Image Anal Stereol 2025;44:69-86
Original Research Paper
doi: 10.5566/ias.3514

IMPROVEMENT OF YOLOV8 OBJECT DETECTION BASED ON LIGHTWEIGHT NECK MODEL FOR COMPLEX IMAGES

TIEN-WEN SUNG[1], JIE LI[1], CHAO-YANG LEE[✉,2] AND QINGJUN FANG[1]

[1]Fujian Provincial Key Laboratory of Big Data Mining and Applications, College of Computer Science and Mathematics, Fujian University of Technology, China, [2]Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Taiwan
e-mail: tienwen.sung@gmail.com, 2221308016@smail.fjut.edu.cn, chaoyang@yuntech.edu.tw, 89100941@qq.com

(Received January 3, 2025; revised March 5, 2025; accepted March 17, 2025)

ABSTRACT

With the advancement of target detection technology, the need for accurate detection of complex scenes is becoming increasingly important in various industries. This can not only improve productivity, but also ensure public safety. However, the current mainstream target detection algorithms have some problems in dealing with complex scenes, for example, some detection models are not able to detect in real time, and the accuracy of the model is degraded when facing disturbing factors such as target occlusion, and low-contrast scenes. In order for these problems to be mitigated, this paper proposes a lightweight convolution LDGConv (Lightweight-DepthGhost Convolution), which is utilized to improve the YOLOv8 network model by replacing part of the traditional convolution of the Neck network with this convolution, and improving the bottleneck module in a lightweight way. In addition, we add the Coordinate Attention mechanism to the Neck part. Our proposed model improves mAP50 on the VOC dataset by 1.3% while reducing computation and parameters by 9.8% and 15.3%, respectively, compared to the original model. In the experiments on the steel surface defects dataset NEU-DET, our model overall outperforms the current mainstream detection models. The model is capable of high-precision and low-computational-cost target detection, thus saving labor costs and improving public health and safety and productivity.

Keywords: Attention Mechanism, Lightweight, Neck Networks, Target Detection.

INTRODUCTION

Target detection is a key area of computer vision research, and its main task is to achieve automatic detection and localization of objects of interest in a given image or video. For the target detection stage, it can be categorized into single-stage and two-stage algorithm types. To improve the model detection performance, some network models such as VGG (Tammina, 2019), ResNet (Li and He, 2018) and EfficientNet (Alhichri *et al.*, 2021; Wang *et al.*, 2020b) have been widely used in backbone for feature extraction. These networks have multi-level convolution and pooling operations to learn richer representations of image features. Especially for some target detection models such as Faster R-CNN (Ren *et al.*, 2015; Fan *et al.*, 2016; Meng *et al.*, 2018; Zhang and Shen, 2021; Yang *et al.*, 2022; Wang *et al.*, 2018), the depth and width of the backbone are usually increased to improve the detection performance, but this may further increase the computational effort. However, since these deep networks have more parameters and complex structures, they require greater computational effort during training and inference. To address the problem of high computational resource requirements, networks such as MobileNet (Li *et al.*, 2018; Michele *et al.*, 2019) and Xception (Carreira *et al.*, 1998) use the convolution operation of DSC (Depthwise Separable Convolution) to reduce the complexity of the model. However, the relatively shallow network structure and reduced number of parameters may also limit the expressive power of the model, leading to relatively poor results on some more complex tasks. To enhance the robustness and generalization of the model, GhostConv (Ghost Convolution) (Cao *et al.*, 2022) divides the input channels of each convolutional layer into a main channel and a ghost channel, and uses the main channel for feature extraction and increases the number of output channels by performing blurring operations through the ghost channel. This design idea helps in areas such as lightweight network design and model compression. According to the analysis in the literature (Redmon and Farhadi, 2017), although the R-CNN family of models (evolving from R-CNN to Fast R-CNN and finally to Faster R-CNN) improves the processing efficiency by integrating the non-CNN processing part into the CNN and fully utilizing the efficient computational features of GPUs, the inference speed of Faster R-CNN is still limited to 5 frames on GPUs,i.e. Only five images are processed per second. This speed may be insufficient for real-time applications that require high processing speed. In contrast, single-stage models such as YOLO and SSD require less memory for inference due to their lower number of parameters and floating-point operations, which not only facilitates deployment on embedded devices, but also has a significant advantage in inference speed. the YOLO model can reach a processing speed of 45 fps, while the SSD500 model can process at 19 fps, which are significantly faster than Faster R-CNN. However, the single-stage target detection model tends to sacrifice the detection accuracy of the model due to this design by simplifying the model structure to reduce the computational burden. Therefore, in terms of accuracy, such models generally have a large disadvantage.

In the field of target detection, the reliance on expensive high-performance GPUs is avoided by reducing the model parameters and computation so that it can run on lower-cost hardware, such as low- to mid-range GPUs or CPUs. Not only that, for situations where surveillance cameras need to be deployed on a large scale in urban transportation systems or public safety projects, and in surveillance systems that need to run 24/7, a lightweight target detection model also reduces the risk of overheating or overloading of the device, which not only prolongs the life of the device, but also saves a lot of electricity and maintenance costs. Reducing model parameters and calculations therefore not only optimizes technical performance, but also brings significant economic benefits.

The accuracy and lightness of target detection models are crucial in autonomous driving technology, especially when dealing with fast-moving scenes and changing street environments. Accuracy ensures that the self-driving vehicle can accurately recognize and judge various objects in the surrounding environment, such as pedestrians, other vehicles, traffic signs, and obstacles. This is crucial to ensure that the vehicle is traveling safely. If the detection accuracy is not high, it may cause the vehicle to misjudge or miss hazards, thus increasing the risk of traffic accidents. However, lightness is equally important because autonomous driving systems need to make decisions in real-time dynamic environments, and this requires very fast processing speeds because excessive delays may affect the timeliness of decisions and may even lead to danger. Therefore, the target detection model not only needs to recognize objects accurately, but also must have high computational efficiency.

Therefore, current research focuses on how to obtain higher model accuracy with lower computational effort. Based on the above background, this paper proposes an improved lightweighting approach to optimize the Neck part of the target detection model YOLOv8 in order to reduce the computational effort and improve the performance. Neck networks play the role of adjusting and fusing feature maps in target detection. Through in-depth study of the properties and functions of Neck network, we propose a lightweight improvement method. The main innovations and contributions of this paper are as follows:

We improve the current existing methods and propose LDGConv (Lightweight-DepthGhost Convolution), a convolution module with fewer parameters than Ghost Convolution, and apply it to Neck networks.

We lighten and improve the C2f by incorporating the convolution proposed in this paper, and propose and use the more lightweight C2f_LDG. The experiments demonstrate the advantages of the lightweight convolution proposed in this paper in terms of accuracy and computation.

The Coordinate Attention mechanism (Hou *et al.*, 2021) is also added to the Neck network to achieve high efficiency and high accuracy detection performance in complex and dense scenes. And it is demonstrated in the VOC dataset that the model can obtain satisfactory results in terms of accuracy with low computational effort.

