Open Access

Journal of Artificial Intelligence & Robotics

Article type: Research Article

Scattered lens recognition method based on DSC-YOLOv5s model

Yanming Huo*; Congkang Zhang; Guo Xu; Weizhe Gao; Guo Zhang; Shenao Hao; Luyuan Jia; Yongdong Song; Jiajing Ma

Hebei University of Science and Technology, Shijiazhuang 050000, Hebei, China.

Corresponding Author: Yanming Huo

Hebei University of Science and Technology, Shijiazhuang 050000, Hebei, China.

Tel: +86-15203318323; Email: huoyanming@hebust.edu.cn

Received: Sep 15, 2025

Accepted: Oct 24, 2025

Published Online: Oct 31, 2025

Website: www.joaiar.org

License: © **Huo Y** (2025). *This Article is distributed under the terms of Creative Commons Attribution 4.0 International License*

Volume 2 [2025] Issue 2

Abstract

To address the accuracy and efficiency challenges of traditional lens placement equipment, a high-precision, lightweight method for scattered lens recognition is proposed. Based on the Yolov5s model, this method first embeds the Depthwise-Conv-BN-ReLU (DCBR) convolutional module into the network structure, replacing the original convolutional layers. This significantly reduces model complexity and improves detection speed. Secondly, by integrating the SE (Squeeze-Excitation) attention mechanism with the composite spatial pyramid pooling (SPPF-CBAM) network structure, it optimizes the integration of spatial and channel features, enhancing feature extraction quality. The proposed DSC-YOLOv5s algorithm was trained on a self-constructed scattered lens dataset and experimentally compared to the Yolov5s model. The average accuracy improved by 2.7%, demonstrating that the proposed model significantly improves the recognition accuracy of semi-transparent stacked lenses. To address the accuracy and efficiency challenges of traditional lens placement equipment, a high-precision, lightweight method for scattered lens recognition is proposed. Based on the Yolov5s model, this method first embeds the Depthwise-Conv-BN-ReLU (DCBR) convolutional module into the network structure, replacing the original convolutional layers. This significantly reduces model complexity and improves detection speed. Secondly, by integrating the SE (Squeeze-Excitation) attention mechanism with the composite spatial pyramid pooling (SPPF-CBAM) network structure, it optimizes the integration of spatial and channel features, enhancing feature extraction quality. The proposed DSC-YOLOv5s algorithm was trained on a self-constructed scattered lens dataset and experimentally compared to the Yolov5s model. The average accuracy improved by 2.7%, demonstrating that the proposed model significantly improves the recognition accuracy of semi-transparent stacked lenses.

Keywords: Pattern recognition; YOLOv5s; Deep separable convolutional network; CBAM; SENet; SPPF module; Image recognition.

Introduction

As an optical component, the market demand for lenses is growing rapidly, but there are problems of overlap and inconsistency in the installation process, which puts higher demands on recognition technology. As a key equipment in electronic manufacturing, placement machines are developing towards high precision, high efficiency, modularization, intelligence and greenness. Traditional mechanical positioning methods have disadvantages such as slow speed, low precision and easy damage to components, so intelligent algorithms and image processing technologies have become the key to achieving these requirements. Liu Hongda et al [1] proposed a CNN architecture design and optimization strategy for image classification tasks, focusing on the selection of convolution layers, pooling layers and activation functions. Gao Han et al [2] proposed a static image behavior recognition method combining ResNet and CBAM, using residual networks and convolutional attention mechanisms to improve the accuracy of behavior recognition in static

Citation: Huo Y, Zhang C, Xu G, Gao W, Zhang G, et al. Scattered lens recognition method based on DSC-YOLOv5s model. J Artif Intell Robot. 2025; 2(2): 1030.

images. For some small detection equipment, their computing power is limited, so the use of lightweight network structures has become an effective way to solve this problem. Depthwise separable convolution is an improved form of convolution operation [3-5], which reduces the amount of computation while maintaining the performance of the model. As a semi-transparent medium, lenses introduce image distortion, defocusing, and feature deformation, complicating feature extraction. Stacking lenses exacerbates the cumulative effect of illumination changes, leading to complex variations in brightness, shadows, and chromaticity, significantly increasing the complexity of object detection and segmentation. To address these issues, the DSC-YOLOv5s model integrates the SEnet channel attention and SPPF-CBAM spatial-channel dual-dimensional attention mechanisms within the YOLOv5s backbone feature extraction network to enhance multi-scale key feature extraction. It also introduces a lightweight DCBR module combined with the FReLU activation function to achieve lightweight deployment of the model. This model has been validated on a self-developed dedicated dataset, effectively improving lens recognition accuracy.

