and speed of the network, and reduce the network's computational cost, the ELAN_PS module is introduced to replace the original ELAN module. The AFPN network is used to replace the original network's Neck part for multi-scale fusion, allocating different spatial weights to feature maps to enhance the model's ability to separate

* Corresponding author

dense targets. The MDIoU loss function is introduced as the algorithm's bounding box loss function to optimize the convergence speed of network training and improve the detection accuracy of dense occluded brown mushroom individuals. The improved algorithm is trained and tested on a self-built industrialized brown mushroom dataset. Compared to the original YOLOv7, the model's detection speed has increased by 15.5 %, detection accuracy has increased by 6.4 %, and average precision mAP@0.5 has increased by 6.9 %.

Keywords: Brown mushroom, dense detection, YOLOv7, AFPN, ELAN_PS, MPDIoU

1 INTRODUCTION

The nutritional content of brown mushrooms is diverse and extremely high, making them of excellent culinary value. Benefiting from the rapid development of China's mushroom industry since 2002, the country has now emerged as a substantial producer in the cultivation of mushrooms, commanding approximately 87 % of the global total output [1]. It can be seen that China's edible mushroom industry has tremendous potential, and the cultivation scale of edible mushrooms, especially brown mushrooms, is expected to expand rapidly. In the industrialized production process of brown mushrooms, harvesting is one of the most crucial steps. Currently, most enterprises in China still rely on manual harvesting as the primary method. During the harvesting process, workers need to observe and climb elevated structures day and night to pick brown mushrooms. This method involves high labour intensity and low harvesting efficiency. The growth environment of brown mushrooms differs from that of other mushrooms. Brown mushrooms are grown densely and overlap, with some tilting or falling, and mycelia forming large entanglements. Therefore, it is timely to develop an algorithm capable of precise and rapid identification of brown mushrooms in such a complex environment to improve harvesting efficiency.

Currently, deep learning-based object detection algorithms exhibit outstanding detection performance, mainly divided into two categories: two-stage detection algorithms based on candidate regions and one-stage detection algorithms based on regression. The representative two-stage detection algorithms include Fast R-CNN [2], Mask R-CNN [3], and Faster R-CNN [4], all built upon the basic idea of the R-CNN detection algorithm proposed by Girshick et al. in 2014 [5]. Although two-stage detection algorithms perform well in detection accuracy, they are slow in detection speed, have large model sizes, and are not suitable for real-time detection tasks such as mushroom harvesting. One-stage object detection algorithms include SSD [6], RetinaNet [7], and the YOLO series. The main characteristic of these algorithms is that the neural network skips the process of independently and explicitly extracting candidate regions when processing input images, directly in-

putting the image and detecting the category and position information of the target objects within it. Therefore, one-stage detection algorithms have much faster detection speeds than two-stage detection algorithms and are more suitable for tasks with real-time requirements. Wibowo et al. [8] segmented and detected mushrooms, extracting semantic features from their surfaces. They employed Support Vector Machines (SVMs) as the classification algorithm to classify edible mushrooms.

A lightweight mushroom detection model based on YOLOv3 is proposed in [9]. They introduced a neck network called shuffle Adaptive Spatial Feature Pyramid Network (ASA-FPN) and constructed a lightweight GhostNet16 to replace DarkNet53 as the backbone network, effectively improving the detection accuracy of the model. A refined YOLOv5s algorithm was introduced to enhance the precision of Agaricus bisporus detection in [10]. By incorporating the Convolutional Block Attention Module (CBAM) into the backbone network of YOLOv5s and implementing the Mosaic image augmentation technique during training, the recognition accuracy significantly increased to 98 %.

A modified version of YOLOv5, Recursive-YOLOv5, has been proposed in [11]. The enhancements include recursion, remerging the first output with the backbone network's convolutional layer, replacing SPP with atrous spatial pyramid pooling (ASPP), using complete intersection over union (CIoU) and distance IoU (DIoU) instead of generalized IoU (GIoU), and employing DIoU_non-maximum suppression (NMS) instead of the original NMS algorithm. The proposed network achieves a 98 % accuracy in identifying edible mushrooms in large-resolution, small-target scenarios, marking a 12.87 % improvement over YOLOv5x. To precisely detect small impurities in walnut kernels, [12] devised an enhanced impurity detection model based on the YOLOv5 network. Initially, a small target detection layer was incorporated in the neck section to enhance the model's capability in identifying small impurities, such as broken shells. Subsequently, the Transformer-Encoder (Trans-E) module replaced some convolution blocks to better capture global image information. The addition of the Convolutional Block Attention Module (CBAM) heightened the model's sensitivity to channel features, facilitating the identification of prediction regions in dense objects. Finally, the GhostNet module was introduced to streamline the model and enhance detection rates. Test results reveal that the mean average precision (mAP) of the improved YOLOv5 model achieved 88.9 %, marking a 6.7 % improvement over the original YOLOv5 network's average accuracy. An automated dragon fruit picking system [13] was proposed to introduce a detection method using RDE-YOLOv7 for enhanced accuracy in dragon fruit identification and localization. RepGhost and decoupled head are integrated into YOLOv7 to improve feature extraction and result prediction. Moreover, multiple Efficient Channel Attention (ECA) blocks are strategically placed in the network to extract valuable information efficiently. Experimental results demonstrate that RDE-YOLOv7 enhances precision, recall, and mean average precision by 5.0 %, 2.1 %, and 1.6 %, respectively.

At present, deep learning-based object detection algorithms have made significant progress in the field of agricultural detection. However, in the industrialized

cultivation environment of edible mushrooms, the application of two-dimensional machine vision faces challenges such as the complex posture of mushrooms, complex imaging environments, and high requirements for camera pixels. These issues result in images captured in two dimensions being easily affected by the environment, with unclear image features, leading to less-than-ideal practical detection outcomes. This limitation is observed in tasks such as detecting and controlling mushroom growth, automated harvesting, and quality grading [14]. Brown mushroom cultivation substrates, belonging to the category of edible mushrooms, are mostly uneven, covered with mycelium, and characterized by the predominant growth of dense clusters of brown mushrooms. The high overlap and unclear adhesion at the edges of brown mushrooms further complicate the detection process. To address these challenges, this paper proposes an improved YOLOv7 algorithm for brown mushroom detection, incorporating the following key enhancements:

1. Utilizing the efficient and lightweight ELAN_PS module to enhance the feature extraction capabilities of the backbone network.
2. Adopting the Progressive Feature Pyramid Network (AFPN) to replace the original SPPCSPC module, improving the model's ability to distinguish dense targets.
3. Enhancing the bounding box loss function of the original model to reduce false negatives and optimize model convergence speed.