In order to verify the robustness and generalization ability of the improved model, we compare it with the original model as well as mainstream models using the steel surface defect dataset.

Through our research, we have made significant progress in the field of target detection in a computationally resource-constrained environment, which is of positive significance for the advancement of the field.

RELATED WORK

YOLO

The YOLO (You Only Look Once) series is the most widely used single-stage target detection algorithm today, and the constant version updates are the main reason for its leading position. The first version of this family, YOLO (Redmon *et al.*, 2016), was launched in 2016 and was trained using an end-to-end approach, where feature maps were extracted from the input images by a convolutional neural network, classification and regression were performed by a fully connected network, and the results were filtered using NMS (Non-Maximum Suppression) (Jiang *et al.*, 2019). Subsequently, the YOLOv2 (Redmon and Farhadi, 2017) algorithm was introduced in 2017, which adds a BN (Batch normalization) (Liu *et al.*, 2018) layer after the convolutional layer to increase the convergence speed and reduce overfitting, and uses the K-Means method for the prediction of the number of anchors. YOLOv3 (Tian *et al.*, 2019), on the other hand, uses Darknet-53 as the backbone network for feature extraction, and FPN (Feature Pyramid Network) (Zhu *et al.*, 2022) as the Neck part, which fuses high-resolution and low-resolution feature maps by up-sampling and residual concatenation, and the YOLO series has been using the Neck network to process and rationally utilize the feature maps extracted from different stages of the backbone network since YOLOv3. The CSPNet (Wang *et al.*, 2020a) network architecture, introduced in 2019, enhances the learning capabilities of CNNs and provides a lot of insight into improvements after the YOLO series. YOLOv4 (Gai *et al.*, 2023) and YOLOv5 (Wu *et al.*, 2021) are both available in 2020, the former combines Darknet53 with CSPNet as CSPDarknet53 to improve network efficiency and accuracy, while the latter improves on CSPDarknet53 and designs the C3 structure by borrowing from CSPNet in order for the model to learn more features.

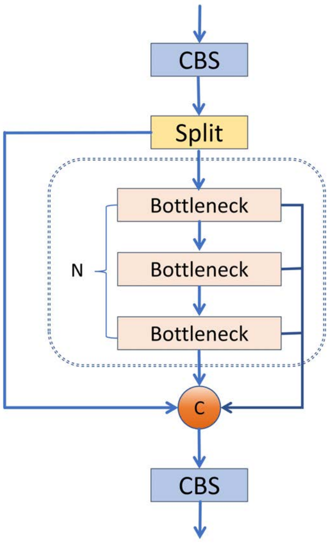


Fig. 1. *C2f structure of the YOLOv8 model*

YOLOv8, on the other hand, was proposed in 2023, combining the ideas of CSPNet and ELAN (Zhang *et al.*, 2022b) networks, and designing a lighter C2f structure to replace the previous C3 structure, the structure adopts the shuffling idea of CSPNet and the concept of residual structure to obtain richer information about the gradient flow and better utilize the upstream information. The number of stacked C2f structures is controlled by the parameter ”N”, which can be different for different scales of models. The C2f structure of the YOLOv8 model is shown in Fig. 1, which shows that the C2f module enriches the gradient flow of the model by connecting more branches across the layers. The Bottleneck module in the C2f structure is shown in Fig. 2, where CBS denotes the combination of ordinary convolution, batch normalization and SiLU activation function combinations.

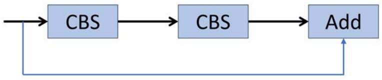


Fig. 2. *Bottleneck Module Structure Diagram*

Although the C2f structure enriches the gradient flow of the model and enhances the feature representation of the model through multi-branch cross-layer connections, its inclusion of a large number of convolution operations also leads to the accumulation of a large amount of redundant information, which may result in more parameter requirements and computational resource consumption. YOLOv5s-M (Ren *et al.*, 2023) proposed in 2023 that the method of utilizing increasing the size of the detection model by adding more detection heads provides a 3.9% improvement in mAP metrics compared to the original model, but the computation and parameters of the model increase to 2.52 and 5.19 times that of the original model, respectively, which further increases the model’s complexity and makes it difficult to be deployed in computationally resource-constrained devices. YOLOv8-GAM-Wise (Xiong *et al.*, 2024) was proposed in 2024, which incorporates Global Attention Mechanism (Zhou *et al.*, 2023) into the backbone network, Global Attention Mechanism can dynamically adjust the weights based on the importance of different parts of the input sequence, allowing the model to focus more on the information that is more important to the information that is more important for the current task. However, because the Global Attention Mechanism needs to weight the whole sequence, the computational complexity increases significantly when the sequence is long, resulting in the inference time of the model becoming 1.26 times of the original, not only that, because the backbone network feature map needs to be sent to the Neck network for a series of feature fusion, when the number of channels is reduced in the convolutional layer or the spatial dimension is reduced by the pooling layer, the features selected and weighted by the attention mechanism will be more important to the task. information of the features selected and weighted by the mechanism may be diluted, causing the model to improve the mAP metric by only 0.1% with a large increase in computational parameters and inference time. Therefore, the focus of the research in this paper is to construct a lightweight network to reduce the computational burden of the model, and to ensure that the model maintains accuracy with a certain degree of stability and robustness, in order to develop a lightweight model that is general and easy to deploy.

CONVOLUTIONAL OPERATIONS IN LIGHTWEIGHT MODE

CNN-based detection models are widely used in various aspects, for example, in the application of tribal dress recognition (Rabbi *et al.*, 2023), by extracting the spatial features and texture information in the dress image, the dress style features of different tribes are obtained, and these features enable the model to accurately recognize and differentiate the dresses of different tribes. While in the medical field (Tawfeeq *et al.*, 2021), Convolutional Neural Networks and heat maps are utilized to predict salient features in medical images, CNN-based models can learn to detailed features in the images and correlate these features with specific diseases or lesions. In the intelligent management of agriculture, CNN-based models also show important value, such as the literature (Mg *et al.*, 2025) proposes to use the improved YOLOv8 model to realize the accurate detection of cattle, combined with the customized tracking algorithm and the motion feature analysis, to construct a fully automated calving time prediction system, which significantly improves the automation level of livestock management. However, in some cases, models need to be deployed on multiple platforms, such as mobile devices, cloud servers, embedded systems, and so on. In these usage environments, lightweight convolution can significantly reduce the number of parameters, computation, and memory consumption of the model, allowing the model to run efficiently under constrained resources, thus reducing the cost of development and deployment.

Ghost Convolution and DWConv (Depthwise Convolution) are common and efficient lightweight convolution operations, and they are usually widely used in lightweight models.

The traditional convolution operation is to first convolve each channel of the input feature map with a convolution kernel separately, and then sum the results of all the channels to get an output feature map. If the number of output channels is 3, the above steps need to be repeated 3 times, as shown in Fig. 3.

