Journal of Agricultural Engineering

https://www.agroengineering.org/

SCS-YOLO11: a robust detection framework for pileus of deer antler mushrooms in greenhouse environments

Shuzhen Yang,[1] Jiahong Du,[1] Dongjian Zhang,[1] Sangsang Li[2]

[1]School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University

[2]Shanghai Xinzheng Specialized Farmers' Cooperative, Shanghai, China

Corresponding author: Dongjian Zhang, School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai, China. E-mail: djzhang@sspu.edu.cn

Publisher's Disclaimer

E-publishing ahead of print is increasingly important for the rapid dissemination of science. The *Early Access* service lets users access peer-reviewed articles well before print/regular issue publication, significantly reducing the time it takes for critical findings to reach the research community.
These articles are searchable and citable by their DOI (Digital Object Identifier).

Our Journal is, therefore, e-publishing PDF files of an early version of manuscripts that undergone a regular peer review and have been accepted for publication, but have not been through the typesetting, pagination and proofreading processes, which may lead to differences between this version and the final one.
The final version of the manuscript will then appear on a regular issue of the journal.

Please cite this article as doi: 10.4081/jae.2025.1930

Submitted: 1 May 2025
Accepted: 3 November 2025

***Note**: The publisher is not responsible for the content or functionality of any supporting information supplied by the authors. Any queries should be directed to the corresponding author for the article.*

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article or claim that may be made by its manufacturer is not guaranteed or endorsed by the publisher.

SCS-YOLO11: a robust detection framework for pileus of deer antler mushrooms in greenhouse environments

Shuzhen Yang,[1] Jiahong Du,[1] Dongjian Zhang,[1] Sangsang Li[2]

[1]School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University

[2]Shanghai Xinzheng Specialized Farmers' Cooperative, Shanghai, China

Corresponding author: Dongjian Zhang, School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai, China. E-mail: djzhang@sspu.edu.cn

Authors' contributions: all authors made a substantive intellectual contribution, read and approved the final version of the manuscript and agreed to be accountable for all aspects of the work.

Conflict of interest: the authors declare no competing interests, and all authors confirm accuracy.

Availability of data and materials: the datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Funding: this work was supported by the Shanghai Municipal Science and Technology Commission under Grant [21N21900600] and Shanghai Agricultural Science and Technology Innovation Project, under Grant [I202303].

Abstract

In the intelligent cultivation of mushrooms within greenhouses, monitoring during the blooming period is crucial. This stage involves the formation and differentiation of young fruiting bodies, where timely detection of mushroom pileus is essential for automated environmental control. However, accurately detecting and counting immature caps remains challenging due to their small size, similar morphology, dense clustering, and complex background interference in greenhouse environments. To address these issues, this paper proposes an improved detection system, named SCS-YOLO11, based on the YOLOv11s architecture. To address the challenges of small-target detection in mushroom pileus recognition, we propose a coupled multi-scale attention (CMCA) module that effectively integrates global context and multi-scale spatial features. Additionally, a lightweight SPConv module is introduced to reduce computational cost while maintaining feature expressiveness, and a compact spatial-channel attention module (SCAM) further enhances feature discrimination in the detection head. It jointly models spatial and channel attention to guide the model to focus on key mushroom cap regions across multi-scale feature maps. Compared with the baseline YOLO11s model, SCS-YOLO11s shows remarkable improvements. Its precision increases from 79% to 84%, and mAP rises from 74.6% to 79%, with only 2.13M parameters and 3.6G FLOPs, demonstrating high efficiency. When applied to mushroom datasets, experiments show that its performance surpasses other YOLO-series models. SCS-YOLO11 strikes a balance between detection accuracy and computational efficiency, making it a promising solution for real-time monitoring of small mushroom pileus in the complex and dynamic settings of greenhouse mushroom cultivation.

Keywords: Mushroom pileus detection, small object detection, deep learning, YOLOv11, SCS-YOLO11.

Introduction

Lyophyllum decastes, commonly known as deer antler mushroom, has become one of the most important cultivated varieties of rare edible fungi in recent years (Chen *et al.*, 2024). Valued for its rich umami flavor, balanced texture (suitable for both dried and fresh consumption), and high nutritional content—including abundant proteins, essential amino acids, vitamins, and β-glucans (Li *et al.*, 2024)—this mushroom has gained increasing popularity in the commercial market. As demand grows, enhancing both yield and quality has become a key focus in its cultivation. Thus, it calls for in-depth research in multiple areas, including the selection and breeding of superior varieties and the development of high-quality cultivation techniques, through systematic phenotyping of *Lyophyllum decastes*.

Currently, the growth monitoring of *Lyophyllum decastes* primarily relies on manual inspection. This approach is labor-intensive, lacks real-time feedback, and is susceptible to subjective errors, thus falling short of the requirements for large-scale intelligent cultivation. Accurate detection and counting of pileus of deer antler mushrooms remain particularly challenging due to their tiny object scale, morphological similarity, dense distribution, and complex background interference commonly found in greenhouse environments. As a result, missed detections and false positives are frequently observed in automated systems. Despite the importance of this task, the detection of immature mushroom caps has received limited attention. One early study by (Gao *et al.*, 2014) proposed an improved HOG-based framework incorporating image pyramids and a sliding window strategy. Although this method achieved moderate improvements in accuracy, it still performs poorly in detecting tiny and densely distributed caps under practical conditions. These limitations demonstrate that traditional or shallow models are insufficient for handling the fine-grained and small-object characteristics of immature *Lyophyllum decastes*. Therefore, there is an urgent need for a more robust, lightweight, and target-specific deep learning framework that can enable accurate recognition in complex agricultural scenes.

Over the past decade, the rapid advancement of deep learning has driven the evolution of object detection algorithms capable of efficiently recognizing objects in images and videos, with strong generalization across various domains. These algorithms are broadly categorized into two types: region-based two-stage detectors and regression-based one-stage detectors. Two-stage approaches, such as R-CNN (Girshick, 2015), Faster R-CNN (Ren *et al.*, 2016), R-FCN (Dai *et al.*, 2016), and Mask R-CNN (He *et al.*, 2017), first generate region proposals and then perform classification and refinement, typically achieving high accuracy. In contrast, one-stage methods, including SSD (Liu *et al.*, 2016), RetinaNet (Lin *et al.*, 2017), and the widely adopted YOLO series (Redmon *et al.*, 2016; Redmon and Farhadi, 2017; Bochkovskiy *et al.*, 2020; Khanam and Hussain, 2024a; Luo *et al.*, 2025; Wang *et al.*, 2024), directly predict object classes and bounding boxes, offering faster inference speeds with relatively lower compuational overhead. Among existing detection algorithms, the YOLO series achieves an excellent trade-off between speed and accuracy by feeding normalized images directly into a convolutional neural network (CNN) for end-to-end detection. Its high efficiency, low false positive rate, and strong generalization capability have led to its widespread adoption in agricultural scenarios such as fruit counting, surface defect detection, and mushroom monitoring (Kiran *et al*., 2025; Huang *et al*., 2024; Chen *et al*., 2025). Building on these strengths, YOLO-based models have been extensively used for agricultural object detection tasks.

Several studies have focused on improving YOLO's performance for specific agricultural applications. For example, Chen *et al*. (2023) proposed YOLOv5s-CBAM, integrating the CBAM attention module and Mosaic data augmentation to improve detection accuracy and robustness in mushroom scenarios. However, the added complexity of the model may hinder real-time deployment in resource-constrained environments. Lu and Liaw (2020) developed a YOLOv3-based system with a scoring penalty algorithm to estimate the growth rate of Agaricus bisporus, but the reliance on hand-crafted scoring limits its precision and adaptability under occlusion or poor lighting. Zhao *et al*. (2023) introduced an improved YOLOv5s model incorporating an attention mechanism for detecting *Oudemansiella raphanipes*, achieving superior precision and mAP in complex environments. Yet, it lacks dedicated mechanisms for detecting extremely small or overlapping targets. Similarly, Shang *et al*. (2023) integrated K-means++ clustering into YOLOv5s to detect *Camellia oleifera* fruits hidden behind leaves, achieving an mAP of 94.1%, but with a model size of 27.1M parameters, unsuitable for embedded systems. Wang *et al*. (2023) presented a channel-pruned YOLOv5s model for detecting small apples, achieving 95.8% accuracy with a size of only 1.4M, but still struggled in heavily occluded or densely packed scenes.