2 IMPROVED YOLOV7 ALGORITHM

This paper introduces the ELAN-PS structure into the Backbone network section of YOLOv7 [15]. This inclusion aims to reduce the computational complexity and redundant features of the model, optimizing the information pathway. Subsequently, the AFPN module replaces the original SPPCSP module, reducing significant semantic gaps between non-adjacent feature maps. This modification significantly mitigates the loss or degradation of feature information during the transmission process, enhancing the model's capability to delineate densely clustered targets. Finally, the MPDIoU is introduced as the bounding box loss function for the model, effectively improving the model's convergence speed and detection accuracy. The modified network structure is illustrated in Figure 1.

3 INSERTING SPECIALS

3.1 Efficient and Lightweight ELAN-PS Structure

The Efficient Layer Aggregation Network (ELAN) is an efficient layer aggregation network composed primarily of multiple CBS modules stacked in a hierarchical manner. Each CBS module consists of standard Convolution (Conv), Batch Normalization (BN), and the Sigmoid Weighted Linear Unit [16] (SiLU) activation function.

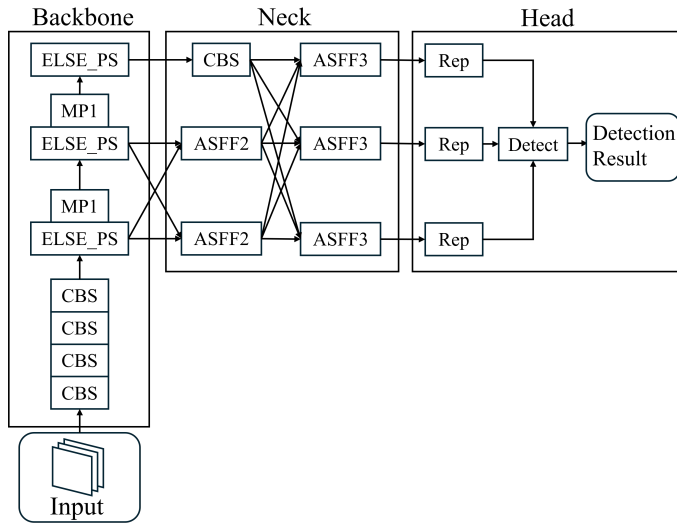

Figure 1. Improved structure of YOLOv7

Leveraging the residual structure concept to increase network depth, ELAN optimizes the gradient length of the entire network through the stacked structure within each computational block, facilitating rich gradient flow information. The ELAN network exhibits robust feature extraction capabilities when processing input data, enabling the network to achieve higher detection accuracy without introducing additional complex architectures. However, due to the presence of multiple layers of standard convolutions in the ELAN module, it unavoidably introduces feature redundancy and computational costs, thereby impacting the model's processing speed and performance.

Considering the demand for both detection accuracy and speed in the model when dealing with densely clustered mushroom groups in complex environments, this design introduces a low-cost pointwise convolution operator, Partial Convolution (PConv), and a spatial and channel reconstruction convolution, Spatial and Channel Reconstruction Convolution (SCConv), to propose the efficient lightweight ELAN_PS module. The specific structure of the ELAN_PS module is illustrated in Figure 2.

In the pursuit of designing fast neural networks, much attention has been devoted to improving the efficiency of floating-point operations per second (FLOPS). However, the low FLOPS in model computations are primarily attributed to frequent memory access by operators, especially in deep convolutional models with multi-level feature extraction. In 2023, Chen et al. [17] introduced the pointwise convolution operator, PConv, for the first time in their work Fasternet. By exploiting the highly similar feature redundancy between channels in feature maps, PConv

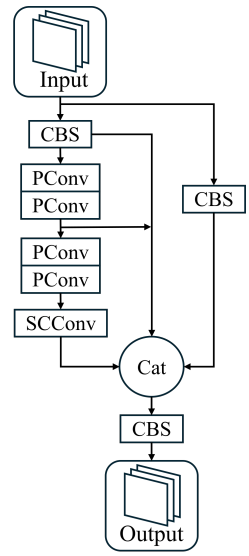

Figure 2. Structure diagram of ELAN_PS module

optimizes costs further, effectively reducing redundant computations and memory access. This innovation enhances both the detection speed and accuracy of the model.

Compared with conventional convolution, the PConv convolution operator is different in the calculation method of feature extraction, as shown in Figures 3 and 4. For the convolutional layer, there are m filters, and the convolutional kernel size is $k * k * c$, then the FLOPs of the regular convolution is $h * w * k^2 * l$, where h and w are the width and height of the feature map, and l is the number of channels for conventional convolution. In the actual implementation, the ratio of the number of channels c to l of the PConv operator is as follows: $r = c/l = 1/4$. Therefore, the FLOPs of the PConv operator are only 1/16 of the regular convolution.

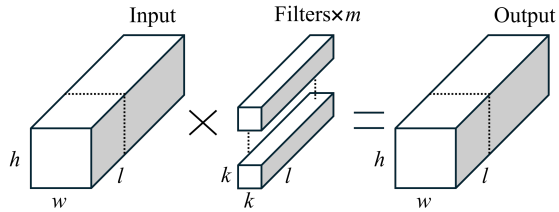

Figure 3. Conventional convolution diagram

Correspondingly, the memory access count for regular convolution is expressed

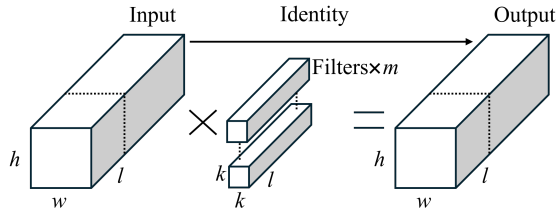

Figure 4. PConv convolution operator diagram

by Equation (1), as $l = 4c$, the memory access count for the PConv operator is only 1/4 of that for regular convolution.

$$h * w * 2l + k^2 * l^2 \approx h * w * 2l. \tag{1}$$

Therefore, introducing PConv to replace the CBS module effectively reduces the computational and parameter burden of the model, leading to improved detection speed. Additionally, PConv leverages the capabilities of computing devices more efficiently and demonstrates good performance in simultaneously extracting spatial features [18]. Although PConv has, to a certain extent, reduced the computational load of the model, in deep learning networks, the inevitable multi-level nested feature extraction results in the loss of certain features in both spatial and channel directions, thereby weakening the model's ability to represent features.