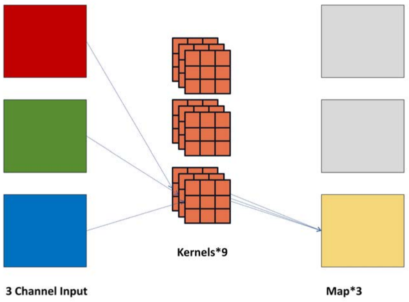


Fig. 3. *Conventional Convolutional Operations*

The Depthwise Convolution operation is shown in Fig. 4, which applies only one convolutional kernel to each output channel, i.e., there is only one corresponding convolutional kernel and input channel for each output channel. It allows the extraction of feature information for each channel and reduces the computational burden of the model by reducing a large amount of computation relative to the traditional convolution operation.

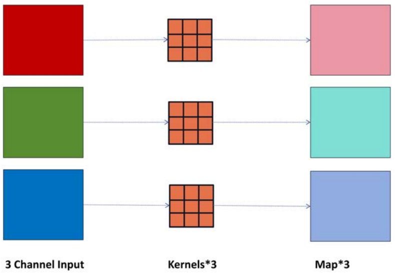


Fig. 4. *Depthwise Convolution Operation*

Ghost Convolution effectively reduces the number of parameters and computational complexity by dividing the input feature map into two parts, i.e., primary and secondary paths. As shown in Fig. 5, the primary path utilizes only a portion of the channels for ordinary convolution operation, while the secondary path uses an inexpensive Depthwise Convolution to linearly map the output feature maps of the primary path. Afterwards, the two are spliced to achieve the number of output channels, and finally the output feature map is obtained. Depthwise Convolution through auxiliary paths helps to increase the perceptual field and information transfer capability of the model. This in turn improves the nonlinear representation and generalization ability of the model.

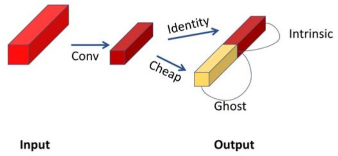


Fig. 5. *Ghost Convolution Operation*

Compare ordinary convolution operation to Depthwise Convolution and Ghost Convolution in terms of computation.

The operation of generating n feature maps in all convolutional layers can be expressed by Equation 1 as.

$$Y = X \times f + b \tag{1}$$

where $X \in R^{c*h*w}$ denotes the input to the convolution, c denotes the number of input channels, h and w denote the height and width of the input feature map, respectively, $Y \in R^{c'*h'*w'}$ and there are n channels to output the feature map, h' and w' are the height and width of the output feature map, and f is the convolution filter of the layer, which has a kernel size $a*a$, and b is the bias term.

The ordinary convolution operation is computed as:

$$n \cdot h' \cdot w' \cdot c \cdot a \cdot a \tag{2}$$

The Depthwise Convolution operation is computed as:

$$n \cdot h' \cdot w' \cdot 1 \cdot a \cdot a \tag{3}$$

Therefore, the computational effort of the Depthwise Convolution operation is about $1/c$ that of the ordinary convolution.

Let the size of the linear arithmetic kernel be $r*r$, and each basic feature corresponds to s redundant features, $c \gg s$. Given that the original method obtains m feature maps, and that there is constancy in the transformation process of Ghost Module, the actual effective transformations are:

$$m \cdot (s-1) = \frac{n}{s} \cdot (s-1) \tag{4}$$

So the calculation for Ghost Convolution is:

$$\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot a \cdot a + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot r \cdot r \tag{5}$$

Dividing the computation of the ordinary convolution operation by the computation of the Ghost Convolution operation.

$$\frac{n \cdot h' \cdot w' \cdot c \cdot a \cdot a}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot a \cdot a + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot r \cdot r} \approx s \tag{6}$$

Therefore, the computational effort of the Ghost Convolution operation is about $1/s$ of the ordinary convolution operation.

ATTENTION MECHANISM

Single-stage algorithms have the advantage of fast detection, but how to improve accuracy while maintaining speed is a key issue in target detection research. In computer vision, the attention mechanism plays an important role. The attention mechanism was first applied to RNN (Mnih *et al.*, 2014), which can help the model to focus on regions or features related to the target, thus improving the accuracy and efficiency of detection. In the target detection task, the attention mechanism makes the model more accurate and robust by weighting and conditioning the important regions and features in the image.

In the current construction of lightweight networks, traditional attention mechanisms such as the SE module (Humphreys and Sui, 2016) and Channel Attention mechanism (Bastidas and Tang, 2019) mainly focus on inter-channel information, but ignore the importance of spatial location information. Although the subsequently developed CBAM (Fu *et al.*, 2021) captures positional attention information by using convolution after reducing the number of channels, this approach can only deal with local relationships and lacks the ability to capture long-distance spatial relationships. Although DANet (Dual Attention Network) (Li *et al.*, 2020) improves the accuracy of feature representation by integrating local features and global dependencies through its unique positional attention module and channel attention module, its excessive complexity leads to significant limitations on real-time processing and resource-constrained devices as well, and thus is not applicable to lightweight networks.

In order to solve the above problems, researchers have proposed the Coordinate Attention mechanism (Hou *et al.*, 2021).The introduction of the Coordinate Attention mechanism is particularly suitable for lightweight networks because it significantly improves the performance by optimizing the parameters and computational complexity without significantly increasing the computational burden. This mechanism is able to extract more representative feature information by finely weighting the channel dimensions of the input feature map. This mechanism not only focuses on inter-channel information, but also incorporates spatial features by assigning weights to each location that reflect the importance of each location in the task at hand, and is used to weight the input sequence so that the model can more accurately focus on key locations.The Coordinate Attention mechanism provides a solution for lightweight networks to significantly improve spatial perception while maintaining efficiency. significantly improving spatial perception. Therefore, in this paper, the Coordinate Attention mechanism is used to weight and adjust the channel dimensions of the input feature map to extract more representative feature information.

The Coordinate Attention mechanism module enhances the network's ability to learn features by transforming an intermediate feature tensor X and later outputting a tensor Y of the same size.

$$X = [x_1, x_2, \ldots, x_c] \in R^{H \times W \times C} \tag{7}$$

$$Y = [y_1, y_2, \ldots, y_c] \in R^{H \times W \times C} \tag{8}$$

The implementation process of the Coordinate Attention mechanism is shown in Fig. 6.