YOLO11 (Khanam and Hussain, 2024b), developed by the Ultralytics team, represents the latest milestone in the YOLO family. It builds on the classic YOLO architecture with significant enhancements in backbone design, attention mechanisms, and detection strategies. YOLO11 performs exceptionally well in complex environments and small-object detection tasks. Researchers have further customized it for application-specific needs. For instance, Zhang *et al*. (2025) proposed YOLO11-Pear by refining the backbone and detection layers for improved pear detection in orchard environments. However, the model is highly specialized and lacks generalization capability. In another direction, Soudeep *et al*. (2024) introduced DGNN-YOLO, integrating dynamic graph neural networks with YOLOv11 to enhance the detection and tracking of small, occluded objects in urban traffic. Despite its superior performance, the high computational cost of graph construction and visualization limits real-time deployment.

Despite these advancements, challenges persist in detecting extremely small or occluded targets in complex agricultural scenarios. Specifically, the detection of *Lyophyllum decastes*, also known as the deer antler mushroom, during its early fruiting stage has received limited attention. During the early blooming stage, when the fruiting bodies transition from undifferentiated primordia ("buds") to initial cap formation ("pileus"), the mushroom caps are extremely small and often obscured by surrounding structures. This morphological phase presents significant challenges for visual detection, including low detection accuracy, increased false positives, and high computational costs in existing models. To address these challenges, this paper proposes a lightweight and efficient detection framework for small mushroom caps based on YOLOv11s, named SCS-YOLO11 (SPConv + CMCA + SCAM, abbreviated as SCS). The proposed model aims to achieve high detection accuracy and real-time performance in complex and dynamic greenhouse environments. The main contributions of this study are summarized as follows:

CMCA - context-aware multi-scale attention for small target detection: to compensate for semantic loss due to network simplification, the CMCA module is introduced in the neck. It fuses global contextual information from the CAFM branch with fine-grained multi-scale features from the MSCA branch, enhancing both detail representation and context awareness.

SPSConv - lightweight and efficient backbone module: the lightweight SPSConv module replaces standard convolutional layers in the backbone. This change reduces model complexity by 59.1% and inference latency by 9.1 ms compared to the baseline YOLOv11s, while maintaining effective feature extraction.

SCAM - spatial-channel attention in the detection head: the SCAM module is integrated into the detection head to jointly model spatial and channel-wise attention. It enhances the model's focus on key regions of mushroom caps across multi-scale feature maps, improving detection accuracy with minimal computational overhead.

Materials and Methods

Image acquisition and data pre-processing

The image data collection for the seafood mushrooms in this study was conducted in the standardized cultivation room of Shanghai Rongmei Agricultural Technology Co., Ltd. The image acquisition system consisted of a fixed imaging device, with data transmitted via Gigabit Industrial Ethernet to a Dell R750 server (total storage capacity: 8 TB). An industrial camera (equipped with a Sony IMX586 image sensor) was mounted on a stainless-steel work platform using an anti-vibration bracket. The object distance was maintained within the range of 500±50 mm, and a schematic diagram of the acquisition system is shown in Figure 1. The camera had a focal length of 0.85 mm and a resolution of 4000 × 3000 pixels. During the experiment, the camera captured growth images of the mushrooms every six hours, and the on-site acquisition environment is illustrated in Figure 1.

This paper constructs a fully automated image preprocessing pipeline. First, the ONNX inference engine deployed on the server automatically identifies the target regions of Lentinula edodes (deer antler mushroom) and extracts regions of interest (ROIs), employing non-maximum suppression (NMS = 0.45) to ensure single-mushroom localization accuracy within ±5 pixels. Subsequently, adaptive size normalization is applied, where the cropped sub-images are uniformly resized to 1280×1280 pixels using bilinear interpolation.

A quality assessment module then operates concurrently, automatically filtering out blurred frames based on the variance of the Laplacian operator in OpenCV (threshold >120), ultimately generating a standardized dataset. The image annotation tool Labelme is used to label bounding boxes on these preprocessed images. Annotation data is initially saved in JSON format and then converted to YOLO format using a custom Python script to facilitate efficient training with deep learning algorithms. Finally, the dataset is split into training, validation, and test sets in a 7:2:1 ratio. To enhance the model's generalization capability and reduce overfitting to specific features in the training set (e.g., lighting conditions, shooting angles, or cap morphologies), various data augmentation techniques are applied. Specifically, a set of randomized transformations -such as random rotation (10-25 degrees), flipping, Gaussian noise injection, and spatial shifting- are employed to simulate variations in growth posture, illumination, and perspective in real-world greenhouse environments. These operations effectively increase the diversity and robustness of the training data.

SCS-YOLO model for small pileus detection

This study proposes the SCS-YOLO system, an enhanced architecture based on YOLOv11s, designed to achieve higher detection accuracy and computational efficiency while mitigating false positives in mushroom cap detection. To enable lightweight deployment, an SPConv module is introduced as a replacement for the original convolutional blocks. This substitution preserves feature extraction capability while reducing computational complexity by 23%, as validated. To address the potential performance degradation caused by model simplification and to specifically improve the perception of small-scale mushroom caps, a coupled multi-scale contextual attention module is incorporated into the neck network.

This module facilitates the synergistic integration of global context modeling and local detail enhancement. Furthermore, a spatial-channel adaptive module (SCAM) is integrated into the detection head, enhancing region-of-interest perception through joint spatial and channel attention mechanisms without introducing significant computational overhead. The overall architecture of the proposed SCS-YOLO model is illustrated in Figure 2.

Multi-scale coupled attention module

Considering the limitation of convolution operations in capturing global context due to their inherently local receptive fields, effectively modeling long-range dependencies remains challenging. In contrast, transformers excel at extracting global features and handling long-range dependencies through their self-attention mechanisms. By integrating convolutional operations with attention mechanisms, both local details and global contextual information can be simultaneously and effectively modeled. Motivated by this synergy, the attention and convolutional module is introduced, as illustrated in Figure 3.

This module is meticulously designed to enhance the modeling of both global and local features, thereby facilitating the capture of long-term feature dependencies and spatial autocorrelation. Furthermore, to overcome the restricted single-scale feature aggregation inherent in feed-forward networks (FFNs) within transformer architectures, we integrate a multi-scale neural network (DCFN), delineated in Figure 4. This innovative architecture is specifically engineered to enhance multi-scale information aggregation by extracting features across a spectrum of scales, thus addressing the limitations of single-scale aggregation in FFNs. This method demonstrates proficiency in accurately detecting and precisely localizing various complex targets within images, fully showcasing its technical advantages.

Within CMCA, we integrate a self-attention mechanism into the global branch to capture a diverse range of global features. Meanwhile, the local branch augments model complexity via channel shuffling, thereby enhancing representational capacity and mitigating the risk of overfitting. The proposed Convolution and Attention Fusion Module (CAFM) comprises both global and local branches. In the global branch, an attention mechanism is introduced to enhance long-distance information interaction. The initial step involves facilitating the generation of query (Q), key (K), and value (V) tensors through the utilization of 1 ×1 convolution and 3×3 depth convolution operations. This process serves to effectuate a transformation upon the input tensor, thereby engendering three distinct tensors distinguished by dimensions denoted as H× W× C. The global branch can be formulated as:

$$\mathrm{f_{att}}=\mathrm{W_{1\times1}}\mathrm{Attention}\left(\hat{Q},\hat{K},\hat{V}\right)+\mathrm{Y} \qquad \text{(Eq. 1)}$$

$$Attention(\hat{Q},\hat{K},\hat{V})=\hat{V}Softmax(\hat{Q}\hat{K}/\alpha) \qquad \text{(Eq. 2)}$$

In the local branch, the multi-scale convolutional architecture (MSCA) branch adopts a parallel convolutional path design, where two groups of depth-wise separable convolutions with kernel configurations of 1×k and k×1 are employed to capture local features at varying scales. Specifically, convolutional kernels with sizes k=3 and k=7 are utilized, focusing on fine-grained and coarse-grained feature extraction, respectively. To integrate multi-scale information effectively, a residual connection mechanism is implemented, where the output of the MSCA branch (denoted as out_msca) is fused with the result of a convolution operation on the input feature map (conv(x)), following the formula: out_msca = out_msca + conv(x). This design enables the model to comprehensively exploit hierarchical feature representations across different spatial scales.