In 2023, Li et al. [19] proposed a plug-and-play SCConv convolution module, comprising two units: one named Spatial Reconstruction Unit (SRU) and the other named Channel Reconstruction Unit (CRU). Although the primary goal of the SCConv convolution module is to reduce the computational cost arising from redundant feature extraction in visual tasks, it enhances the model's ability to refine feature map recognition by performing feature reconstruction in both spatial and channel directions. The structure of SCConv is illustrated in Figure 5.

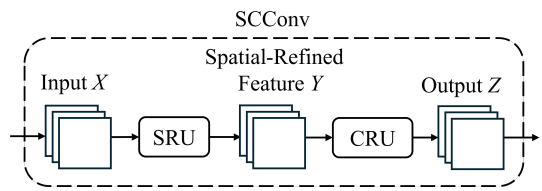

Figure 5. Structure of SCConv convolutional operator

In this convolutional operator, the feature X input by the previous convolution layer first passes through the spatial reconstruction unit SRU to obtain the spatially

refined feature Y. Then, through the channel reconstruction unit CRU, the channel extraction feature Z is obtained as the output. The SRU unit is mainly divided into two steps: feature separation and feature reconstruction. Firstly, the input information-rich feature map is separated from the feature map with less information corresponding to the spatial content, and then the information-rich features are added and reconstructed with the less information-rich features to generate more information-rich features.

The structure of the SRU is shown in Figure 6. For the feature map X of the Nth batch of input in the previous layer, the number of channels is M and the size is $H * W$.

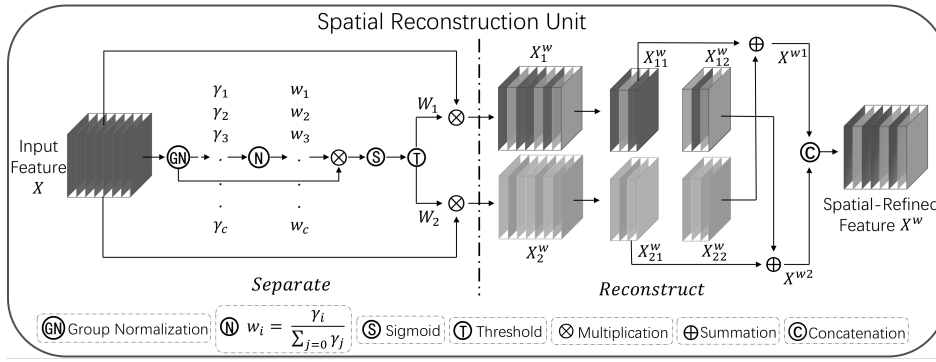

Figure 6. Structure diagram of the SRU

First, the SRU unit performs group normalization (GN) of X, normalizing the input feature X by subtracting the mean μ and dividing by the standard deviation δ, as shown in Equation (2).

$$X_{out} = F_{GN}(X) = \gamma \frac{X - \mu}{\sqrt{\delta^2 + \epsilon}}, \tag{2}$$

where ϵ is the minimum constant used to ensure stability, and the trainable parameter γ represents the richer spatial information in the feature map, so that the importance weight W_γ of multiple feature maps is obtained:

$$W_i = \frac{\gamma_i}{\Sigma_{j=0}\gamma_i}. \tag{3}$$

Set a threshold value, denoted as x, corresponding to the normalization of W_γ. Compare the values mapped to the (0,1) range with x. If the value is higher than x, assign an effective information weight $W_1 = 1$, otherwise, assign a redundant information weight $W_2 = 0$. Next, perform element-wise multiplication of W_1 and W_2 with the feature map X to obtain $X^{\omega 1}$ and $X^{\omega 2}$ respectively. Finally, through cross-reconstruction, concatenate the weighted, different information features along the channel to obtain the spatial refined feature map X^ω.

$$\begin{cases} X_1^{\omega} = W_1 * X, \\ X_2^{\omega} = W_2 * X, \\ X_{11}^{\omega} + X_{22}^{\omega} = X^{\omega 1}, \\ X_{12}^{\omega} + X_{21}^{\omega} = X^{\omega 2}, \\ X^{\omega 1} \cup X^{\omega 2} = X^{\omega}. \end{cases} \tag{4}$$

wherein, "x" represents element-wise multiplication, "+" denotes element-wise addition, and "∪" signifies the union operation. After obtaining the refined feature Y through the SRU unit, it is fed into the CRU unit for feature channel refinement. The structure of CRU is illustrated in Figure 7, employing a segmentation-transform-fusion approach to reduce channel redundancy. Y is divided into two parts with different channel numbers: one part with channels αC $(0 < \alpha < 1)$ and the remaining channels with $1 - \alpha C$. Subsequently, each part is separately input into two $1*1$ convolutional layers to obtain X_{up} and X_{low}. The input X_{up} undergoes GWC and PWC operations separately, followed by element-wise addition to generate enriched feature Y_1. The input X_{low} undergoes PWC calculation, and its union with itself yields the complementary feature Y_2.

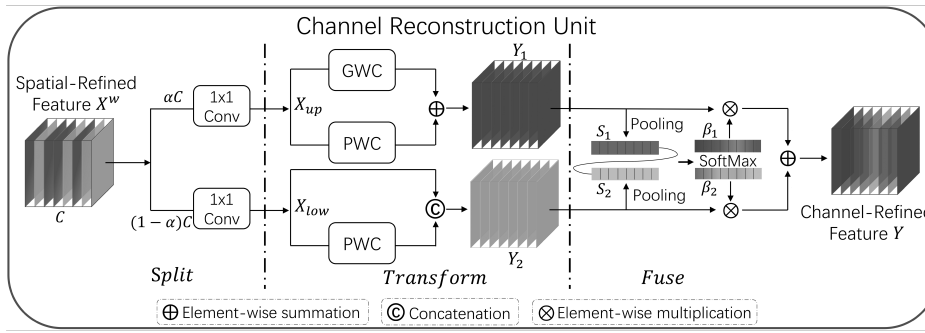

Figure 7. Structure diagram of the CRU

Finally, the adaptive merging of Y_1 and Y_2 is performed using the simplified SKNet [20] method. Global average pooling is applied to Y_1 and Y_2, resulting in S_1 and S_2. The purpose of this step is to integrate the global spatial information and channel statistical information of the feature maps. Subsequently, applying Softmax to S_1 and S_2 yields feature weight vectors φ_1 and φ_2. Based on these two feature weight vectors, the features Y_1 and Y_2 are combined to obtain the spatial-channel refined feature Y:

$$Y = \varphi_1 Y_1 + \varphi_2 Y_2. \tag{5}$$

The SCConv convolution effectively limits feature redundancy and enhances feature representation capabilities. Based on the above discussion, the improvement

of the original ELAN module mainly targets its computational resource requirements and feature extraction capabilities. The summary of the discussion is as follows.