Materials and methods

Yolov5, a prominent model in the Yolo series, includes four models: Yolov5s, Yolov5m, Yolov5l, and Yolov5x, depending on the depth and width of the Yolov5 network [6,7]. Yolov5s, as the smallest and fastest model, meets the deployment requirements of small-scale placement machines. Therefore, the Yolov5s model was selected as the base model for this experiment. However, due to the Yolov5s network's shallow detection depth, its detection accuracy for multi-scale objects is relatively poor.

The DSC-YOLOv5s model replaces traditional convolutional layers (Conv) with depthwise separable convolutional modules in its backbone network, significantly reducing the number of model parameters. Furthermore, the SiLU activation function in the CBS module is replaced with FReLU, improving model flexibility and computational efficiency. Furthermore, the SE channel attention mechanism is introduced in the backbone network to adaptively adjust feature channel weights. The SPPF-CBAM spatial-channel attention mechanism is further integrated to optimize the spatial and channel feature information of the SPPF module, resulting in an SPPF-CBAM network architecture to improve feature extraction quality. Finally, we proposed a DSC-YOLOv5s detection model that integrates modules such as DCBR, SE, and SPPF-CBAM, significantly enhancing the performance of YOLOv5s.

The DSC-YOLOv5s model detection process is as follows: Collect a dataset, divide the collected dataset samples into training and test sets according to a certain ratio, and annotate them; configure the training set into the training algorithm for training; save the best model weight file generated during the training process; use the trained model to verify the test set to obtain the final model evaluation results. The details and effects of the algorithm improvements are shown in Figure 1.

Depthwise separable convolutional module

Because this model is designed to identify a large number of scattered lenses, requiring both high accuracy and rapid recognition, and small placement equipment is limited by computing resources, the Depthwise Separable Convolution Module (DCBR) was introduced to reduce the model's parameters and

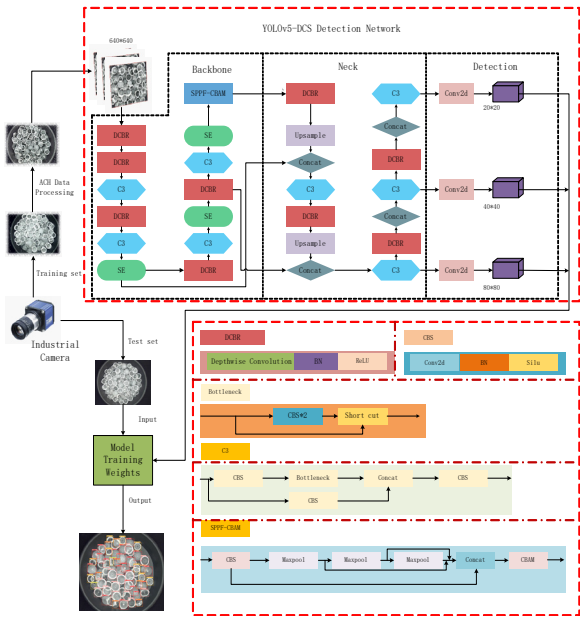

Figure 1: DSC-YOLOv5s detection framework.

speed up recognition. Deep separable convolution, consisting of Depthwise (DW) convolution and Pointwise (PW) convolution [8,9], reduces the number of parameters by 18% compared to traditional convolutional prediction models.

Figure 2 shows a standard convolutional module. The image input is a 5×5 pixel, 3-kernel image. The convolutional layer has four filters, each containing three 3×3 (C) and 3×3 (W) kernels. After the image passes through the 3×3 convolutions of the four filters, four feature maps are output. Therefore, the formula used is as follows:

$$N = W \times H \times C_i \times C_o \quad (1)$$

$$C = N \times (W_p - W + 1) \times (H_p - H + 1) \quad (2)$$

Where W is the width of the convolution kernel, H is the height of the convolution kernel, W_p is the width of the input image, H_p is the height of the input image, C_i is the number of channels in the input image or the output feature map of the previous layer, C_o is the number of output channels, N is the number of parameters in the convolution layer, and C is the computational cost of the convolution layer. This calculation is shown in Figure 2, where N is 108 and C is 972.