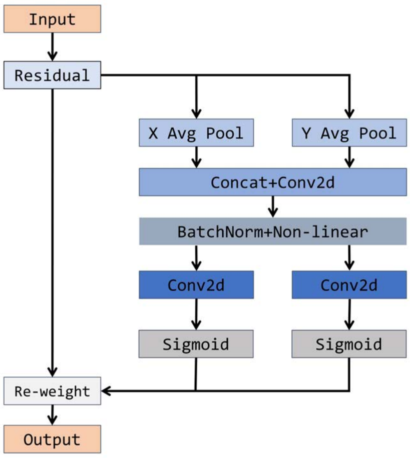


Fig. 6. *Coordinate Attention Mechanism Structure Diagram*

The Coordinate Attention mechanism first performs global average pooling of the input feature maps in the width and height directions, respectively, to obtain the feature maps in the corresponding directions. At the same time, the mechanism also encodes the location information as shown in Equation 9 and Equation 10.

$$z_c^h(h) = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i) \tag{9}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \leq j < H} x_c(j, w) \tag{10}$$

Then the feature maps in both directions of width and height of the obtained global receptive field are spliced together, after which they are fed into the convolution module with a shared convolution kernel of 1*1 to make their dimensionality reduced, and then the batch normalized feature map F_1 is fed into the Sigmoid activation function to obtain the feature map f, as shown in Equation 11.

$$f = \delta\left(F_1\left(\left[z^h, z^w\right]\right)\right) \tag{11}$$

Then the feature map f is sliced into f^h and f^w along the height and width, and then two 1*1 convolution kernels F_h and F_w are used to convolve f^h and f^w respectively to obtain a tensor with the same number of channels as the input X. Finally after Sigmoid activation function the attention weights α^h and α^w are obtained for the feature map in height and width respectively, as shown in Equation 12 and Equation 13.

$$\alpha^h = \sigma\left(F_h\left(f^h\right)\right) \tag{12}$$

$$\alpha^w = \sigma\left(F_w\left(f^w\right)\right) \tag{13}$$

The obtained attentional weights α^h and α^w will be calculated by multiplicative weighting, and finally the feature map with attentional weights in both width and height directions will be obtained as shown in Equation 14.

$$y_c(i, j) = x_c(i, j) \times \alpha_c^h(i) \times \alpha_c^w(j) \tag{14}$$

METHOD

IMPROVE NETWORK STRUCTURE WITH YOLOV8N

The C2f structure in YOLOv8 plays a key role in feature enhancement and information fusion in the network, but it also poses some problems. The feature map information output from the YOLOv8 Neck network contains features at multiple resolutions, and the C2f structure enhances the feature representation of the model, which helps to better capture multi-scale information about the target. However, this also means

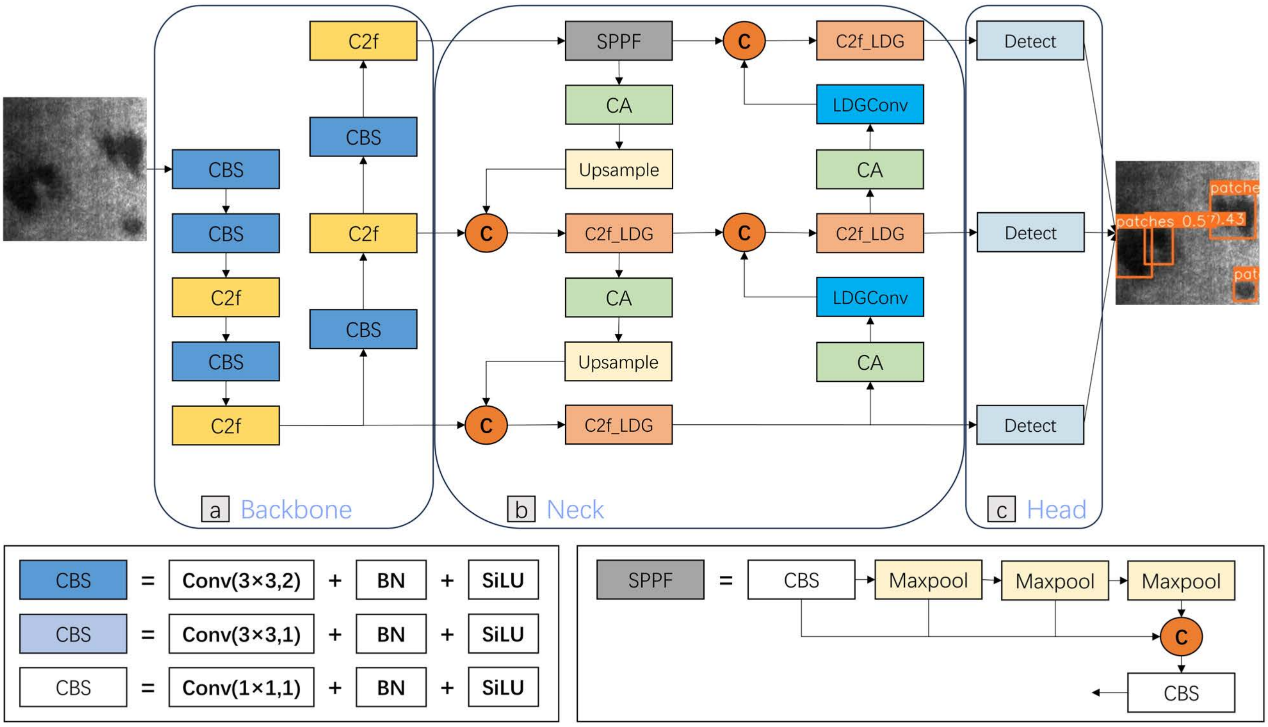


Fig. 7. *YOLOv8n-improve model structure*

that it performs a large number of standard convolution operations on feature maps of different resolutions. This leads to feature redundancy as the information in the feature maps is somehow extracted and represented repeatedly.

Although the C2f module improves the accuracy of the algorithm, it accumulates a large amount of redundant information while increasing the feature expressiveness, which may lead to more parameter requirements and computational resource consumption. This may limit the deployment and application of the model in resource-constrained environments. To alleviate these issues, this study proposes an improved network structure based on YOLOv8n. The specific improvement is shown in Fig. 7.

As shown in Fig. 7, the image is first fed into the model and processed through the three main parts of the network: the Trunk (solid box a), the Neck (solid box b) and the Head (solid box c). The Trunk starts with two convolution operations using a 3×3 kernel with a step size of 2. After several convolution operations and a C2f feature enhancement module, the feature map reaches the Neck part. In the Neck part, after the SPPF spatial pyramid pooling module, we first introduce the Coordinate Attention mechanism before the up-sampling phase for feature fusion of different resolution feature maps in the Neck network and before the down-sampling phase.

Coordinate Attention mechanism By focusing more precisely on important locations or regions, the model can reduce the processing of unnecessary information, thus reducing redundant feature information and improving efficiency and performance. To further reduce the model complexity, we use the Lightweight-DepthGhost Convolution module proposed in this paper for the downsampling part in the Neck network. The feature map will fuse feature information of different sizes after upsampling as well as after downsampling. After feature fusion, feature enhancement is performed using the C2f_LDG module proposed in this paper. Finally the feature map fused with different levels of feature information is fed into the Head part of the network. In the Head part, three Detect modules process and predict the feature maps of different sizes in order to capture the detection targets of different sizes to obtain the final detection map.