SPConvNet

The original backbone of YOLOv11 consists of Conv, C3K2, SPPF, and a newly added C2PSA module. In this architecture, each convolutional layer employs fixed-size kernels, which limits the receptive field and generates a substantial amount of redundant feature maps during image processing. These redundant feature maps not only increase computational complexity but also significantly elevate the number of parameters, adversely impacting the overall efficiency of the model. To address these issues and reduce training time, this study draws inspiration from the efficient convolution process of the SPConv module in the SPConvNet architecture. Specifically, SPConv convolution was introduced, and SPConv was integrated into the backbone network, replacing the original CBS module in the backbone network. These improvements were incorporated into the YOLOv11 architecture to reduce the computational burden of the deep neural network and enhance the overall performance of the model.

All existing filters, such as vanilla convolution, Ghost Conv (Han *et al.*, 2020), Oct Conv (Chen *et al.*, 2019) and Het Conv (Singh *et al.*, 2019), perform k × k convolution on all input channels. However, after traditional convolution divides all input channels into two major parts, redundancy may occur between the representative parts. Meanwhile, there are also no two identical channels so that we cannot throw away these redundant channels neither. In other words, representative channels can be divided into several parts, each representing a type of primary feature, such as color and texture. SPConv convolution performs grouped convolution on the channels to further reduce redundancy, as shown in the middle part of Figure 5. Grouped convolution can be viewed as a virtual convolution with sparse block-diagonal convolution kernels, where each block corresponds to a channel partition, and there are no connections between partitions. This means that after grouped convolution, this fusion method can further reduce redundancy between representative parts while inevitably cutting off potentially useful cross-channel connections. All SPConv structures compensate for this information loss by adding pointwise convolution between all representative channels. Unlike conventional depth-wise separable convolution, where grouped convolution and pointwise convolution are applied sequentially, the proposed structured point convolution (SPC) performs both grouped weight convolution (GWC) and pointwise convolution (PWC) in parallel on the same set of representative channels. Specifically, as shown in Eq. (3), the output feature y consists of two components: (1) a diagonal matrix W^{p} applied to the representative channels z, which models channel-wise interactions (2) $W^{p} \in {}^{\sim\, G\times G}$ is a diagonal weight matrix, where each diagonal element W_{gg} scales the corresponding representative channel z_{g}, enabling efficient inter-group communication while preserving group-wise specialization.

$$\begin{bmatrix} w^{p}_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w^{p}_{GG} \end{bmatrix} \begin{bmatrix} z_{1} \\ \vdots \\ z_{G} \end{bmatrix} + \begin{bmatrix} w_{11} & \cdots & w_{1,\alpha L} \\ \vdots & \ddots & \vdots \\ w_{M,1} & \cdots & w_{M,\alpha L} \end{bmatrix} \qquad \text{(Eq. 3)}$$

So far, SPConv splits the vanilla 3×3 convolution into two operations. For the representative part, it conducts a direct fusion of 3×3 group convolution and 1×1 pointwise convolution to counteract information loss caused by grouping. For the redundant part, it applies 1×1 kernels to preserve minor yet useful details. As a result, this processing strategy generates two distinct types of features. Given that these features originate from different input channels, it is essential to introduce an effective fusion mechanism to regulate and integrate the information flow.

Efficient lightweight multi-path detection head

Following the contextual multi-scale cross aggregation (CMCA) module, the feature maps inherently encode local contextual information and exhibit enhanced discriminative capabilities for small object representation. To further model the global dependencies between small objects and complex backgrounds, we integrate SCAM into the detection head. Unlike conventional approaches that rely on backbone networks for global relationship modeling, SCAM operates at the head stage to efficiently leverage cross-space pixel interactions. Inspired by GNet (Cao *et al.*, 2019) and SCP (Liu *et al.*, 2024a), SCAM employs a triple-branch architecture: The first branch aggregates global spatial statistics through parallel global average pooling (GAP) and global max pooling (GMP), capturing both holistic feature distributions and salient regional patterns. The second branch generates linearly transformed feature representations ("value" in attention terminology; (Bera *et al.*, 2021) *via* a 1×1 convolution, preserving spatial coherence while enabling feature recombination. The third branch simplifies the computation of query-key correlations using a dedicated 1×1 convolution (denoted as QK in Figure 4), effectively reducing dimensionality without compromising attention efficacy. The structure of SCAM is shown in Figure 6.

The SCAM mechanism strategically combines these branches through dual matrix multiplications. The GAP/GMP-enhanced global context from the first branch interacts with the value features to model channel-wise dependencies, while the QK-optimized spatial attention weights from the third branch refine spatial correlations. These operations yield two complementary contextual representations: cross-channel relationships and spatial attention maps. A broadcast Hadamard product subsequently fuses these representations, dynamically suppressing irrelevant background regions while amplifying object-background discriminability. This design ensures computational efficiency while addressing the critical challenge of small object detection in cluttered environments. In each layer, the pixelwise spatial context can be expressed as follows:

$$\mathrm{Q}_{\mathrm{i}}^{\mathrm{j}}=\mathrm{P}_{\mathrm{i}}^{\mathrm{j}}+\alpha_{\mathrm{i}}^{\mathrm{j}}\sum_{\mathrm{j=1}}^{\mathrm{N_i}}\left[\frac{\exp\left(\omega_{\mathrm{qk}}\mathrm{P}_{\mathrm{i}}^{\mathrm{j}}\right)}{\sum_{\mathrm{n=1}}^{\mathrm{N_i}}\exp\left(\omega_{\mathrm{qk}}\mathrm{P}_{\mathrm{i}}^{\mathrm{n}}\right)}\cdot\omega_{\mathrm{v}}\mathrm{P}_{\mathrm{i}}^{\mathrm{j}}\right] \qquad \text{(Eq. 4)}$$

$$\mathrm{a}_{\mathrm{i}}^{\mathrm{j}}=\frac{\exp\left(\left[\arg\left(\mathrm{P}_{\mathrm{i}}\right);\max\left(\mathrm{P}_{\mathrm{i}}\right)\right]\mathrm{P}_{\mathrm{i}}^{\mathrm{j}}\right)}{\sum_{\mathrm{n=1}}^{\mathrm{N_i}}\exp\left(\left[\arg\left(\mathrm{P}_{\mathrm{i}}\right);\max\left(\mathrm{P}_{\mathrm{i}}\right)\right]\mathrm{P}_{\mathrm{i}}^{\mathrm{n}}\right)}\cdot\omega_{\mathrm{v}} \qquad \text{(Eq. 5)}$$

Where $\mathrm{P}_{\mathrm{i}}^{\mathrm{j}}$ and $Q_{\mathrm{i}}^{\mathrm{j}}$ represent the input and output of the j th pixel in the i-level feature map, respectively. N_i denotes the total number of pixels. ω_{qk} and ω_{v} are the linear transform matrices for projecting the feature maps, which simplify by 1x1 convolution. The operators correspond to GAP and GMP, respectively. By aggregating spatial information across the entire feature map, GAP and GMP explicitly model channel-wise discriminative cues, enabling SCAM to highlight channels with critical semantic information. This mechanism facilitates the learning of long-range contextual dependencies along the channel dimension, thereby enhancing the module's capability to capture cross-channel semantic relationships.

Experiment and Results

Experimental configuration and evaluation indicators

The experiments in this study were conducted using an NVIDIA GeForce RTX 4090 Laptop 64G GPU on the Windows 11 operating system. The experimental environment was configured with CUDA 11.0, Python 3.10, and PyTorch 2.1.0. The model was constructed, trained, and tested using the PyCharm 2023.2.1 deep learning framework. In addition, the hyperparameter settings for training and testing across all models were kept consistent. The detailed hyperparameter settings are provided in Table 1.

The performance evaluation metrics used in this study include precision (P), recall (R), mean average precision (mAP@0.5), the number of parameters (Param), and floating-point operations per second (FLOPs/G). The definitions of these metrics are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{(Eq. 6)}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \text{(Eq. 7)}$$

$$\text{AP} = \int_0^1 \text{P(R)dR} \qquad \text{(Eq. 8)}$$

$$\text{map} = \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} \text{AP}_\text{i} \qquad \text{(Eq. 9)}$$

$$\text{Params} = \text{C}_0 \times (\text{k}_\text{w} \times \text{k}_\text{h} \times \text{C}_\text{i} + 1) \qquad \text{(Eq. 10)}$$

$$\text{FLOPS} = \text{params} \times \text{W} \times \text{H} \qquad \text{(Eq. 11)}$$

where true positives (TP) represent the number of samples correctly identified by the model as containing wood defects. False positives (FP) refer to the number of samples incorrectly classified by the model as having defects when no defects are present. False negatives (FN) denote the number of samples where the model fails to detect wood defects, misclassifying them as background. AP evaluates the balance between precision and recall, specifically represented by the area under the precision-recall (P-R) curve. MAP is the average of AP values across all mushroom cap and serves as a comprehensive metric to assess the overall performance of the model. Floating point operations (FLOPs) indicate the number of floating-point operations performed during a single forward pass of the model, typically expressed in billions of FLOPs (GFLOPs). FLOPs can be viewed as a measure of the model's temporal complexity. Models with fewer parameters and FLOPs are generally more lightweight and can operate with lower computational overhead.