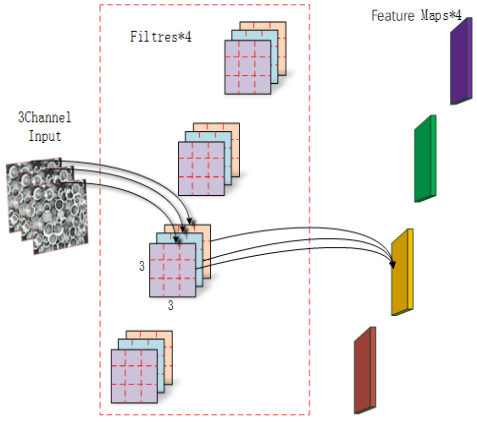

Figure 2: Standard convolution graph.

In the depthwise convolution module, one convolution kernel of the depthwise convolution is responsible for one channel, and one channel is convolved by only one convolution kernel. The number of channels of the Feature Maps generated by this process is the same as the number of channels of the input. In the convolution part of the depthwise convolution, a 5×5 pixel, 3-kernel image is input. First, the first convolution operation is performed. The DW is performed entirely in the two-dimensional plane. The number of convolution kernels is the same as the number of channels in the previous layer. The channels and convolution kernels correspond one to one, so a 3-channel convolution kernel generates 3 Feature Maps after operation, as shown in Figure 3. According to formulas (1) and (2), the DW convolution channel parameter N_{DW} is 27, and the calculation amount C_{DW} = 243.

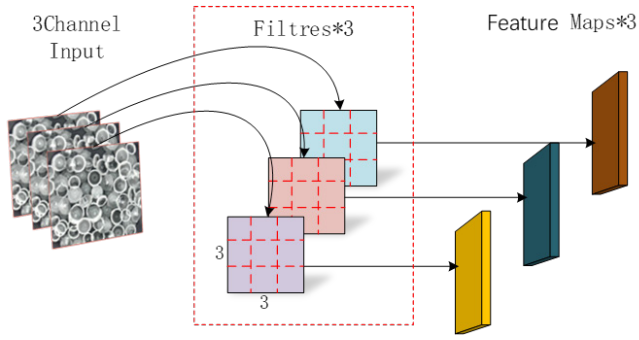

Figure 3: DW convolutional channel graph.

After Depthwise Convolution is completed, the number of feature maps obtained is the same as the number of channels in the input layer. However, this operation performs convolution operations on each channel of the input layer independently, and does not effectively integrate the feature information into spatial dimensions or fuse the features between different channels. Therefore, Pointwise Convolution is needed to combine these feature maps to generate new feature maps. The operation of Pointwise Convolution is very similar to that of standard convolution. The difference is that its convolution kernel size is 1×1×M, where M is the number of channels in the previous layer. As shown in Figure 4, the feature map output after DW convolution is used as the input of PW convolution. The convolution operation here will perform a weighted combination of the maps in the depth direction of the previous step to generate new feature maps. According to formulas (1) and (2), the number of PW convolution channel parametersis N_{DW} 12, and the calculation amount C_{DW} is 108.

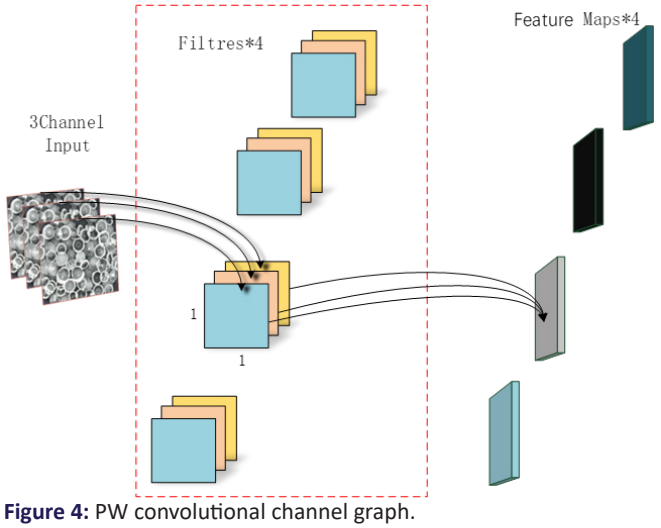

Figure 4: PW convolutional channel graph.

Table 1: PW convolutional channel graph.

Performance indicators	Conv	DCBR
Convolutional layer parameters	108	39
Computational amount	972	351

As shown in Table 1, for the same input and the same four feature maps, the number of convolutional layer parameters of depthwise separable convolution is approximately one-third of that of standard convolution, reducing the computational effort by 60%. Therefore, with the same number of parameters, neural networks using separable convolution can be deeper.