MORE LIGHTWEIGHT CONVOLUTION OPERATIONS

In order to alleviate the large amount of feature redundancy caused by repeated extraction and representation of information in the feature map in Neck networks, which makes the model need to take up more parameters and computational resources, this paper proposes an effective convolution method that further combines Ghost Convolution with Depthwise Convolution, namely LDGConv (Lightweight-DepthGhost Convolution) module. Depthwise Convolution is a deep convolution operation that can effectively capture the spatial features of the input data while the number of parameters is small, and its main focus is on the feature transformations within the input channel. Ghost Convolution divides the input feature map into two parts, and without changing the size of the output feature mapping, the total number of parameters required and the computational complexity have been reduced. Therefore, we combine Depthwise Convolution and Ghost Convolution to further utilize the lightweight feature of Depthwise Convolution and the feature utilization of Ghost Convolution, and use low-cost linear transformation to generate many feature maps that can fully reveal the intrinsic feature information, in order to reduce the feature redundancy. Finally, channel shuffle operation is used for interaction and information transfer between different parts. This increases the expressive power of the model and enables it to better capture relevant features in the input data.

The specific steps of the Lightweight-DepthGhost Convolution operation are as follows.

First use half of the channels for the Ghost Convolution convolution operation, the computation of this part is:

$$\frac{1}{s}\cdot\frac{n}{2}\cdot h'\cdot w'\cdot c\cdot a\cdot a \tag{15}$$

where c denotes the number of input channels, n denotes the number of output channels, h' and w' denote the height and width of the output feature map, and the kernel size is $a*a$. s is the number of redundant features corresponding to each basic feature, and $c \gg s$.

Using the other half of the output channel the output of Ghost Convolution is subjected to the Depthwise Convolution operation, the computation of this part is:

$$\frac{n}{2}\cdot h'\cdot w'\cdot 1\cdot a\cdot a \tag{16}$$

So the computation of Lightweight-DepthGhost Convolution operation is:

$$\frac{n}{2}\cdot h'\cdot w'\cdot\left(\frac{c+s}{s}\right)\cdot a\cdot a \tag{17}$$

Because of $c \gg s$, the computation of ordinary convolution operation is close to $2*s$ times the computation of Lightweight-DepthGhost Convolution operation.

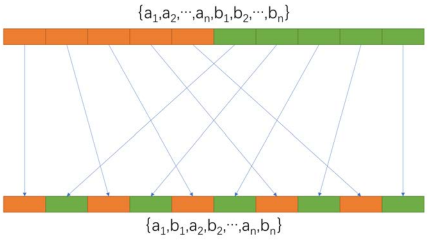

$\{a_1,a_2,\cdots,a_n,b_1,b_2,\cdots,b_n\}$
$\{a_1,b_1,a_2,b_2,\cdots,a_n,b_n\}$

Fig. 8. *Channel shuffle operation*

The two results of the deep convolution of Ghost Convolution and Depthwise Convolution are then spliced together to ensure that the two branches exchange information. Finally, the channel shuffle operation is performed, the splicing result is first divided into two parts according to the number of channels, and the channels of the two parts are interleaved in order, as shown in Fig. 8. The richness of the features and the expressive power of the model can be improved by the shuffle operation, which allows the model to mix information more efficiently between successive layers and ensures the exchange of information across the feature channels. The structure of the Lightweight-DepthGhost Convolution module is shown in Fig. 9.

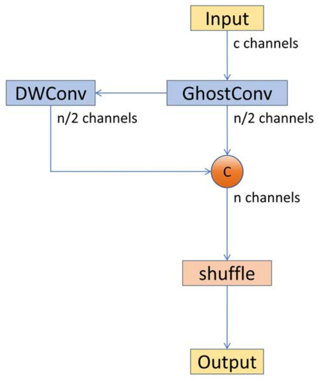


Fig. 9. *Lightweight-DepthGhost Convolution structure diagram*

Lightweight-DepthGhost Convolution generates a number of intrinsic feature mappings in a cost-effective manner and then utilizes linear operations to increase the number of channels. This approach further reduces the model volume and reveals deeper information about the intrinsic features. Finally, interaction and information transfer between different parts is facilitated by using channel shuffle operations to enhance the expressive power of the model.

BOTTLENECK MODULE IMPROVEMENT

The C2f structure of YOLOv8 serves as a feature enhancement and information fusion in the network in order to better capture the multi-scale information of the target. The output feature map information contains feature map information at multiple resolutions, which enhances the feature representation of the model. By extracting features at different resolutions, the model can more comprehensively capture relevant features in the input data, thus improving target recognition and understanding. Although the C2f module, improves the accuracy of the algorithm, it contains a large number of standard convolutions, resulting in increased model parameters and complexity.

In order to further lighten the model, the Lightweight-DepthGhost Convolution module proposed above is combined with Bottleneck. The structure of the LDG Bottleneck module is shown in Fig. 10, which consists of two Lightweight-DepthGhost Convolution modules stacked together for feature mapping processing in the network. The first Lightweight-DepthGhost Convolution module is used as an extension layer to increase the number of channels, and the second Lightweight-DepthGhost Convolution module is used to reduce the number of channels to match the number of input channels. Finally, the shortcut function is implemented in order to speed up the convergence of the model and improve the accuracy of the model through the use of residual concatenation.

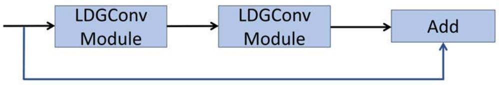


Fig. 10. *LDG Bottleneck Module Structure*

The C2f_LDG structure is shown in Fig. 11, where the LDG Bottleneck replaces the Bottleneck module in the original C2f structure so that the model has enough information to understand the input data. This improvement reduces the redundant information without increasing the network parameters and effectively improves the speed of the model to understand the image information. The detection accuracy of the network is guaranteed while realizing the light weight of the network.

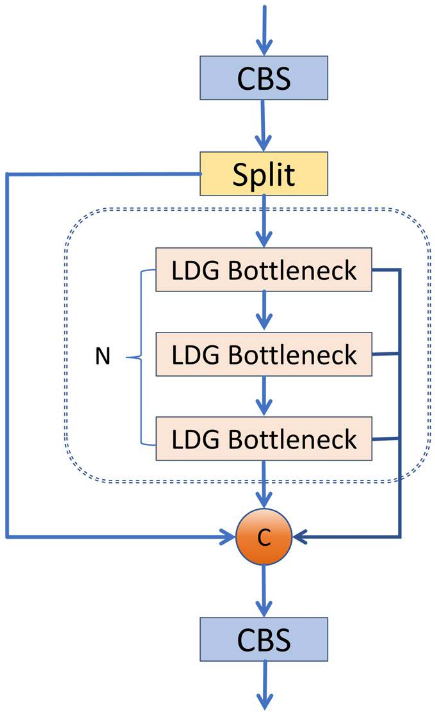


Fig. 11. *C2f_LDG Structure Diagram*

EXPERIMENT AND RESULT ANALYSIS

EXPERIMENT ENVIRONMENT

Table 1 describes the software and hardware environments used for model training and testing.