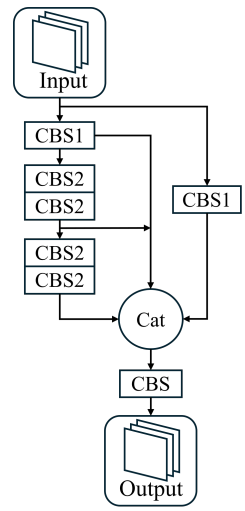

Figure 8. Structure of the original ELAN module

The original ELAN module structure is shown in Figure 8, which controls the shortest and longest gradient paths to enable the network to learn more features. The shortest path includes a 1×1 standard convolution, while the longest path consists of a 1×1 standard convolution followed by four 3×3 standard convolutions. Through calculations, it is found that the FLOPs of standard convolutions are 16 times that of PConv, indicating that with equivalent computational resources, using the PConv convolution operator can save more computational costs and improve algorithm execution speed.

Although PConv reduces the computational load to some extent, in deep learning networks, due to the inevitable hierarchical extraction of features, there is a loss of features in both spatial and channel directions, which may weaken the model's ability to represent features effectively.

The output feature maps of convolutional layers and input channels contain both high and low-frequency components. The low-frequency components support the overall contour of the target, while the high-frequency components focus more on detailed parts. SCConv convolutional operator, based on further reducing the model load, refines the extraction of features in the spatial and channel directions of the feature maps, emphasizing the high-frequency components of the feature maps. This allows the model to acquire discriminative feature representations, enhancing its ability to identify target edges.

3.2 Feature Fusion Module

After the improved bottom-up feature extraction from the main network, the resulting feature maps contain rich spatial and channel information. However, during this process, the resolution of the images gradually decreases, leading to the potential loss or degradation of details in relatively small or overlapping target regions. Yang et al. [21] proposed the Asymptotic Feature Pyramid Network (AFPN) in 2023 to facilitate direct interaction between features from non-adjacent layers, enabling the mutual supplementation of details between shallow (Low-Level) and deep (High-Level) features. AFPN incorporates the characteristics of HRNet [22], repeatedly fusing shallow and deep features to generate richer top-level features. Through a multi-stage, cross-level, progressively adaptive fusion method, AFPN merges Low-Level features to Top-Level features within the entire input pyramid network, avoiding the loss or degradation of detail information in the multi-level transmission.

AFPN extracts features at different levels from the backbone network. Initially, the two Low-Level feature maps are input into the feature pyramid network for fusion, followed by the gradual addition of High-Level and Top-Level feature maps. Subsequently, through the process of feature-adaptive spatial fusion, a set of corresponding multi-scale, comprehensive features is generated. Finally, this set of multi-scale features is input into the detection head for prediction. As shown in Figure 9, the feature maps input from the backbone network undergo two rounds of feature weight adaptation and feature extraction. The semantic information of different-level features becomes more closely aligned during the progressive fusion process, ultimately resulting in a set of comprehensive features $D = \{D_1, D_2, D_3\}$.

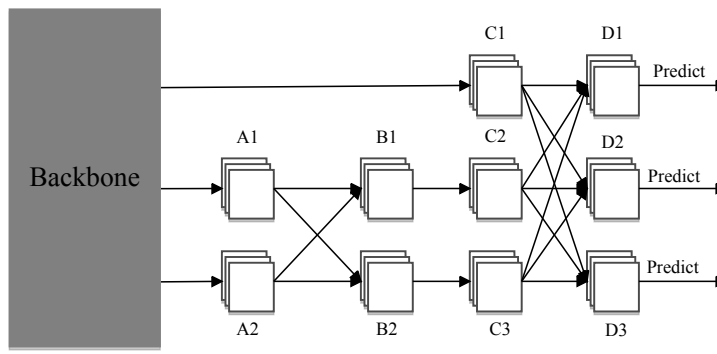

Figure 9. Architecture of AFPN

In order to highlight the proportion of important features of the key level, AFPN borrowed the adaptive geoscience Xi method of fusing the spatial weights of each scale feature map of the Adaptively Spatial Feature Fusion (ASFF) network, and

assigned different spatial weights to the features at different levels, so as to enhance the importance of the key level features and reduce the influence of contradictory information from different targets. The specific structure is shown in Figure 10.

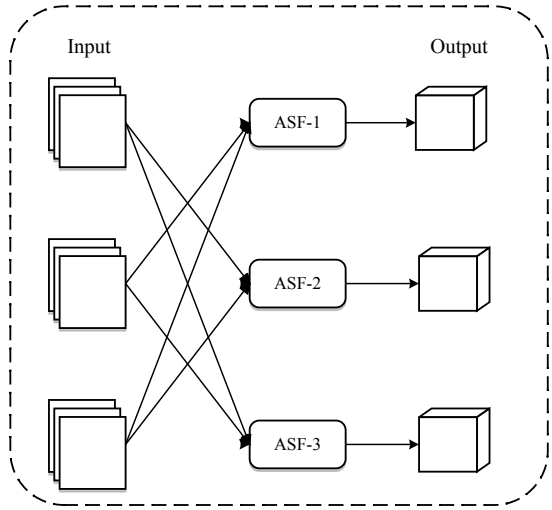

Figure 10. Schematic diagram of adaptive weight allocation

The vectors at the spatial position (i, j) of the resulting feature map after multi-level feature map fusion are the weighted fusion of the vectors at the first three feature maps (i, j) of the fusion, the spatial importance weights of feature maps are adaptively learned by the network, and they are shared across all channel spaces. Let $x_{ij}^{n\to l}$ be represented as the fusion eigenvector at the position (i, j) from feature map n to feature map l, and obtain the resulting eigenvector y_{ij}^{l} through the adaptive spatial fusion of multi-level features, which is defined by the linear combination of eigenvectors $x_{ij}^{1\to l}$, $x_{ij}^{2\to l}$ and $x_{ij}^{3\to l}$ as follows:

$$\begin{cases} y_{ij}^{l} = \alpha_{ij}^{l} x_{ij}^{1\to l} + \beta_{ij}^{l} x_{ij}^{2\to l} + \gamma_{ij}^{l} x_{ij}^{3\to l}, \\ \alpha_{ij}^{l} + \beta_{ij}^{l} + \gamma_{ij}^{l} = 1, \end{cases} \tag{6}$$

where α_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} represent the spatial weights α_{ij}^{l}, β_{ij}^{l}, $\gamma_{ij}^{l} \in [0, 1]$ of each feature n in the resulting feature l.