Ablation experiments

Ablation experiment of the attention module

To verify the effectiveness and superiority of the proposed CMCA for multi-behavior recognition of mushrooms in the complex environment of mushroom greenhouses, this section presents a comparative analysis with several classic attention mechanisms, including CBAM (Woo *et al.*, 2018), ECA (Wang *et al.*, 2020), SE (Hou *et al.*, 2021), CA (Gu *et al.*, 2020), and GAM (Liu *et al.*, 2021). The experimental results, summarized in Table 2, highlight the performance differences across these modules when integrated into the baseline YOLO11s model and their performance differences are evaluated using three key metrics (precision, recall, and mAP@0.5) with the results visualized in the zigzag line of Figure 7.

Compared to the baseline YOLO11s, all tested attention mechanisms (CBAM, ECA, SE, CA, GAM, and CMCA) exhibit varying degrees of improvement in recognition performance, as reflected by precision, recall, and mAP@0.5 metrics. Among these, CMCA stands out: it not only achieves the highest gains in key accuracy indicators (precision, recall, and mAP@0.5) but also demonstrates the lowest computational complexity (FLOPs) among all compared modules. This dual advantage-superior accuracy enhancement and reduced computational cost-validates the effectiveness of CMCA in addressing the challenges of multi-behavior recognition in mushroom greenhouses.

Overall ablation experiment of improved YOLOv11 model

To systematically evaluate the effectiveness of the lightweight SCS-YOLO architecture and its enhanced modules (SPConv, CMCA, and SCAM), ablation experiments were conducted by integrating different combinations of these modules. The experimental design, detailing the module configurations of each model, is presented in Table 3; the baseline is YOLO11s, and Models A to F represent incremental improvements with specific module combinations (SPConv, CMCA, and SCAM), while SCS-YOLO integrates all three modules. The performance metrics of these models, including precision, recall, mAP@0.5, FLOPs, and parameters, are summarized in Table 4, enabling a comprehensive analysis of the modules contributions to both accuracy and complexity.

The final SCS-YOLO (SPConv + CMCA + SCAM) achieves the optimal balance: precision (84.0%), recall (68.9%), and mAP@0.5 (79.0%) reach the highest levels, surpassing all ablation models. Meanwhile, FLOPs (18.5×109) and parameters (8.2×106) are minimized, even lower than most single- and pairwise combinations. This confirms that SPConv, CMCA, and SCAM collectively enhance feature representation, reduce computational redundancy, and synergistically improve both detection accuracy and model lightweight Ness. In summary, the ablation experiments validate that each module contributes distinctively -SPConv reduces complexity, CMCA enhances accuracy, and SCAM balances both- and their integration in SCS-YOLO realizes a superior lightweight detection architecture with state-of-the-art performance.

A comprehensive comparative analysis of eight models is presented in the radar chart (Figure 8), which evaluates performance across multiple indicators including mAP50, model volume, parameter count, computational complexity, and average inference time. In this visualization, each curve corresponds to a model, with the distance from the center to the edge reflecting performance on individual metrics - the closer to the edge, the better. Larger enclosed areas indicate superior overall performance. The complete SCS-YOLO framework, represented by the outermost curve, achieves a precision of 84.0%, recall of 68.9%, and mAP@0.5 of 79.0%, while maintaining low computational cost (18.5 GFLOPs, 8.2M parameters). This demonstrates the effectiveness of the proposed architecture in jointly optimizing accuracy and efficiency. The performance gains over ablated variants (A-F) and partial combinations (inner curves) further validate the synergistic contribution of our three core components: i) the SPConv module for efficient hierarchical feature extraction; ii) the CMCA module integrating multi-scale spatial and global contextual attention; and iii) the SCAM module refining features through spatial-channel interactions. Together, these modules form a balanced and complementary design that outperforms alternative configurations in detecting small mushroom caps under complex greenhouse conditions.

To validate the superiority of the proposed algorithm, performance comparison experiments were conducted between SCS-YOLO11 and several widely used algorithms in the field of object detection, while keeping the experimental environment unchanged. These algorithms include: Faster-RCNN, SSD, RetinaNet, DETR, YOLOv5s, YOLOv6s, YOLOv7s-tiny, YOLOv8s, YOLOv9c, YOLOv10s and YOLOv11s. The detailed comparison results are shown in Table 5. Compared with the benchmark models, the proposed algorithm demonstrates a notable balance between detection accuracy and computational efficiency. Specifically, two-stage detectors such as Faster R-CNN achieve a relatively high mAP@0.5 of 74.9%, yet suffer from exorbitant computational costs (120.6G FLOPs, 42.7M Params) and slow inference (79.2 ms), making them ill-suited for real-time applications. As a single-stage classic, SSD exhibits limited performance with an mAP@0.5 of 70.1% and a slow inference speed of 35 ms, trailing behind most competitors. RT-DETR, despite its mAP@0.5 of 76.8%, incurs substantial model complexity (84.7G FLOPs, 48.6M Params) and 20.9 ms inference time, reflecting challenges in lightweight deployment.

Figure 9 presents a comprehensive comparison of ten object detection models across three critical metrics: detection accuracy (mAP50), parameter efficiency (in millions, M), and computational complexity (in gigaflops, G), visualized as a bubble chart. In this representation, the horizontal axis represents model size (parameter count), and the vertical axis shows detection accuracy (%). Optimal models are located toward the top-left corner. The size and color intensity of each bubble correspond to the model's computational cost, with smaller and lighter-colored bubbles indicating lower FLOPs and thus greater computational efficiency. This visualization enables an intuitive assessment of the trade-offs between accuracy and efficiency, clearly demonstrating the superior balance achieved by the proposed method in terms of high detection performance with minimal resource consumption.

As supported by the data in Table 5, while models such as RT-DETR (76.8% mAP@0.5) and YOLOv9s (77.2% mAP@0.5) demonstrate competitive detection accuracy, SCS-YOLO outperforms them in critical metrics for practical deployment. Among YOLO-series algorithms, the improved algorithm proposed in this study demonstrates significant advantages in both accuracy and model efficiency. While the: YOLOv11s (baseline) shows the lowest mAP@0.5 (72.4%) and moderate computational load (21.3G FLOPs, 9.4M Params), highlighting its inadequacy as a reference. YOLOv5s (73.6% mAP@0.5) and YOLOv7-tiny (71.8% mAP@0.5) prioritize lightweight design (7.9M and 6.5M Params, respectively) but sacrifice detection accuracy. YOLOv8s (75.5% mAP@0.5) and YOLOv10s (78.1% mAP@0.5) achieve faster inference (5.2 ms) but with relatively higher FLOPs (22.0G and 20.8G) and Params (9.2M and 8.8M). YOLOv9s (77.2% mAP@0.5) improves accuracy but at the cost of prolonged inference (35.5 ms), indicating suboptimal real-time performance.

In contrast, SCS-YOLO11 outperforms all competitors in accuracy and speed. It achieves the highest precision (84.0%), recall (68.9%), and mAP@0.5 (79.0%), surpassing the second-best YOLOv10s by 0.9 percentage points in mAP@0.5. Concurrently, it maintains the lowest computational overhead: 18.5G FLOPs (lower than YOLOv11s, YOLOv8s, and YOLOv10s),8.2M Params (lighter than YOLOv10s and YOLOv11s), and an inference time of 7.3 ms (significantly faster than RT-DETR, SSD, YOLOv9s, and Faster R-CNN). These results underscore the proposed model's superiority in lightweight design, real-time capability, and detection performance, particularly for resource-constrained agricultural tasks where both accuracy and efficiency are critical.

Figure 10 presents the detection results of the proposed SCS-YOLO11 model on representative images from the greenhouse mushroom dataset. It can be observed that SCS-YOLO11 achieves superior detection performance for small and densely clustered mushroom caps, particularly during the early fruiting stage. The enhanced localization and confidence scores effectively reduce missed detections, even under conditions of partial occlusion and background interference.