Table 1. *Experiment environment*

Hardware and Software	Models and Versions
CPU	AMD R5 5600
GPU	NVIDIA GeForce RTX 3060Ti
OS	Windows 10
Development Language	Python 3.8.17
Deep Learning Framework	PyTorch 1.12.0
Image size	640*640
Optimizer	Adam
Initial learning rate	0.01
Final learning rate	0.01
Optimizer weight decay	0.0005
Warmup epochs	3.0
Warmup initial momentum	0.8
Warmup initial bias lr	0.1
Momentum	0.937

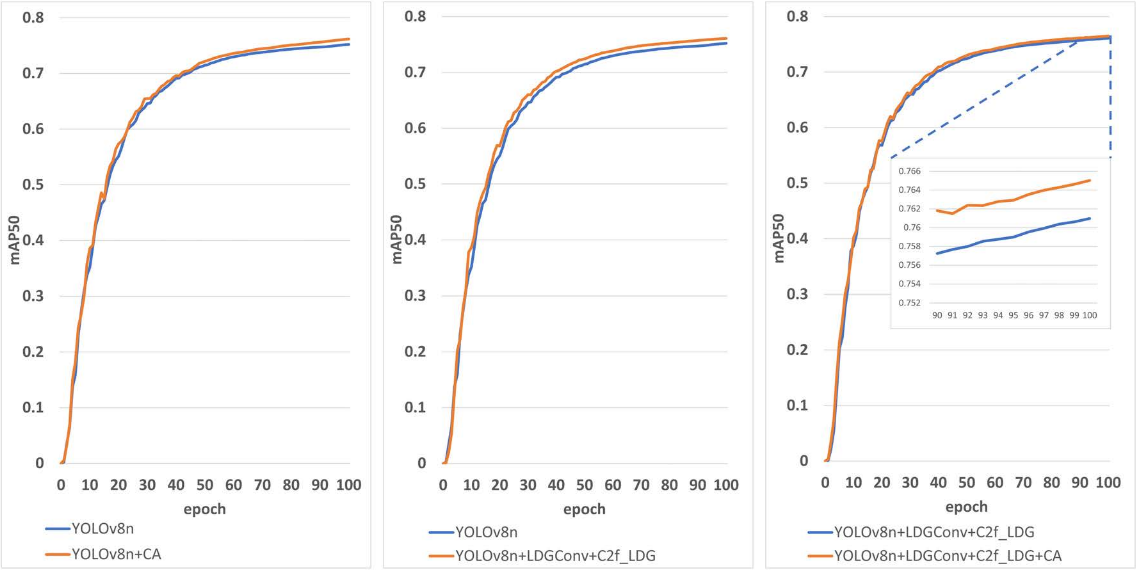
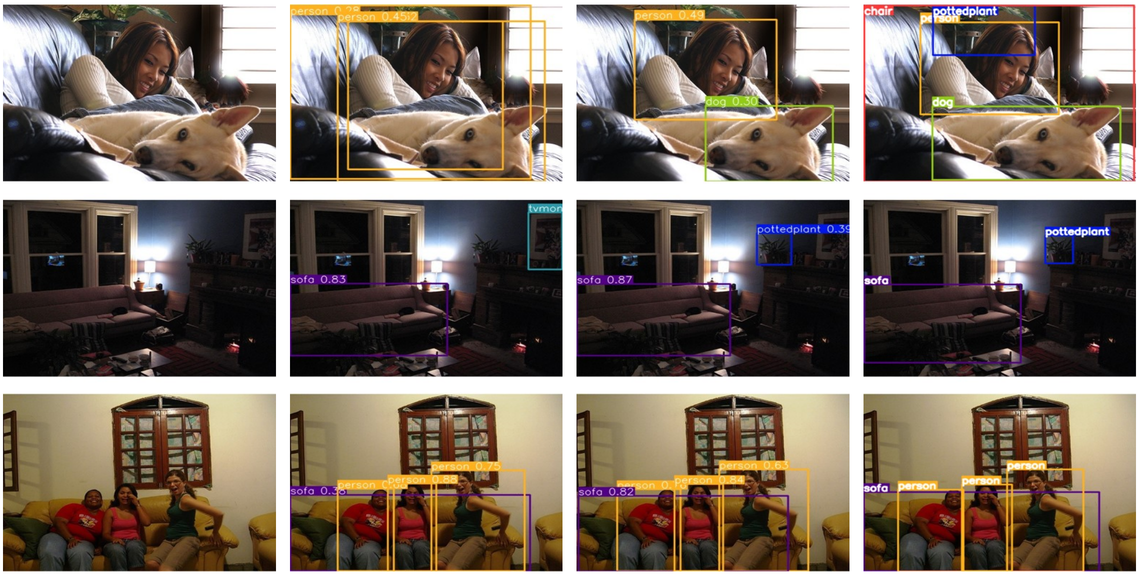


Fig. 12. *Comparison of the trend of mAP before and after adding the module*



(a) Original Image (b) YOLOv8n (c) YOLOv8n-improve (d) Ground Truth

Fig. 13. *Detection effect of VOC dataset*

A COMPARATIVE ANALYSIS OF ABLATION EXPERIMENTS

The ablation experiment dataset used in this paper is Pascal VOC 2007+2012, which has 21503 images that can be categorized into 20 classes, including people, TV, sofa, car, cat, etc. The ”Pascal VOC 2007 train+val” and all the ”Pascal VOC 2012” images are used as the training set. Among them, ”Pascal VOC 2007 train+val” and all ”Pascal VOC 2012” images are used as the training set, which consists of a total of 16551 images, and the rest of the images are used as the training set. ”Pascal VOC 2007 test” 4952 images are used as test set and validation set. In this paper on the Pascal VOC2007+2012 dataset, the hyperparameters were set to epoch 100 and batch size 32 during training.

Through ablation experiments, it is verified that the application of the Lightweight-DepthGhost Convolution module to the YOLO model of the Neck network reduces a large number of parameters and computations and guarantees model accuracy. The metrics of mAP, parameter size, GFLOP (gigafloating point operations per second), and model size are compared in the same experimental setup, and F_1

scores are introduced to further evaluate the model's precision and recall in a comprehensive manner, as shown in Equation 18.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{18}$$

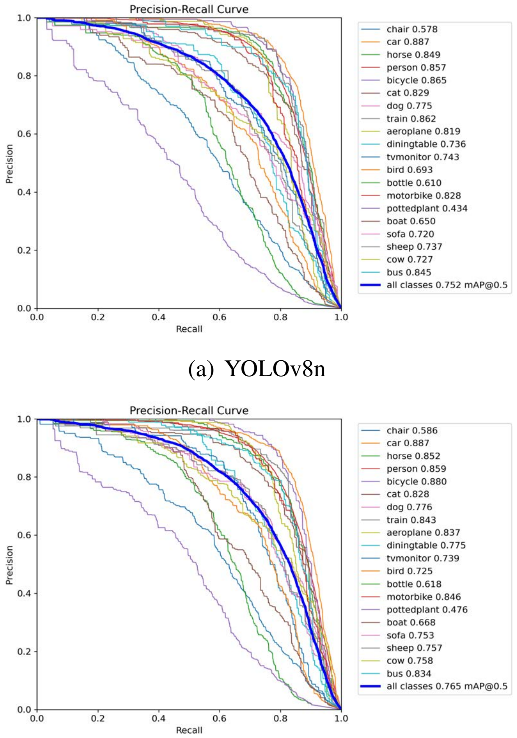
The results of the experiment are shown in Table 2.