The dense and overlapping clusters of brown mushrooms, as the primary cultivation method for industrialized mushroom farming, pose significant challenges to the detection capabilities of the model. Additionally, the mushrooms share similar colors, and the overlapping edges are not easily discernible, making it difficult for the original YOLOv7 algorithm to effectively detect clusters dominated by mushrooms. In this paper, we propose replacing the SPPCSPC module of the original YOLOv7 with the AFPN network. This modification facilitates adaptive fusion of

non-adjacent hierarchical feature maps, significantly reducing the substantial semantic gaps between them. This adaptation effectively prevents the loss or degradation of crucial feature information, such as the edges and roots of brown mushrooms, during the transmission process. The proposed approach enhances the algorithm's ability to learn features of densely packed brown mushrooms, thereby improving detection accuracy and generalization capabilities.

3.3 MPDIoU Loss Function

Currently, most high-performance object detection algorithms rely on the bounding box regression (BBR) loss function to determine the specific location of detected targets. However, the majority of existing BBR loss functions assign the same values when predicting boxes with the same aspect ratio but vastly different width and height values, leading to a reduction in the convergence speed and accuracy of bounding box regression.

MPDIoU (Minimum Point Distance IoU) [23] is a novel bounding box regression loss function based on the minimum point distance. It directly minimizes the point distances between the predicted bounding box's top-left and bottom-right corners and the corresponding corners of the annotated bounding box to determine the target's position.

The implementation process of MPDIoU is as follows: For an input image of size $w * h$, let (x_1^A, y_1^A) and (x_2^A, y_2^A) are the coordinates of the top-left and bottom-right corners of predicted box A, (x_1^B, y_1^B) and (x_2^B, y_2^B) are the coordinates of the top-left and bottom-right corners of the annotated box B. Firstly, calculate the point distances for the top-left and bottom-right corners of these two rectangles as follows:

$$\begin{cases} d_1^2 = (x_1^A - x_1^B)^2 + (y_1^A - y_1^B)^2, \\ d_2^2 = (x_2^A - x_2^B)^2 + (y_2^A - y_2^B)^2. \end{cases} \tag{7}$$

Then, the minimum point distance intersection union ratio between them is calculated as:

$$\text{MPDIoU} = \text{IoU} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2}, \tag{8}$$

where IoU is the intersection and union size of two rectangular boxes, which is usually used as the degree of overlap between the prediction box and the actual labeling frame. In the model training stage, by minimizing the value of the MPDIoU loss function, each bounding box $B_{prd} = \left[x^{prd}, y^{prd}, w^{prd}, h^{prd}\right]$ predicted by the model tends to be close to the actual bounding box $B_{gt} = [x^{gt}, y^{gt}, w^{gt}, h^{gt}]$ of the target. The loss function expression is as follows:

$$L_{\text{MPDIodu}} = 1 - \text{MPDIoU}. \tag{9}$$

Compared to most IoUs, MPDIoU simplifies the similarity comparison between two bounding boxes, and is more prominent in tasks where multiple bounding boxes

overlap or non-overlap. Therefore, in this paper, MPDIoU is used to replace the original CIoU [24], which can not only improve the accuracy of the model in the detection of multi-overlapping brown mushroom populations, but also improve the robustness and training speed of the model.

4 EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Research Subjects

In the industrial environment of brown mushroom cultivation, a vision module is required to accurately identify individual brown mushrooms to assist the picking robot in precise harvesting. However, brown mushrooms in factory cultivation grow primarily in dense clusters. The mushrooms planted at the same time grow densely and overlap, with unclear edges. The growth medium is disrupted during the growth process, causing some mushrooms to tilt or even collapse. The growth patterns of brown mushrooms are illustrated in Figure 11. Situation A shows overlapping and tilted mushrooms; Situation B shows completely collapsed mushrooms, with the caps not visible; Situation C shows unclear mushroom edges under strong lighting.

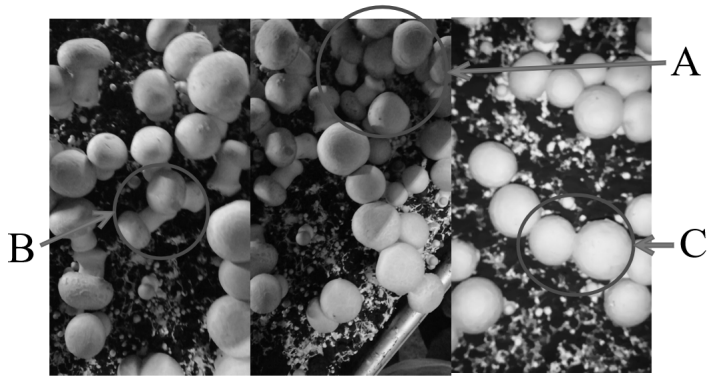

Figure 11. Schematic diagram of the growth mode of brown mushroom

The dense growth of brown mushroom clusters is typically classified into high density and moderate density levels. As shown in Figure 12, clusters with high density exhibit various growth postures of brown mushrooms, such as tilted growth, overlapping caps of multiple mushrooms, individual mushrooms obstructing others, and even some mushrooms collapsing. On the other hand, clusters with moderate density, as depicted in Figure 13, show minor overlapping and are relatively easier to detect.

To provide an effective and precise brown mushroom recognition module for the picking robot, addressing challenges such as various growth postures and difficulties

Figure 12. High-density clusters of brown mushrooms

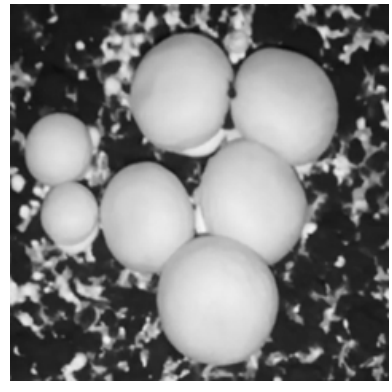

Figure 13. Moderate-density clusters of brown mushrooms

in shape recognition of brown mushrooms, this paper proposes improvements to the YOLOv7 algorithm. These enhancements aim to increase individual mushroom recognition and reduce the false detection rates within brown mushroom clusters.

4.2 Collection and Production of Datasets

This study focuses on brown mushrooms cultivated in a plantation facility. Multiple videos capturing brown mushroom clusters at different growth stages within the plantation base serve as the data source. Images are extracted at regular intervals to form the dataset. To address detection challenges such as inclined and ground-hugging mushrooms, two shooting angles are utilized: one perpendicular to the cultivation layers and another at a 30-degree angle relative to the mushroom beds.