SCS-YOLO generalization evaluation on public dataset (VisDrone)

To further validate the generalization ability and small object detection performance of the proposed SCS-YOLO11s model, we conducted additional experiments on the publicly available VisDrone dataset, which is widely used for benchmarking object detection algorithms in complex aerial scenarios characterized by small, dense, and occluded targets.

Both the baseline YOLOv11s and the proposed ours were trained and evaluated under the same experimental settings. SCS-YOLO11s consistently outperformed the baseline model in terms of precision, recall, and mAP, especially for small and occluded targets. Specifically, the mAP increased from 41.6% to 43.1%, and the F1-score also showed a notable improvement, demonstrating that the introduced modules (SPConv, CMCA, SCAM) effectively enhance feature representation and attention across scales even beyond the original mushroom dataset. The proposed SCS-YOLO11s demonstrates consistent performance improvements over the baseline YOLOv11s across multiple evaluation metrics on the VisDrone dataset. These results underscore the model's strong generalization capability and robustness in complex real-world environments beyond greenhouse mushroom cultivation. The enhanced accuracy, combined with its lightweight design, makes SCS-YOLO11s a promising candidate for real-time detection tasks involving small and densely distributed objects in various practical scenarios.

Discussion

SCS-YOLO model demonstrates superior performance in small-object detection tasks, particularly under complex greenhouse conditions. As validated through systematic ablation and comparative experiments, the model achieves a significant improvement in detection accuracy while maintaining competitive efficiency. Specifically, it reaches the highest mAP@0.5 of 79.0%, outperforming all baseline and variant configurations. Ablation studies (Table 3 and Table 4) reveal the individual contributions of each module. The CMCA module significantly enhances feature discriminability, increasing mAP@0.5 by 2.4% (from 72.4% to 74.8%) with minimal computational cost (FLOPs: 21.1G vs. 21.3G baseline). The SCAM module improves attention to key regions, contributing a 1.6% gain in mAP@0.5 (to 74.0%) while reducing parameters by 9.6% (to 8.5M) and FLOPs by 12.2% (to 18.7G). The SPConv module achieves a 15.5% reduction in FLOPs and a 19.1% reduction in parameters (to 7.6M), though it slightly reduces accuracy (mAP@0.5 drops by 0.5%). When integrated, these modules synergize: SCS-YOLO achieves the best overall performance (mAP@0.5 = 79.0%), surpassing the YOLOv11s baseline by 6.6%, while maintaining a compact architecture (FLOPs: 18.5G; Params: 8.2M).

Comparative experiments further highlight SCS-YOLO's effectiveness. While two-stage detectors such as Faster R-CNN deliver relatively high accuracy (74.9% mAP@0.5), they incur excessive computational costs (120.6G FLOPs, 42.7M parameters, 79.2 ms inference time). One-stage detectors like SSD and RT-DETR offer better efficiency but either lack accuracy (SSD: 70.1% mAP@0.5) or have limited robustness. Within the YOLO family, SCS-YOLO achieves the highest accuracy, lowest latency (7.3 ms), and smallest model size (8.2M), outperforming YOLOv10s by 0.9 percentage points in mAP.

Despite its strengths, SCS-YOLO faces several limitations. First, under extreme scenarios - such as heavy occlusion (>70%) or low illumination (<50 lux)- the false detection rate increases to 9.1%, partly due to data imbalance (only 3.2% of training samples cover such conditions). Second, the inference speed is hardware-dependent: although 37.2 FPS is achieved on high-end GPUs (e.g.,4090D), performance drops significantly on edge devices, indicating the need for quantization or hardware-aware optimization. Third, the model’s generalizability across species is limited, with a 6.7% drop in mAP when applied to other fungi (e.g., shiitake), suggesting reliance on species-specific morphological cues.

To address these challenges, several future improvements are envisioned. First, synthetic data augmentation (e.g., GAN-based occlusion simulation and low-light synthesis) will be used to improve model robustness under rare or difficult conditions. Second, hardware-algorithm co-design strategies, such as dynamic resolution switching and neural architecture search, will be explored to optimize deployment across a range of computing environments. Third, the incorporation of cross-species features alignment techniques, such as meta-learning, could enhance generalization and extend the model’s applicability to broader agricultural tasks such as fruit counting or pest monitoring. These enhancements are expected to further bridge the gap between controlled laboratory conditions and real-world deployment.

Conclusions

This paper presents SCS-YOLO, a lightweight and high-accuracy object detection framework specifically designed for detecting small mushroom caps in complex agricultural environments. The proposed model integrates multi-scale attention mechanisms, spatial-channel feature fusion, and efficient convolutional operations to enhance detection accuracy while significantly reducing computational cost. Experimental results demonstrate that compared to the YOLOv11s baseline, SCS-YOLO achieves a 9.1% improvement in mAP@0.5, reduces the number of parameters by 12.8%, and runs at a faster inference speed of 7.3 ms per image. It also outperforms state-of-the-art models such as Faster R-CNN and YOLOv10s in both detection accuracy and efficiency. The proposed framework is well-suited for edge deployment in real-time agricultural applications, including early pest detection and small-cap monitoring. Moreover, the approach is also applicable to the detection of other edible fungi such as *Flammulina filiformis* (enoki mushroom) and *Hypsizygus marmoreus* (beech mushroom), demonstrating strong generalization capability and broad application potential in intelligent agriculture.

References

Bera, A., Wharton, Z., Liu, Y., Zhang, M., Wang, T., Kumar, R. 2021. Attend and Guide (AG-Net): A keypoints-driven attention-based deep network for image recognition. IEEE Trans. Image Process 30:3691-3704

Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M. 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934.

Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H. 2019. GCNet: Non-local networks meet squeeze-excitation networks and beyond. Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW). pp. 1971-1980.

Chen, C., Wang, F., Cai, Y., Yi, S., Zhang, B. 2023. An improved YOLOv5s-based Agaricus bisporus detection algorithm. Agronomy 13:1871.

Chen, W., Lu, J., Pei, T., Yuan, G. 2025. YOLOv8-AFA: A photovoltaic module fault detection method based on multi-scale feature fusion. Energy Sources Part A 47:657-676.

Chen, X., Liu, Y., Guo, W., Wang, M., Zhao, J., Zhang, X., Zheng, W. 2024. The development and nutritional quality of Lyophyllum decastes affected by monochromatic or mixed light provided by light-emitting diode. Front. Nutr. 11:1404138.

Chen, Y., Fan, H., Xu, B., Zhang, Z., Li, X., Wang, J. 2019. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR). pp. 3435-3444.

Dai, J., Li, Y., He, K., Sun, J. 2016. R-FCN: Object detection via region-based fully convolutional networks. Adv. Neural Inf. Process. Syst. 29:379-387.

Gao, C., Qiao, F., Zhang, X., Wang, H. 2014. An improved HOG based pedestrian detector. In: F. Sun, D. Hu, H. Liu (eds.), Foundations and practical applications of cognitive systems and information processing. Advances in intelligent systems and computing. Berlin, Springer. pp. 577-590.

Girshick, R. 2015. Fast R-CNN. Proc. IEEE Int. Conf. Comput. Vis. (ICCV). pp. 1440-1448.

Gu, R., Wang, G., Song, T., Zhang, Y., Li, H., Zhao, J. 2020. CA-Net: Comprehensive attention convolutional neural networks for explainable medical image segmentation. IEEE Trans. Med. Imaging 40:699-711.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C. 2020. GhostNet: More features from cheap operations. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR). pp. 1580-1589.

He, K., Gkioxari, G., Dollár, P., Girshick, R. 2017. Mask R-CNN. Proc. IEEE Int. Conf. Comput. Vis. (ICCV). pp. 2961-2969.

Hou, Q., Zhou, D., Feng, J., Wang, L., Chen, X., Zhao, H. 2021. Coordinate attention for efficient mobile network design. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR). pp. 13713-13722.

Huang, J., Zhang, X., Jia, L., Zhou, Y. 2024. A high-speed YOLO detection model for steel surface defects with the channel residual convolution and fusion-distribution. Meas. Sci. Technol. 35:105410.

Khanam, R., Hussain, M. 2024. What is YOLOv5: A deep look into the internal features of the popular object detector. arXiv:2407.20892.

Khanam, R., Hussain, M. 2024. YOLOv11: An overview of the key architectural enhancements. arXiv:2410.17725.

Kiran, J.S., Singh, N., Abbas, H.M., Saravanan, R. 2025. Automated fruit counting with YOLOv5 model for harvest management in orchards. Proc. Int. Conf. Autom. Comput. (ICAC). pp. 235-240.