Table 2. *Ablation Experiment Results*

LDGConv	CA	C2f_LDG	mAP50(%)	Parameters(M)	GFLOPs	F_1(%)	Model size(MB)
			75.2	3.01	8.2	71.3	5.94
√			75.6	2.88	8.0	71.8	5.70
	√		76.2	3.02	8.2	71.6	5.99
		√	75.6	2.67	7.5	71.9	5.34
√		√	76.1	2.54	7.4	72.3	5.09
√	√	√	76.5	2.55	7.4	72.5	5.14

A comparison of the trend of the evolution of the mAP indicator with epoch before and after adding the module is shown in Fig. 12.

The detection effect in the VOC dataset is shown in Fig. 13, where (a) is the original image, (b) is the detection effect of YOLOv8n, (c) is the detection effect of the improved model YOLOv8n-improve proposed in this paper, and (d) shows the ground truth sample box. The Precision-Recall curves for each class and the Average Precision for each class are shown in Fig. 14, where (a) is the Precision-Recall curve of the original YOLOv8n, and (b) is the Precision-Recall curve of the improved model YOLOv8n-improve.



(a) YOLOv8n

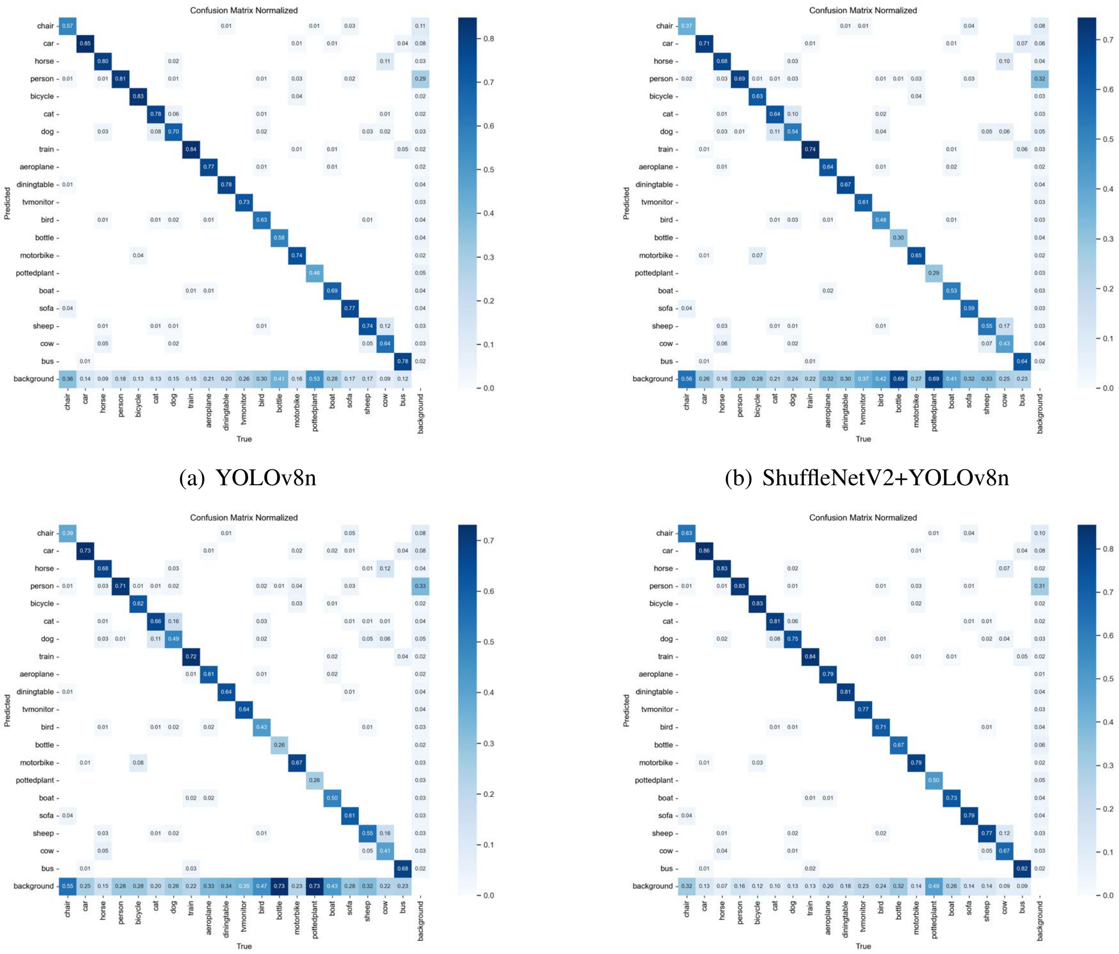
(b) YOLOv8n-improve

Fig. 14. *PR curves for target detection models*

From Table 2, it can be seen that the improved algorithm adopts a more efficient network structure for YOLOv8n, which improves the mAP and F_1 scores to a certain extent, and reduces the number of parameters and computation of the model. It is also demonstrated that the C2f_LDG module and Lightweight-DepthGhost Convolution effectively reduce the number of parameters and computation of the model without reducing the accuracy of the algorithm. The introduction of Coordinate Attention mechanism only increases a small number of parameters, which effectively improves the detection accuracy. Combining the above improvements with the YOLOv8n algorithm can minimize the model volume, with only 2.55M model parameters and 7.4G computation, which are reduced by 15.3% and 9.8%, respectively. The weight file size of the model is reduced by 13.4%.

As can be seen from Fig. 12, the addition of Lightweight-DepthGhost Convolution and C2_LDG modules in the Neck network makes the model reduce the computational amount while the accuracy is steadily improved, which further proves the effectiveness of the method, which not only reduces the redundant features, but also promotes the interaction and information transfer between the different parts, and improves the transfer effect of the features in order to improve the stability of the model. The introduction of Coordinate Attention in the original Neck network reduces information loss and better establishes spatial relationships, which enables the network to better understand the image content and significantly improves the detection performance. While the addition of Lightweight-DepthGhost Convolution and C2f_LDG lightweight module to lighten the Neck network, in the case of sparser relevant features, makes Coordinate Attention capture less spatial structure and correlation information between features in the input data, leading to a certain degree of reduction in the magnitude of the enhancement to the performance of the model, nevertheless, the Coordinate Attention effectively improves the detection efficiency and performance by reducing the processing of low-value information while occupying very few parameters.