The research primarily targets the difficulty in detecting densely packed brown mushrooms. Consequently, most videos collected feature highly dense clusters, with very few depicting moderately dense clusters. High-density clusters include intact individual or paired mushrooms exhibiting normal, inclined, collapsed, or adherent growth. Moderate-density clusters mainly feature slight overlaps or tilting, serving as balanced image features for model training.

Through video recording and photography, a total of 1 321 images of brown mushrooms were captured. Among these, 998 images were taken perpendicularly and 322 images were taken at an angle. Additionally, using a Python web crawler program, 386 images of brown mushroom clusters were obtained and manually filtered. These included approximately 132 images of moderately dense mushroom clusters and 254 images of highly dense mushroom clusters. In total, 1 707 images depicting brown mushroom cluster growth were collected through these methods. All images were annotated using the LabelImg tool, with the label "mushroom". An example of the annotation interface is shown in Figure 14.

Figure 14. Data labeling page

To enhance the robustness of the detection model, 1 707 images were processed using five methods: sharpness adjustment, image flipping, contrast adjustment, image blurring, as illustrated in the figure. This processing resulted in a total of 8 535 brown mushroom images, which served as the experimental dataset for this study.

4.3 Training Platform and Parameter Settings

The model training and testing for this experiment were conducted on a training platform set up in the laboratory. The experimental training platform runs on the Ubuntu 18.04 operating system, equipped with a GeForce RTX 4080 GPU, an Intel(R) Core(TM) i7-13700 CPU, and utilizes the PyTorch deep learning framework. The CUDA version used is 12.2, with Python 3.9 as the development software and Visual Studio Code 2019 as the development compiler.

The model training parameters were set as follows: 200 epochs, an initial learning rate of 0.01, input image size (imgsz) of 640×640 pixels, a batch size of 16 for each training iteration, and the model training adopted a method without pre-initialized weights.

4.4 Evaluation Indicators

In order to accurately evaluate the improved model, this paper uses the highest recall rate in the field of object detection algorithms, Recall (R), precision (P), number of images detected per second, FPS, and Mean Average Precision (mAP) when the IoU value is 0.5 as the evaluation indexes of the experiment in this paper. The definition of each evaluation indicator is as follows:

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{10}$$

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{11}$$

$$\mathrm{mAP} = \frac{1}{N} \sum_{1}^{N} \int_{0}^{1} P(R) d(R), \tag{12}$$

where, true positives (TP) represent the total number of brown mushroom targets correctly identified by the model. False Negatives (FN) indicates the total number of brown mushroom targets that were not recognized by the model, that is, the model missed the detection of brown mushrooms, False Positives (FP) represent the count of non-brown mushroom objects incorrectly identified as brown mushrooms by the model, indicating the quantity of false alarms. N represents the number of categories of targets identified by the model, and only brown mushroom individuals are identified in this paper, so N is equal to 1.

4.5 Improved Analysis of the ELAN Module

In the ELAN module, new convolutional operators can be embedded and replaced at multiple different positions. This experiment combines the convolutional characteristics of two types of operators. To mitigate feature loss during convolution and maintain control over the shortest and longest gradient paths, we retained the first conventional convolution and the shortest path conventional convolution in the original ELAN module, focusing improvements solely on the longest path conventional convolution. To validate this enhancement, PConv and SCConv were embedded or replaced at different positions (A, B, C, D) along the longest path within the ELAN module, forming various comparative configurations as detailed in Table 1. The specific embedding or replacement positions of PConv and SCConv within the ELAN module are illustrated in Figure 13.

Combination List		
Composite Number	The Position of SCConv	The Position of PConv
①	Placed into Position C	Replaces Position A
②	Placed into Position C	Replaces Position B
③	None	Replaces Positions A and B
④	Placed into Position D	Replaces Position A and B

Table 1. Experimental protocol

The models trained with the above configurations utilized a proprietary dataset from Section 4.2, comprising 8535 images of brown mushrooms, divided into training, testing, and validation sets in a ratio of 7 : 2 : 1. Evaluation metrics included recall (R), precision (P), and mAP@0.5. No initial weights were used in model training. Experimental results are presented in Table 2.

Composite Number	Recall	Precision	mAP@0.5
①	0.861	0.923	0.896
②	0.880	0.913	0.887
③	0.913	0.922	0.908
④	0.904	0.958	0.935

Table 2. Experimental results

Schemes ① and ② exhibit overall lower performance metrics compared to schemes ③ and ④. A possible reason is that after replacing the convolution with SCConv at position C, the network prematurely learns higher-frequency components of the feature map, leading to the early exclusion of low-level features. Consequently, subsequent conventional convolutions capture more detailed information but overlook the overall context, resulting in poorer recall and average precision. When SCConv convolution refines the extraction of spatial and channel features in the feature map, it focuses more on high-frequency components. By integrating these with the shortest path feature maps, it produces more discriminative feature maps. Therefore, although the recall rate of scheme ③ is higher than that of scheme ④, its precision and mAP@0.5 are 3.6 % and 2.7 % lower, respectively.

Given the relatively small performance difference between the models, selecting the scheme with a higher mAP@0.5 value is more cost-effective for brown mushroom detection tasks. Therefore, the improvements in scheme ④ are more advantageous.

4.6 Ablation Experiments

To validate the rationality of the proposed model improvements, ablation experiments were conducted by comparing against the baseline model. These experiments involved gradually adding or replacing the ELAN_PS module, AFPN module, and

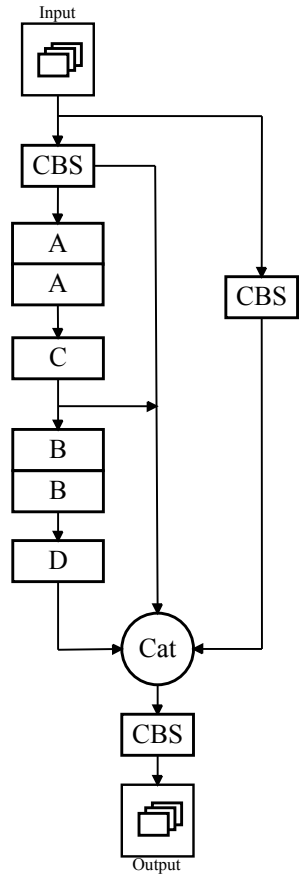

Figure 15. Diagram of ELAN_PS position structure

MPDIoU loss function within the original YOLOv7 model. Performance metrics of each model were then compared against the proposed enhancements to complete the ablation experiments. The experiments utilized the same dataset, experimental environment, and model parameters as described in Section 4.3.