CBAM

The SPPF of the YOLOv5s backbone extraction network can extract and pool features at different receptive field scales, capturing objects of varying sizes and achieving adaptive output. However, its output feature maps have uniform weights, making it difficult to highlight key features. Given the large number of objects and their significant overlap, resulting in complex variations in brightness and shading within the image, the SPPF incorporates a spatial-Channel Attention Mechanism (CBAM). The CBAM module integrates the Channel Attention Mechanism (CAM) with the Spatial Attention Mechanism (SAM) to adaptively adjust the weights of the channel and spatial information in the feature map. By learning the relationship between feature map channels and spatial positions, it adaptively adjusts the weights of each channel, allowing the network to focus on useful feature channels and suppress irrelevant information responses, helping to extract important image features and enhancing the model's discriminative capabilities. Its structure is shown in Figure 5:

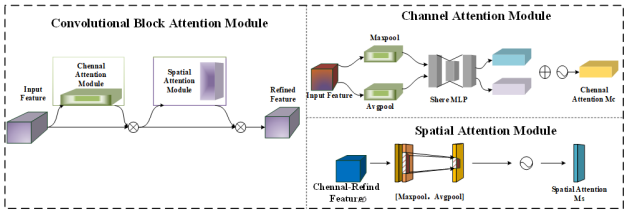

Figure 5: CBAM Structure Diagram.

The CAM channel attention mechanism in CBAM utilizes the channel relationship between features to generate a channel attention map. Each channel of the feature map is considered a feature detector. CAM focuses on the important features in a given image. To effectively calculate channel attention and enhance spatial information aggregation capabilities, CAM uses both average pooling and maximum pooling to pool features, greatly improving the network's representational capabilities. The calculation formula is as follows:

$$Mc(F) = \sigma\left(MLP\left(AvgPool(F)\right) + MLP\left(MaxPool(F)\right)\right) \quad (3)$$
$$= \sigma\left(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1\left(W_0\left(F^c_{max}\right)\right)\right)$$

Where: σ is the sigmoid function, $W_0 \in R^{C/r\times C}$, $W_1 \in R^{C\times C/r}$ is the weight of MLPD, r=16, W_0 and W_1 are shared for both inputs, and W_0 is obtained after the ReLU activation function.

CBAM uses the spatial relationship between features to generate a spatial attention map (SAM). Unlike channel attention, spatial attention focuses on the position information of important features in the input image and supplements the position information of the feature map. The calculation formula is as follows:

$$M_s = \sigma\left(f^{7\times7}([AvgPool(F); MaxPool(F)])\right) = \sigma\left(f^{7\times7}\left[F^s_{avg}; F^s_{max}\right]\right) \quad (4)$$

Where: σ is the sigmoid function, $f^{7\times7}$ is the convolution operation, $F^s_{avg} \in R^{1\times H\times W}$; $F^s_{max} \in R^{1\times H\times W}$; represent the average pooling feature and maximum pooling feature across channels respectively.

CBAM is a lightweight general module that meets the deployment requirements of small-scale placement equipment [10-12]. Its main function is to perform weighted processing on feature maps at different levels to increase the network's attention to key features, thereby improving the model's expressiveness and performance.

Squeeze-excitation attention

SE is the most representative module of the channel attention mechanism. The SE structure diagram is shown in Figure 6. It focuses on the channel relationship and simulates the relationship between channels through the Squeeze-and-Excitation operation [13]. Each dimension of the feature vector is multiplied by a weight coefficient to adaptively adjust the importance of each channel. As shown in Figure 1, SENet is introduced in the 5th, 8th, and 11th layers of the Backbone network. The SE attention mechanism first maps the feature map $X \in R^{H'\times W'\times C'}$ to the feature map $U \in R^{H\times W\times C}$ through the convolution operation F_{tr}. In order to consider the feature map information of each channel, the Squeeze operation is used to compress the spatial information into a spatial descriptor z_c. The calculation formula is as follows:

$$z_c = F_{sq}(u_c) = \frac{1}{H_p \times W_p}\sum_{i=1}^{H_p}\sum_{j=1}^{W_p} u_c(i,\ j) \quad (5)$$

Where: F_{sq} represents the global average pooling operation, H_p and W_p represent the height and width of the input feature map, and u_c represents the new feature map $u_c \in R^{H_p\times W_p}$ obtained by the convolution operation of the feature map **X**;

The Excitation operation uses the Sigmoid activation function to act on a simple gating mechanism to capture channel dependencies.

$$s = F_{ex}(z_c,\ W) = \sigma\left(g(z_c,\ W)\right) = \sigma\left(W_2\delta(W_1 z_c)\right) \quad (6)$$

$$x_c = F_{scale}(u_c,\ s_c) = s_c u_c \quad (7)$$

Where z_c is obtained by the Squeeze operation, δ is the ReLU function, $W_1 \in R^{\frac{c}{r}\times C}$ and $W_2 \in R^{C\times\frac{c}{r}}$, r is a weight parameter (r=16 in this paper) to reduce the number of channels and thus the computational effort, and s_c is a weight parameter. Finally, by changing the weight s_c, the feature map is scaled to obtain the output of the SE Block.