Li, X.J., Xiao, S.J., Xie, Y.H., Chen, J., Xu, H.R., Yin, Y., Peng, H.P. 2024. Structural characterization and immune activity evaluation of a polysaccharide from Lyophyllum decastes. Int. J. Biol. Macromol. 278:134628.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P. 2017. Focal loss for dense object detection. Proc. IEEE Int. Conf. Comput. Vis. (ICCV). pp. 2980-2988.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. 2016. SSD: Single shot multibox detector. Proc. Eur. Conf. Comput. Vis. (ECCV). pp.:21-37.

Liu, Y., Shao, Z., Hoffmann, N., Zhang, L., Chen, W. 2021. Global attention mechanism: Retain information to enhance channel-spatial interactions. arXiv:2112.05561.

Lu, C., Liaw, J. 2020. A novel image measurement algorithm for common mushroom caps based on convolutional neural network. Comput. Electron. Agric. 171:105336.

Luo, S., Xu, Y., Zhang, C., Jin, J., Kong, C., Xu, Z., et al. 2025. LIDD-YOLO: A lightweight industrial defect detection network. Meas. Sci. Technol. 36:015001.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A. 2016. You only look once: Unified, real-time object detection. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). pp. 779-788.

Redmon, J., Farhadi, A. 2017. YOLO9000: Better, faster, stronger. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). pp. 7263-7271.

Ren, S., He, K., Girshick, R., Sun, J. 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39:1137-1149.

Shang, C., Zou, X., Zhou, X., Xiang, Y., Wu, M. 2023. Study on fusion clustering and improved YOLOv5 algorithm based on multiple occlusion of Camellia oleifera fruit. Comput. Electron. Agric. 209:107706.

Singh, P., Verma, V.K., Rai, P., Kumar, A., Sharma, R., Gupta, T. 2019. HetConv: Heterogeneous kernel-based convolutions for deep CNNs. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR). pp. 4835-4844.

Soudeep, S., Mridha, M.F., Jahin, M.A., Dey, N. 2024. DGNN-YOLO: Dynamic graph neural networks with YOLO11 for small object detection and tracking in traffic surveillance. arXiv:2411.17251.

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G. 2024. YOLOv10: Real-time end-to-end object detection. Adv. Neural Inf. Process. Syst. (NeurIPS) 37:107984-108011.

Wang, H., Feng, J., Yin, H. 2023. Improved method for apple fruit target detection based on YOLOv5s. Agriculture 13:2167.

Wang, Q., Wu, B., Zhu, P., Li, M., Zhang, Y., Liu, X. 2020. ECA-Net: Efficient channel attention for deep convolutional neural networks. Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR). pp. 11534-11542.

Woo, S., Park, J., Lee, J.Y., Kim, H., Choi, S., Hwang, S. 2018. CBAM: Convolutional block attention module. Proc. Eur. Conf. Comput. Vis. (ECCV). pp. 3-19.

Zhang, M., Ye, S., Zhao, S., Wang, W., Xie, C. 2025. Pear object detection in complex orchard environment based on improved YOLO11. Symmetry 17:255.

Zhao, M., Wu, S., Li, Y. 2023. Improved YOLOv5s-based detection method for Termitomyces albuminosus. Trans. Chin. Soc. Agric. Eng. 39:267-276.

Zhu, X., Su, W., Lu, L., Li, B., Cheng, T., Luo, J. 2020. Deformable DETR: Deformable transformers for end-to-end object detection. arXiv:2010.04159.

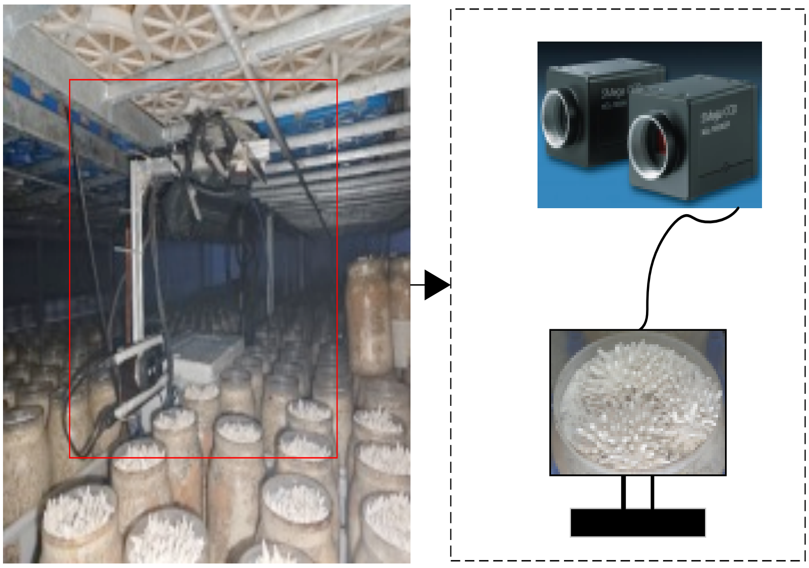

Figure 1. Diagram of dataset image acquisition.

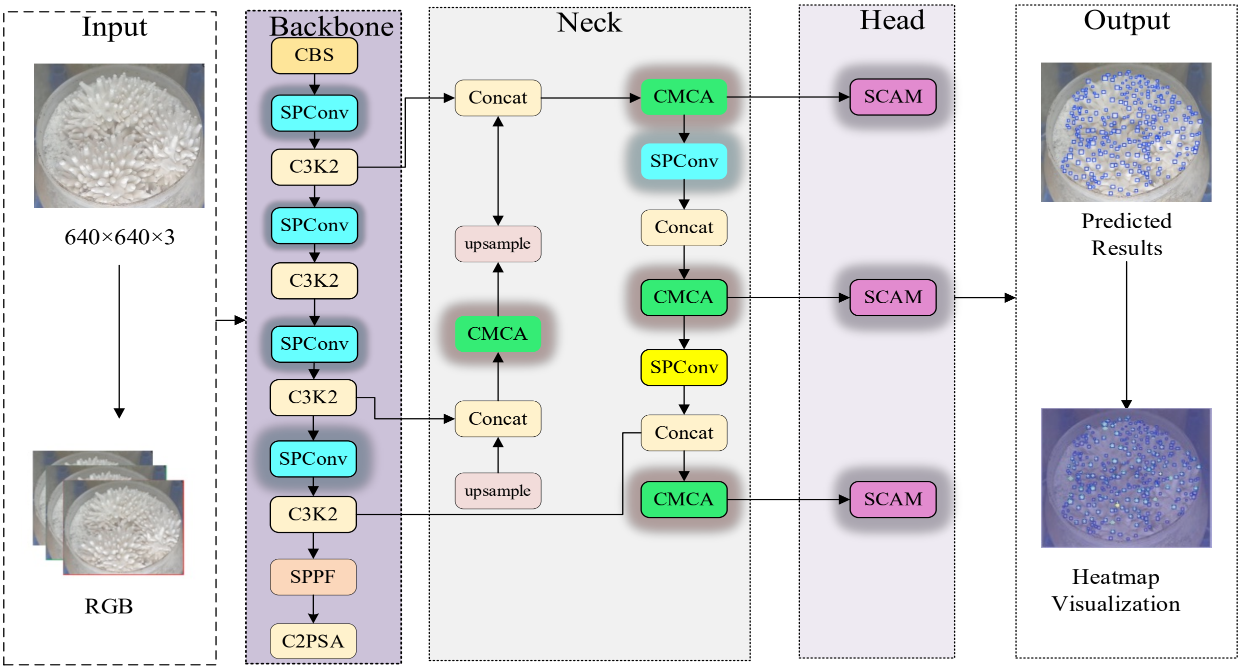

Figure 2. The architecture of the SCS-YOLO.