From Table 3, it can be observed that replacing the bounding box regression loss function in the original YOLOv7 model with MPDIoU results in a 1.9 % increase in detection precision, albeit with a slight decrease in recall. Similarly, substituting the AFPN module for the original SPPCSPC module enhances the model's detection accuracy by 1.3 %, but there is a loss in both recall and mean average precision. Inserting the ELAN_PS module into the backbone network to replace the original ELAN module leads to a 3.5 % improvement in the model's detection accuracy, though with a slight sacrifice in detection speed. Simultaneously incorporating the AFPN module and MPDIoU loss function into the original YOLOv7 model results

Model	Recall	Precision	mAP@0.5	FPS
v7	0.929	0.923	0.905	84
v7 + MPD	0.916	0.942	0.896	91
v7 + AFPN	0.897	0.936	0.911	88
v7 + ELAN_PS	0.921	0.958	0.935	94
v7 + AFPN + MPD	0.939	0.954	0.914	110
Improvement	0.988	0.987	0.974	97

Table 3. Results of ablation experiments

in slight improvements in recall, precision, and other metrics, with a 3.1 % increase in detection precision and a 30.9 % boost in detection speed. Combining these three improvement approaches in the proposed enhancement on the original YOLOv7 yields a 6.4 % increase in precision and a 15.5 % improvement in detection speed, with noticeable improvements in recall and mean average precision. The ablation experiments demonstrate the effectiveness of the proposed improvements in enhancing the performance of the YOLOv7 model in the dense brown mushroom detection task.

4.7 Comparative Experiments of Different Object Detection Algorithms

To further analyze and demonstrate the effectiveness and superiority of the proposed improved model in dense brown mushroom detection, this study, based on the self-made dataset in Section 4.2, conducted comparative experiments between the improved algorithm and the original YOLOv7 algorithm, SSD, YOLOv3 [25], YOLOv4 [26], YOLOv5, and Faster R-CNN algorithms. The evaluation metrics for this section's experiments are the model's mean average precision (mAP@0.5) and frames per second (FPS) for detection. The experimental results are presented in Table 4.

Model	mAP@0.5	FPS
YOLOv7	0.905	84
SSD	0.831	47
YOLOv3	0.806	39
YOLOv4	0.858	68
YOLOv5	0.896	86
Faster R-CNN	0.953	22
YOLOv5 + MPD	0.824	76
YOLOv5 + AFPN	0.866	84
v5 + MPD + AFPN	0.875	89
Improvement	0.974	97

Table 4. Performance comparison of different object detection algorithms

From Table 4, it is evident that the proposed improved model outperforms other object detection algorithms in both mAP@0.5 and FPS. Specifically, the improved

model achieves an average precision at an IoU threshold of 0.5 that is 6.9 %, 14.3 %, 16.8 %, 11.6 %, 7.8 %, and 2.1 % higher than the original YOLOv7 algorithm, SSD, YOLOv3, YOLOv4, YOLOv5, and Faster R-CNN, respectively. This indicates that the improved model indeed exhibits superior performance compared to other commonly used models. On the other hand, the detection speed of the model directly impacts its real-time capabilities and usability. In comparison to the mainstream algorithm models with the highest FPS values in Table 2, YOLOv5, and the original YOLOv7, the proposed improved model achieves a 12.8 % higher FPS than YOLOv5 and a 15.5 % higher FPS than the original YOLOv7. In summary, the proposed improved model not only meets the requirements for real-time detection but also achieves a detection accuracy as high as 97.4 %. Additionally, the model's performance surpasses that of other object detection algorithms, showcasing outstanding overall performance and effectively demonstrating the superiority of the proposed improvement.

5 CONCLUSIONS

The growth of brown mushrooms in a factory planting environment is complex, with mushrooms densely and overlappingly planted during the same period. Due to the disruption of the cultivation soil during the growth process, some mushrooms tilt or even fall down. The visual detection module operates top-down, encountering challenges such as unclear mushroom edges and high overlap rates. This results in low recognition accuracy and high false detection rates in current mainstream object detection models. Addressing these issues, this paper uses YOLOv7 as the baseline model and introduces the ELAN_PS module into the original model's backbone network. Additionally, the improved model incorporates a modified bounding box regression loss function and introduces the AFPN module into the original model's neck network. Experimental results demonstrate that the improved algorithm enhances the precision of brown mushroom recognition on the mushroom cultivation substrate by 6.4 %. Meanwhile, the model's FPS remains at 97, ensuring good real-time performance. The improved model exhibits high accuracy and good performance, making it more suitable for real-time individual brown mushroom detection compared to other object detection algorithms. However, in environments with low brightness and poor visibility, these external factors can affect the recognition performance of the proposed algorithm, reducing detection accuracy.

Therefore, there is considerable room for improvement in this algorithm. In future work, more research can be performed on the recognition of brown mushrooms or fungi in adverse environments. This may involve quantifying adverse environmental factors and incorporating them into the detection model for training to enhance the model's resistance to environmental challenges.

REFERENCES

[1] Royse, D. J.—Baars, J.—Tan, Q.: Current Overview of Mushroom Production in the World. In: Cunha Zied, D., Pardo-Giménez, A. (Eds.): Edible and Medicinal Mushrooms: Technology and Applications. Wiley Online Library, 2017, pp. 5–13, doi: 10.1002/9781119149446.ch2.

[2] Girshick, R.: Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[3] He, K.—Gkioxari, G.—Dollár, P.—Girshick, R.: Mask R-CNN. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 42, 2020, No. 2, pp. 386–397, doi: 10.1109/TPAMI.2018.2844175.

[4] 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, Vol. 39, 2017, No. 6, pp. 1137–1149, doi: 10.1109/TPAMI.2016.2577031.

[5] Girshick, R.—Donahue, J.—Darrell, T.—Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[6] Liu, W.—Anguelov, D.—Erhan, D.—Szegedy, C.—Reed, S.—Fu, C. Y.—Berg, A. C.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[7] Lin, T. Y.—Goyal, P.—Girshick, R.—He, K.—Dollár, P.: Focal Loss for Dense Object Detection. 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[8] Wibowo, A.—Rahayu, Y.—Riyanto, A.—Hidayatulloh, T.: Classification Algorithm for Edible Mushroom Identification. 2018 International Conference on Information and Communications Technology (ICOIACT), IEEE, 2018, pp. 250–253, doi: 10.1109/ICOIACT.2018.8350746.

[9] Cong, P.—Feng, H.—Lv, K.—Zhou, J.—Li, S.: MYOLO: A Lightweight Fresh Shiitake Mushroom Detection Model Based on YOLOv3. Agriculture, Vol. 13, 2023, No. 2, Art. No. 392, doi: 10.3390/agriculture13020392.