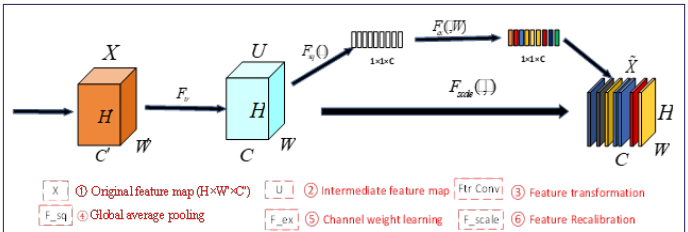

Figure 6: SENet structure diagram.

Experiment and analysis

This experiment was trained on Huawei's Ascend cloud servers. Ascend serves as the foundation for intelligent computing. Its AI infrastructure includes the Atlas series hardware, the heterogeneous computing architecture CANN, and the ModelArts one-stop AI development platform. This training utilized the ModelArts intelligent development platform. The Atlas series hardware included the Ascend 910 processor, the Atlas 800 9000 training server, and a 24-core CPU, providing AI computing power at the center and edge.

Collecting labeled datasets

Lenses, as key optical components, are widely used in fields such as lighting and optical instruments, and are in high demand. However, the assembly process can present issues with lens stacking and discrepancies between the front and back sides. This study focused on collecting data from large stacks of focusing convex and diffuse lenses. A vibrating chassis was used to dynamically alter the front and back sides of the lenses to identify lenses that met assembly standards. The experiment simulated a scenario where the number of lenses decreases during assembly by continuously vibrating the lens storage area and capturing images while gradually removing lenses, thereby optimizing identification and assembly efficiency. A total of 1,842 images were captured in this dataset.

After the dataset was collected, it was annotated. The dataset label classification is shown in Table 2. Labelimg software was used to annotate each category and generate a label file in YOLO format. 1,242 images from the annotated dataset were then used as the training set, 400 images as the validation set, and 200 images as the test set.

Table 2: Data set label classification.

Lens classification	Front	Obverse	Vertical
Focusing convex lens	CCLF	CCLB	Upright
Diffuse lens	DLF	CCLF	

ACH image enhancement

Among image processing methods, adaptive methods are those that process images based on their inherent information and characteristics. Because optical lens datasets can experience reflections and ghosting during capture, image details can be lost or confused. Therefore, the ACH image enhancement algorithm [14-16] is used to enhance local contrast using the local standard deviation, as shown in Figures 7(a) and (b). The formula is as follows:

$$M(i,\ j) = \frac{1}{(2n+1)(2m+1)}\sum_{s=i-n}^{i+n}\sum_{k=j-m}^{j+m} f(s,\ k) \quad (8)$$

$$\sigma(i,\ j) = \frac{1}{(2n+1)(2m+1)}\sum_{s=i-n}^{i+n}\sum_{k=j-m}^{j+m}\left(f(s,\ k) - M(i,\ j)\right) \quad (9)$$

For each point in the image, calculate its local mean and local standard deviation. In formulas (8) and (9), $f(s,\ k)$ represents the pixel value at coordinate $(s,\ k)$; $M(i,\ j)$ is the local mean of the area centered at point $(i,\ j)$ with window sizes of $(2n+1)$ and $(2m+1)$; and $\sigma(i,\ j)$ is the local variance. After obtaining the local mean and standard deviation, the image enhancement operation is performed. The formula is as follows:

$$I(i,\ j) = M(i,\ j) + G\left(f(i,\ j) - M(i,\ j)\right) \quad (10)$$

$$G = \alpha\frac{M}{\sigma(i,\ j)} \quad (11)$$

Where $I(i,\ j)$ is the enhanced pixel value and M is the global average value.

Using ACH of color images, first, the input image is converted from BGR color space to LAB color space, and then the feature-enhanced LAB color space image is obtained through local

image enhancement algorithm. Finally, the equalized LAB image is converted back to BGR color space.