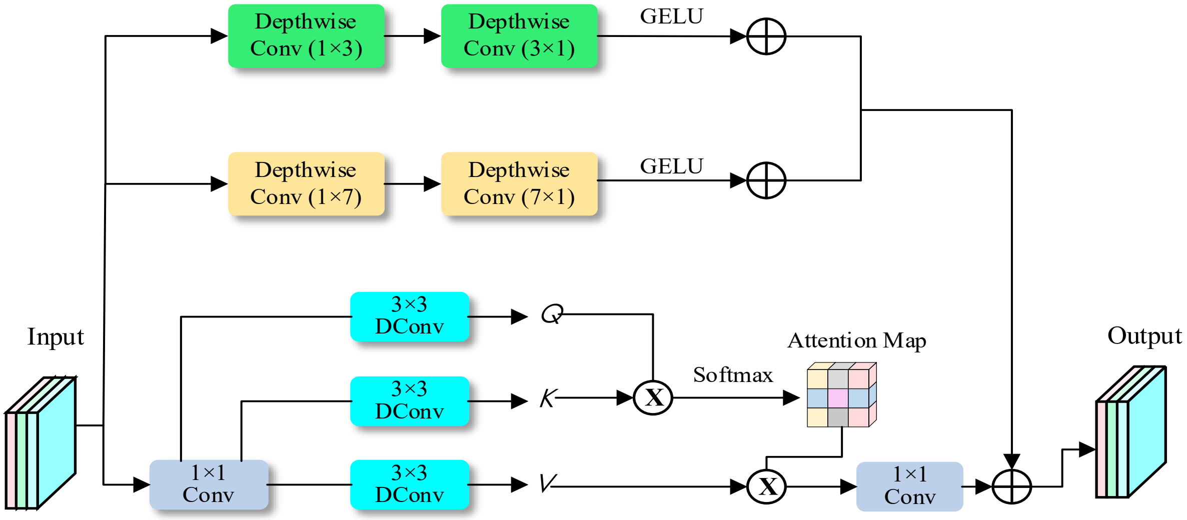

Figure 3. Attention and convolutional module.

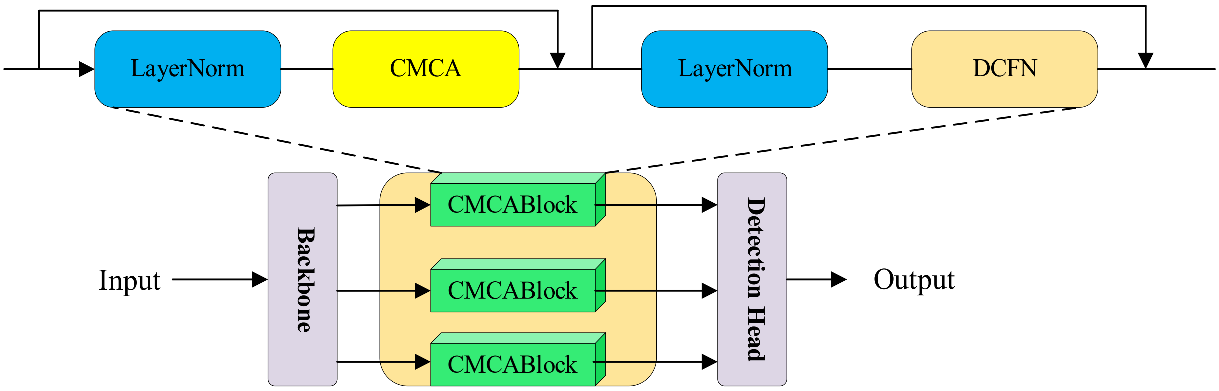

Figure 4. Attention and convolutional module with multi-scale neural network.

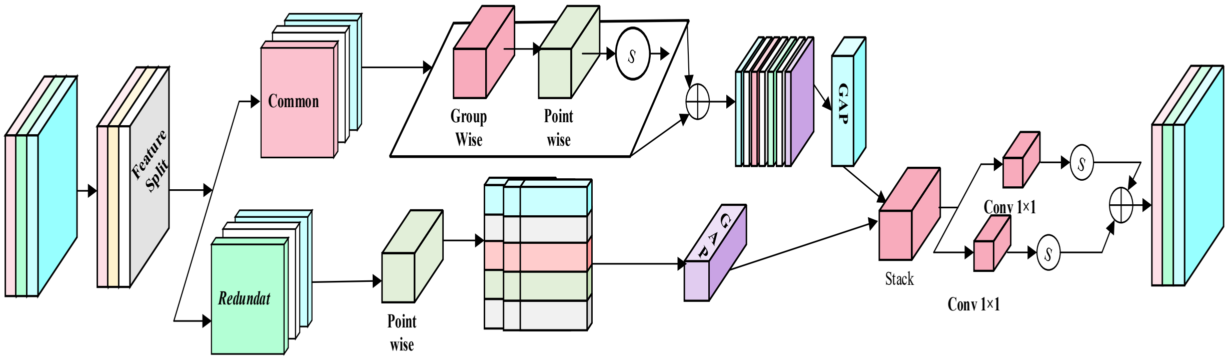

Figure 5. SPConv module.

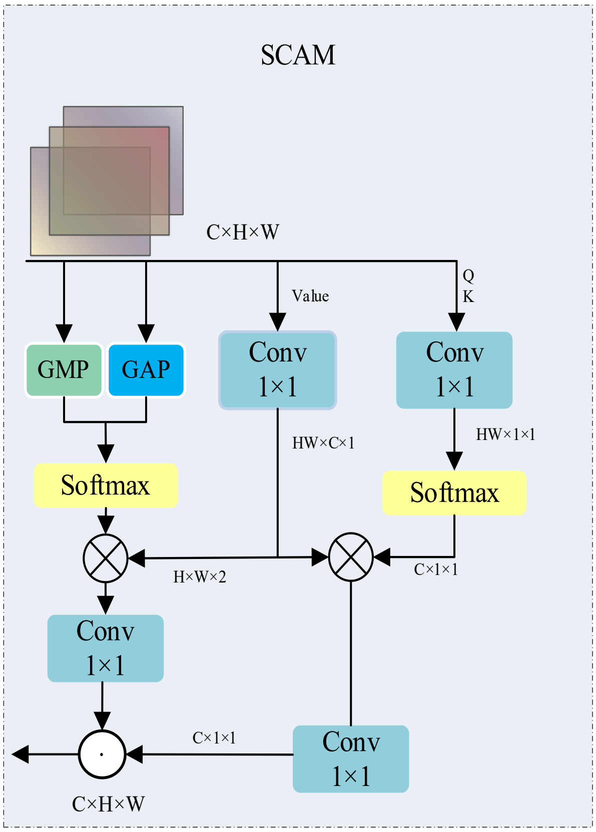

Figure 6. SCAM module.

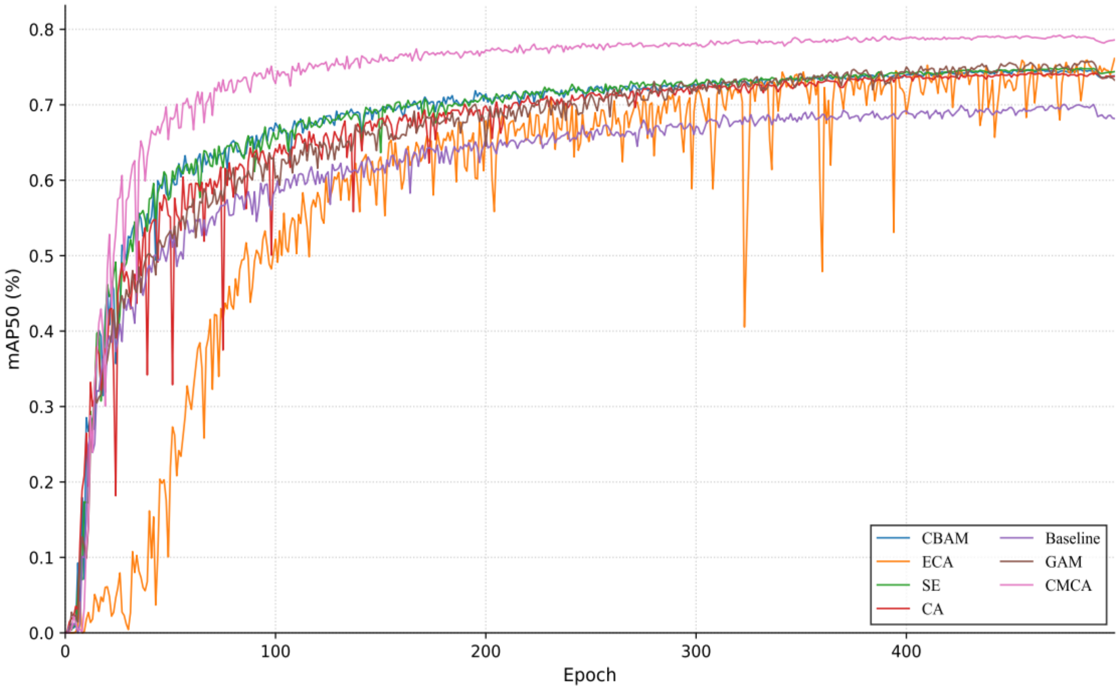

Figure 7. Comparison of mAP50 performance across 500 epochs.