[10] Chen, C.—Wang, F.—Cai, Y.—Yi, S.—Zhang, B.: An Improved YOLOv5s-Based Agaricus Bisporus Detection Algorithm. Agronomy, Vol. 13, 2023, No. 7, Art. No. 1871, doi: 10.3390/agronomy13071871.

[11] Wei, B.—Zhang, Y.—Pu, Y.—Sun, Y.—Zhang, S.—Lin, H.—Zeng, C.—Zhao, Y.—Wang, K.—Chen, Z.: Recursive-YOLOv5 Network for Edible Mushroom Detection in Scenes with Vertical Stick Placement. IEEE Access, Vol. 10, 2022, pp. 40093–40108, doi: 10.1109/ACCESS.2022.3165160.

[12] Yu, L.—Qian, M.—Chen, Q.—Sun, F.—Pan, J.: An Improved YOLOv5 Model: Application to Mixed Impurities Detection for Walnut Kernels. Foods, Vol. 12, 2023, No. 3, Art. No. 624, doi: 10.3390/foods12030624.

[13] Zhou, J.—Zhang, Y.—Wang, J.: RDE-Yolov7: An Improved Model Based on YOLOv7 for Better Performance in Detecting Dragon Fruits. Agronomy, Vol. 13, 2023, No. 4, Art. No. 1042, doi: 10.3390/agronomy13041042.

[14] Smith, L. N.—Zhang, W.—Hansen, M. F.—Hales, I. J.—Smith, M. L.: Innovative 3D and 2D Machine Vision Methods for Analysis of Plants and Crops in the Field. Computers in Industry, Vol. 97, 2018, pp. 122–131, doi: 10.1016/j.compind.2018.02.002.

[15] Wang, C. Y.—Bochkovskiy, A.—Liao, H. Y. M.: YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 7464–7475, doi: 10.1109/CVPR52729.2023.00721.

[16] Elfwing, S.—Uchibe, E.—Doya, K.: Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. Neural Networks, Vol. 107, 2018, pp. 3–11, doi: 10.1016/j.neunet.2017.12.012.

[17] Chen, J.—Kao, S. h.—He, H.—Zhuo, W.—Wen, S.—Lee, C. H.—Chan, S. H. G.: Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 12021–12031, doi: 10.1109/CVPR52729.2023.01157.

[18] Zhang, J.—Wei, X.—Zhang, L.—Yu, L.—Chen, Y.—Tu, M.: YOLO v7-ECA-PConv-NWD Detects Defective Insulators on Transmission Lines. Electronics, Vol. 12, 2023, No. 18, Art. No. 3969, doi: 10.3390/electronics12183969.

[19] Li, J.—Wen, Y.—He, L.: SCConv: Spatial and Channel Reconstruction Convolution for Feature Redundancy. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 6153–6162, doi: 10.1109/CVPR52729.2023.00596.

[20] Hu, J.—Shen, L.—Sun, G.: Squeeze-and-Excitation Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[21] Yang, G.—Lei, J.—Zhu, Z.—Cheng, S.—Feng, Z.—Liang, R.: AFPN: Asymptotic Feature Pyramid Network for Object Detection. 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2023, pp. 2184–2189, doi: 10.1109/SMC53992.2023.10394415.

[22] Sun, K.—Xiao, B.—Liu, D.—Wang, J.: Deep High-Resolution Representation Learning for Human Pose Estimation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5686–5696, doi: 10.1109/CVPR.2019.00584.

[23] Ma, S.—Xu, Y.: MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression. CoRR, 2023, doi: 10.48550/arXiv.2307.07662.

[24] Zheng, Z.—Wang, P.—Ren, D.—Liu, W.—Ye, R.—Hu, Q.—Zuo, W.: Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. IEEE Transactions on Cybernetics, Vol. 52, 2022, No. 8, pp. 8574–8586, doi: 10.1109/TCYB.2021.3095305.

[25] Redmon, J.—Farhadi, A.: YOLOv3: An Incremental Improvement. CoRR, 2018, doi: 10.48550/arXiv.1804.02767.

[26] Bochkovskiy, A.—Wang, C. Y.—Liao, H. Y. M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. CoRR, 2020, doi: 10.48550/arXiv.2004.10934.

Shanheng Tan obtained his Bachelor's degree in automation from the Beihua University, China in 2021 and his Master's degree in computer technology from the Qingdao University of Science and Technology, China in 2024. He is currently a Teaching Assistant at the Hunan Applied Technology University, China. His research directions are deep learning and the application research of intelligent robots in agriculture.

Shengjie Ma received her Ph.D. degree in computer engineering at the Kwangwoon University, Seoul, South Korea. She is currently a Lecturer at the School of Information Science and Technology, Qingdao University of Science and Technology. Her research interests include deep learning, vehicular positioning and path loss modeling, embedded systems and intelligent robots.

Xinglu Ma is Professor in the School of Information Science and Technology at Qingdao University of Science and Technology and Supervisor for postgraduate students. He is currently the Director of the Software Engineering Teaching and Research Section and the Director of the Intelligent Hardware and Robot Research Laboratory. His research directions include intelligent hardware, embedded AI, educational robots, and so on.

Weian Dong received his Bachelor's degree in mechanical engineering from the Zhengzhou University of Aeronautics in 2022 and is pursuing the Master's degree in mechanical engineering from the Department of Mechanical Engineering at the Hohai University. He is currently researching robotic vision and somatosensory interaction at the Hohai University. His research interests include robot vision, deep learning, and human-computer interaction.

Yangbo Sheng received his Bachelor's degree in mechanical engineering from the Quzhou University in 2023 and is pursuing the Master's degree in mechanical engineering at the Department of Mechanical Engineering at the Hohai University. He is currently researching robotic vision and control at the Hohai University. His research interests include robot vision, deep learning, and control systems.

Chunxu Li received his Ph.D. degree from Swansea University in 2019 and was appointed as Lecturer at the School of Engineering, Computing and Mathematics, University of Plymouth in January 2020. He has published over 40 academic papers, 33 of which are SCI/EI indexed. He won the Best Student Paper Award and was shortlisted for the Best Paper Award at IEEE international conferences. He has expertise in ROS, JAVA, C++, Python and MATLAB with over 8 years of working experience on multiple robot platforms, e.g., KUKA iiwa, Baxter, NAO, Universal, etc. He is an Associate Fellow of the Higher Education Academy and a member of IEEE (Institute of Electrical and Electronics Engineers). As the PI and the main co-I, he successfully granted several projects and funding.