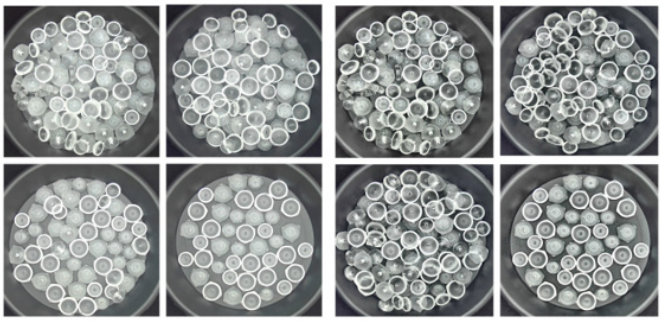

Figure 7: Effects before and after ACH enhancement. **(A)** Before ACH enhancement. **(B)** After ACH enhancement.

Experimental results analysis

To test the advantages of the DSC-YOLOv5s model, the model was trained on a self-constructed dataset and compared with mainstream algorithms such as the original YOLOv5s, Faster-Rcnn, SSD, EfficientNet, YOLOv4, and YOLOv5-GhostNe. The training parameters are shown in Table 3. All input images were resized to 640 × 640 pixels to improve the detection accuracy of the model and adapt to the input required by the network framework. The detection effect of the DSC-YOLOv5s model is shown in Figure 8(a), and the detection effect of the YOLOv5s model is shown in Figure 8(b). The batch size is 6, and the number of training epochs is set to 200. The average precision mAP@50, mAP@50:95, and Frames Per Second (FPS) are used as evaluation indicators. AP is expressed as:

$$AP = \int_0^1 P_R dR \qquad (12)$$

Where: P_R is the precision, this function corresponds to different confidence levels; mAP is expressed as:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \qquad (13)$$

Where: N represents the epoch of training; FPS represents the detection speed, which is expressed as:

$$FPS = \frac{N}{t_N} \qquad (14)$$

Where: t_N represents the total time spent on detecting the image.

Table 3: Training parameter.

Input image size	Batch size	Initial learning rate	Decay index
640×640	6	0.01	0.01

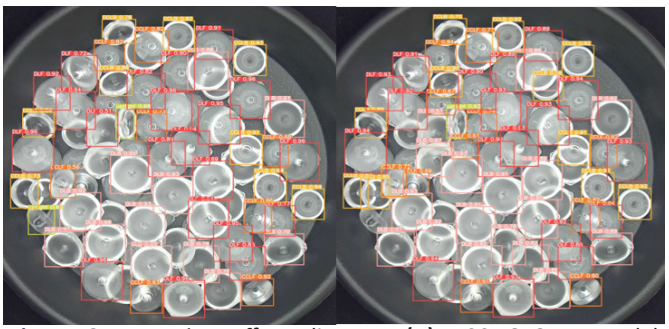

Figure 8: Detection effect diagram. **(A)** DSC-YOLOv5s model detection effect. **(B)** YOLOv5s detection effect.

Table 4 shows the comparison results of the DSC-YOLOv5s algorithm with mainstream algorithms such as YOLOv5s, Faster-RCNN, SSD, EfficientNet, YOLOv4, and Y-OLOv5-GhostNet. Compared with the original YOLOv5s, DSC-YOLOv5s' average accuracy (mAP@0.5 and mAP@0.5:0.95) improved by 2.7% and 0.3%, respectively, and its FPS increased by 4.1, increasing detection speed by approximately 10%. Compared with other algorithms, DSC-YOLOv5s achieves significantly higher average accuracy. While its detection speed is slightly slower than YOLOv5-GhostNet, it still meets the requirements of real-time detection. Its parameter count and computational complexity are superior to most mainstream algorithms.

Table 4: Comparison of experimental results of DSC-YOLOv5s and other models on the lens dataset.