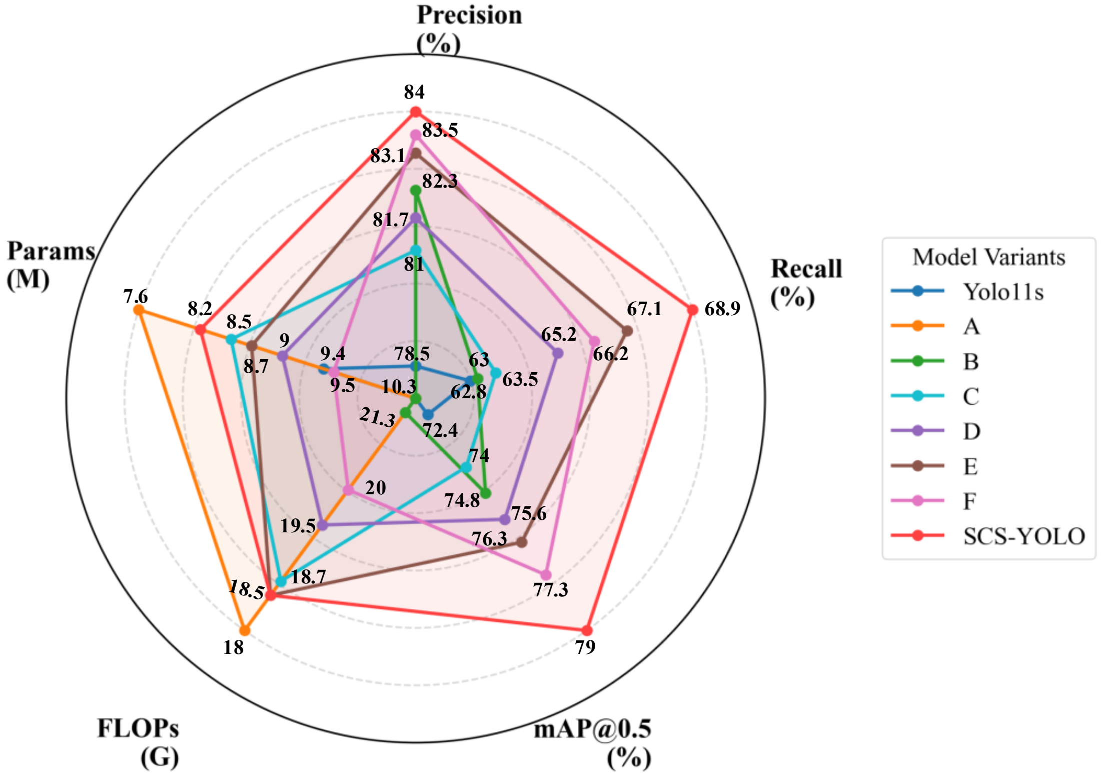

Figure 8. Performance comparison of eight models across multiple evaluation metrics.

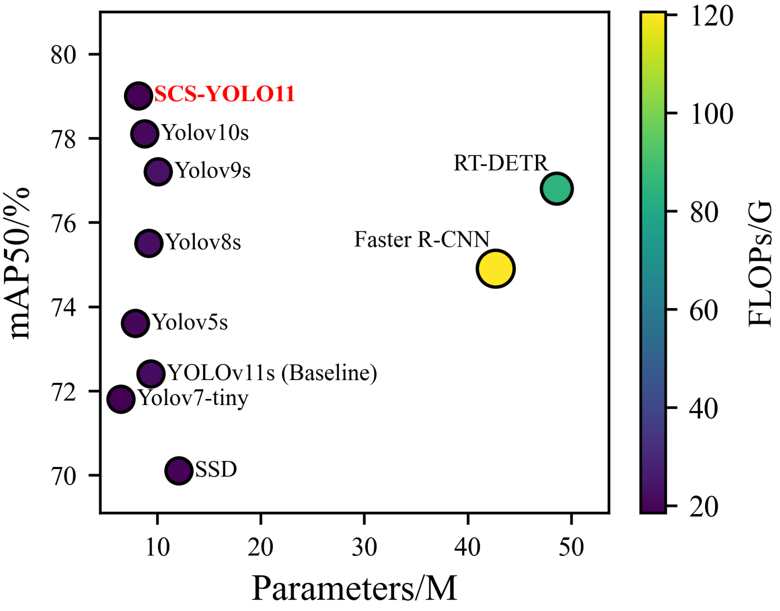

Figure 9. Comparison of 10 object detection models based on accuracy, parameter count, and computational complexity.

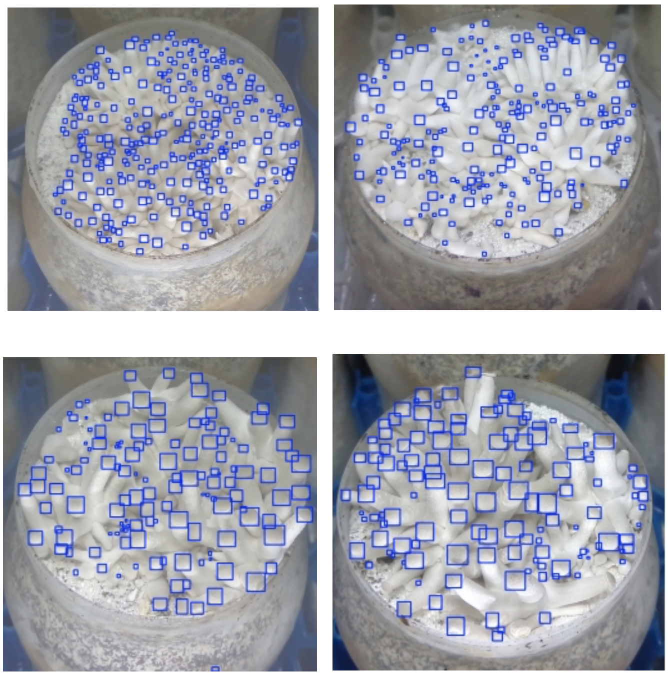

Figure 10. SCS-YOLO11 detection effect.

Table 1. Training parameter setting.

Hyper parameters	Values
Image size	640*640
Batch size	16
Epoch	500
Learning	0.01
Momentum	0.937
Weight_decay	0.0005

Table 2. Experiment results of different attention mechanism module in the same position.

Method	Precision %	Recall %	mAP@0.5 %	FLOPs 10^9
Yolo11s	78.5	62.8	72.4	21.3
CBAM	81.5	65.2	75.8	22.1
ECA	79.2	63.5	73.1	20.8
SE	80.1	64.3	74.6	21.7
CA	80.3	64.7	75.0	21.9
GAM	80.5	65.0	75.3	22.0
CMCA	**84.0**	**68.9**	**79.0**	**18.5**

Table 3. Description of the methods.

Method definition	SPConv	CMCA	SCAM
Yolo11s	-	-	-
A	√	-	-
B		√	
C			√
D	√	√	
E	√		√
F		√	√
SCS-YOLO	√	√	√

Table 4. Comparison of ablation experiment results.

Method definition	Precision %	Recall %	mAP@0.5 %	FLOPs 10^9	Parameters 10^6
YOLO11s	78.5	62.8	72.4	21.3	9.4
A	77.8	61.3	71.9	18	7.6
B	82.3	63.0	74.8	21.1	10.3
C	81	63.5	74	18.7	8.5
D	81.7	65.2	75.6	19.5	9.0
E	83.1	67.1	76.3	18.5	8.7
F	83.5	66.2	77.3	20.0	9.5
Ours	**84.0**	**68.9**	**79.0**	**18.5**	**8.2**

Table 5. Comparison of experimental results.

Model	Precision %	Recall %	mAP@0.5 %	FLOPs G	Params M	Time (ms)
YOLOv11s (baseline)	78.5	62.8	72.4	21.3	9.4	4.1
RT-DETR	82.1	66.5	76.8	84.7	48.6	20.9
SSD	75.8	60.2	70.1	19.5	12.1	35
Faster R-CNN	80.3	64.7	74.9	120.6	42.7	79.2
YOLOv5s	79.2	63.5	73.6	20.4	7.9	19.1
YOLOv7-tiny	77.9	62.0	71.8	18.7	6.5	68.3
YOLOv8s	81.0	65.3	75.5	22.0	9.2	5.2
YOLOv9s	82.5	66.8	77.2	23.1	10.1	35.5
YOLOv10s	83.2	67.5	78.1	20.8	8.8	5.2
Ours	**84.0**	**68.9**	**79.0**	**18.5**	**8.2**	**7.3**