Method	Image size	mAP @0.5/%	mAP @0.5:0.95/%	FPS	FLOPs
SSD [17]	640×640	91.2	34.7	6.4	36.63
EfficientNet [18]	640×640	93.5	76.8	6.9	50.43
YOLOv10 [19]	640×640	92	72.3	4.9	67.43
YOLOv5-GhostNet [20]	640×640	94.5	80	55.2	8.7
YOLOv5s	640×640	92.4	80.6	42.6	15.8
Our	640×640	95.1	80.9	46.7	6.3

As shown in Figure 9, by comparing the training performance curves of different object detection models, the DSC-YOLOv5s model demonstrates significant advantages in both convergence speed and detection accuracy. The mAP@0.5 curve shows that the DSC-YOLOv5s model exhibits a sustained and rapid upward trend in the early stages of training, achieving data convergence in only approximately 30 epochs, 60 epochs earlier than the YOLOv4 and SSD models, and exhibits no significant fluctuations during the convergence process. Ultimately, the DSC-YOLOv5s model achieves an mAP@0.5 score of 0.951, surpassing the other compared models (YOLOv5-GhostNet: 0.945, SSD: 0.912, EfficientNet: 0.935). This result demonstrates that the DSC-YOLOv5s model, through its introduced attention mechanism and lightweight design, achieves high-precision and stable convergence in a very short training epoch.

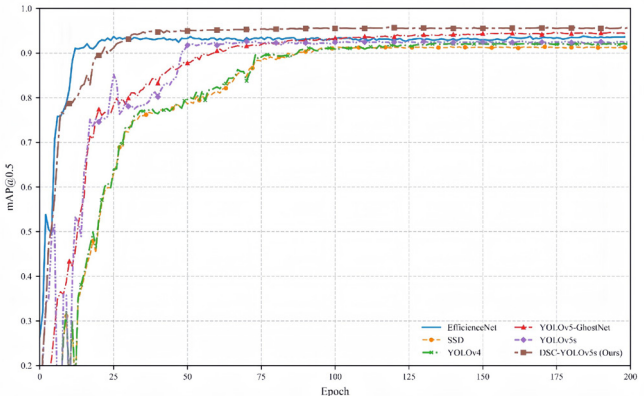

Figure 9: Comparison of mAP@0.5 value curves of different models.

Experimental results analysis

To verify the effectiveness of the various improved modules in the DSC-YOLOv5s algorithm, an ablation experiment was conducted. Using the same dataset and training parameters, the experiment systematically analyzed the impact of each improved structure on model performance by gradually introducing the SE, DCBR, and CBAM modules. The experimental results, as shown in Table 5, show that the introduction of the SE attention mechanism improves detection accuracy, but the increase in parameters and computational cost leads to a decrease in detection speed. Replacing the Conv with a lightweight DCBR module significantly reduces the number of parameters, in-

creases detection speed by 33%, and improves detection accuracy by 2.3% compared to using only the SE attention mechanism. The introduction of the CBAM module increases the network dimension and slightly decreases detection speed, but further improves detection accuracy by 0.3% compared to using only the SE attention mechanism. Each improved module plays a significant role in improving detection accuracy and optimizing computational efficiency.

Conclusion

Based on the YOLOv5s framework, the DSC-YOLOv5s algorithm is proposed for detecting semi-transparent stacked lenses by introducing the depth wise separable convolutional block (DCBR), the SE channel attention mechanism, and the CBAM attention mechanism. The algorithm replaces standard convolutional layers with the DCBR module, significantly reducing the number of model parameters. The SE and CBAM attention mechanisms enhance key feature extraction, effectively improving the recognition accuracy of stacked objects. The improved model achieves a balanced optimization between detection accuracy and computational efficiency while maintaining its lightweight. Experiments demonstrate that the algorithm maintains high detection stability in complex stacking scenarios, while its inference speed meets industrial real-time requirements and is adaptable to resource-constrained embedded devices.

This model is optimized and validated primarily for the detection of semi-transparent lenses in small placement machines. While its generalization remains to be improved, future applications of the model to other automated electronic component placement applications will be expanded to further enhance its versatility and scalability.

- **HighlightsNovel model:** This paper proposes a novel DSC-YOLOv5s model that achieves lightweight performance by embedding depth wise separable convolutions (DCBR) and integrating SE and CBAM attention mechanisms to enhance feature extraction. This model improves the average accuracy of scattered lens detection by 2.7%, significantly improving the recognition performance and speed of semi-transparent stacked objects.
- **Superior performance:** The model significantly improves the average detection precision (mAP) by 2.7% on the self-built dataset, and while significantly reducing the model complexity, it enhances the recognition robustness and detection speed of semi-transparent and stacked lenses.
- This method uses depthwise separable convolution to achieve model lightweighting and combines SE and CBAM dual attention mechanisms to enhance feature extraction capabilities. This collaborative strategy significantly improves computational efficiency while effectively ensuring recognition accuracy for complex targets.

Author declarations

Author contributions: Conceptualization, X.G. and C.Z.; methodology, C.Z. and J.M.; hardware, X.G. and C.Z.; writing code, C.Z. and W.G.; validation, G.Z. and L.J.; data resources, S.H., Y.S.; writing original manuscript, C.Z.; editing manuscript, G.Z. and Y.H. All authors have read and agreed to the published version of the manuscript.

Informed consent statement: Informed consent was obtained from all subjects involved in the study.

Data availability statement: The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

Acknowledgments: We thank Dr. Huo for insightful comments on the manuscript and colleagues in the lab for help with data analysis.

Conflicts of interest: The authors declare no conflicts of interest.

References

1. Liu H, Sun X, Li Y, et al. A survey of deep learning models for image classification based on convolutional neural networks. Computer Engineering and Applications. 2024; 60: 1-29.
2. Gao H, Wan F, Ma M. Static image action recognition method combining ResNet and CBAM. Journal of Zhengzhou University (Natural Science Edition). 2025; 57: 1-7.
3. Han W, Han X. Research on improved Faster R-CNN in stacked artifact recognition. In: Proceedings of the 12th International Conference of Information and Communication Technology (ICTech). Wuhan, China; 2023: 146-150.
4. Cao X, Chen X, Wei T. RISC-V-based accelerator for depthwise separable convolutional neural network. Chinese Journal of Computers. 2024; 47: 2536-2551.
5. Yang H, Zhang D, Xie P, et al. DSC-GRUNet: A lightweight neural network model for multimodal gesture recognition based on depthwise separable convolutions and GRU. Pattern Recognition Letters. 2025; 190: 35-44.
6. Qian Y, Wang Y, Yang X, et al. Algorithm-based optimization of YOLOv5s model for robot trajectory detection study. In: Proceedings of the 2024 IEEE 6th International Conference on Power, Intelligent Computing and Systems (ICPICS). Shenyang, China: IEEE; 2024: 1300-1304.
7. Chuang X, Qiang C, Yinyan S, et al. Improved lightweight YOLOv5n-based network for bruise detection and length classification of asparagus. Computers and Electronics in Agriculture. 2025; 233: 110194.
8. Rajanand A, Singh P. Stock price prediction using depthwise pointwise CNN with sequential LSTM. In: Proceedings of the 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC). Salem, India: IEEE; 2023: 82-86.
9. Li Z, Zhang Y, Wang Y, et al. Rapeseed seedling counting method based on YC-YOLO v7 model. Transactions of the Chinese Society for Agricultural Machinery. 2024; 55: 322-332.
10. Zu S. Coal gangue detection algorithm based on deep attention model and YOLOv8. Coal Mine Modernization. 2025; 34: 64-69.
11. Sun Z, Wang C. A Transformer network combing CBAM for low-light image enhancement. Computers, Materials & Continua. 2025; 82: 5205-5220.
12. Yang L, Lu Z, Shen M. Research on the effect and path of CBAM on green technology innovation in China's high-carbon manufacturing industries. Sustainability. 2025; 17: 2305.
13. Ali A. Multipath feature fusion for hyperspectral image classification based on hybrid 3D/2D CNN and squeeze-excitation network. Earth Science Informatics. 2023; 16: 175-191.
14. Thakur V, Singh H. An artificial neural network based adaptive histogram equalization algorithm for enhancement of low contrast images. In: Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development

(INDIACom). New Delhi, India. IEEE. 2021: 268-273.

15. Tan P, Ou B. Reversible contrast enhancement algorithm for medical images based on adaptive histogram equalization. Computer Science. 2024; 51: 394-400.

16. Rizwan K, Atif M, Zhonglong Z. Robust contrast enhancement method using a retinex model with adaptive brightness for detection applications. Optics Express. 2022; 30: 37736-37752.

17. Tao J, Miao W. Research on detection method of daily staff work management violation based on convolutional neural network and SSD algorithm. In: Proceedings of the 2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA). Dalian, China: IEEE; 2022: 605-609.

18. Naddaf-Sh S, Naddaf-Sh M, Kashani AR, et al. An efficient and scalable deep learning approach for road damage detection. In: Proceedings of the 2020 IEEE International Conference on Big Data (Big Data). IEEE; 2020: 5602-5608.

19. Xiao C, Chang L. Facial mask detection system based on YOLOv4 algorithm. In: Proceedings of the 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). Dalian, China: IEEE; 2022: 1032-1035.

20. Dou X, Wang T, Shao S. A lightweight YOLOv5 model integrating GhostNet and attention mechanism. In: Proceedings of the 2023 4th International Conference on Computer Vision, Image and Deep Learning (CVIDL). Zhuhai, China: IEEE. 2023: 348-352.