Fig. 13 shows that accurate target detection for occluded targets or images with dimly lit scenes is a challenging task due to the fact that a portion of the target features are difficult to obtain, and the improved model in this paper can make better detection results in some of these types of scenes while consuming less computational resources than the original model, reflecting the feasibility of the proposed method.



(a) YOLOv8n

(b) ShuffleNetV2+YOLOv8n

(c) EfficientNetV2+YOLOv8n

(d) YOLOv8n+LDGConv+C2f_LDG

Fig. 15. *Confusion matrix for detection models*

As shown in Fig. 14, the experimental results indicate that the improved YOLOv8n-improve model exhibits higher detection accuracy in most target categories, especially in the recognition tasks of occlusion-susceptible objects (e.g., diningtable, sofa, motorbike) and small-scale targets (e.g., bird, pottedplant), and its average precision (Average Precision, AP) is significantly improved compared to the baseline model. This phenomenon fully verifies the effectiveness of the proposed optimization strategy in complex scenarios, and indicates that the algorithmic improvement significantly enhances the model's ability to capture occlusion-sensitive targets and small-scale features, which provides theoretical support for improving the robustness of the target detection system.

COMPARATIVE EXPERIMENTS ON LIGHTWEIGHT MODELS

To further validate the performance of the lightweight model proposed in this paper, we use other lightweight convolutional modules added to the YOLOv8n model for comparison in the VOC dataset. In our experiments we compare Ghost Convolution as a convolutional operation for downsampling in Neck networks with Lightweight-DepthGhost Convolution operation as a convolutional operation for downsampling in Neck networks, and at the same time we perform experiments comparing GSConv+VoVGSCSP (Zhao and Song, 2023) with LDGConv+C2f_LDG. In order to test the effect of the lightweight module on the overall model, we replace the downsampling convolution and feature enhancement modules in the backbone network with the lightweight convolution and lightweight feature enhancement modules based on the previous ones. The experimental results are shown in Table 3. And we also use ShuffleNetV2 (Ma *et al.*, 2018), EfficientNetV2 (Tan and Le, 2021) as a Backbone network for YOLOv8n model in order to build a lightweight target detection model to compare with the method in this paper, and the experimental results are shown in Table 4. Fig. 15 shows the confusion matrix for each model in Table 4, where (a) is YOLOv8n, (b) is ShuffleNetV2+YOLOv8n, (c) is EfficientNetV2+YOLOv8n, and (d) is YOLOv8n+LDGConv+C2f_LDG.

Table 3. *Lightweight module comparison experiment*

Model	Backbone	Neck	mAP50(%)	Parameters (M)	GFLOPs
YOLOv8n			75.2	3.01	8.2
YOLOv8n+GhostConv		√	75.5	2.92	8.1
	√	√	74.9	2.73	7.7
YOLOv8n+LDGConv		√	75.6	2.88	8.0
	√	√	74.6	2.59	7.4
YOLOv8n+GSConv+VoVGSCSP		√	75.9	2.81	7.4
	√	√	73	2.53	6.1
YOLOv8n+LDGConv+C2f_LDG		√	**76.1**	2.54	7.4
	√	√	72.5	**1.95**	**6**

Table 4. *Comparison of lightweight detection models*

Model	mAP50(%)	Parameters(M)	GFLOPs
YOLOv8n	75.2	3.01	8.2
ShuffleNetV2+YOLOv8n	61.2	1.83	5.1
EfficientNetV2+YOLOv8n	60.9	2.13	2.6
YOLOv8n+LDGConv+C2f_LDG	76.1	2.54	7.4

As can be seen from Table 3, the reasonable use of lightweight convolution in the model Neck network can not only further lighten the model, but also improve the accuracy to some extent. This is because Neck networks are usually used to process features from different levels and scales, and synthesize them to form a more semantically informative feature representation. Lightweight modules have the flexibility to perform feature selection and combination in this process to accommodate features from different scales. This flexibility allows the lightweight module to better adapt to multi-scale target detection tasks while maintaining a certain level of accuracy. As for the lightweight improvement of the Model Neck network, the Lightweight-DepthGhost Convolution and C2f_LDG lightweight modules proposed in this paper have better performance on the VOC dataset compared to Ghost Convolution and GSConv+VoVGSCSP.

From the comparison data in Table 4, YOLOv8n+LDGConv+C2f_LDG achieves a significant increase in mAP50 while maintaining a lower number of parameters, which proves the effectiveness of the lightweighting improvement of the Neck network in this paper; The other two comparison models, although significantly compressing the parameters through EfficientNetV2 and ShuffleNetV2 lightweight backbones, respectively, also lead to a serious drop in the mAP50 index, reflecting the serious degradation of feature extraction ability due to excessive lightweight backbones, especially in complex scenarios with serious semantic information loss, highlighting the devastating impact of simply compressing backbones on detection accuracy's destructive impact.

As seen from the confusion matrix comparison in Fig. 15, the improved algorithm in this paper significantly improves the detection accuracy in most of the key categories by optimizing the Neck network structure, e.g., the normalized accuracies of the categories of car, bird, and aeroplane are higher than those of the other models. While the lightweight Backbone models EfficientNetV2+YOLOv8n and ShuffleNetV2+YOLOv8n significantly increase the false detection rate in complex and smaller targets, such as pottedplant , bottle , person . It shows that over-compression of the backbone network leads to degradation of feature extraction capability. The improved algorithm mitigates the accuracy loss problem by balancing Neck optimization and lightweight design while maintaining efficient inference.

To summarize, in the feature extraction stage, in order to improve the prediction speed, the CNN model usually needs to transform the input image step by step through the Backbone layer to transfer the spatial information to the channel dimension gradually. However, each compression of the spatial feature map and channel expansion may lead to the loss of some semantic information. In this case, the dense convolution operation maximally preserves the connections hidden between each channel, while the lightweight convolution completely cuts off these connections, which will further lead to feature information loss. Therefore, a more appropriate approach is usually to use lightweight convolution only in Neck networks, which can reduce the computational burden while adapting more flexibly to multi-scale detection tasks, thus improving the performance and efficiency of the model.

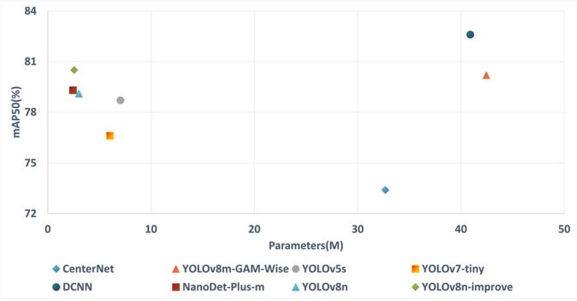
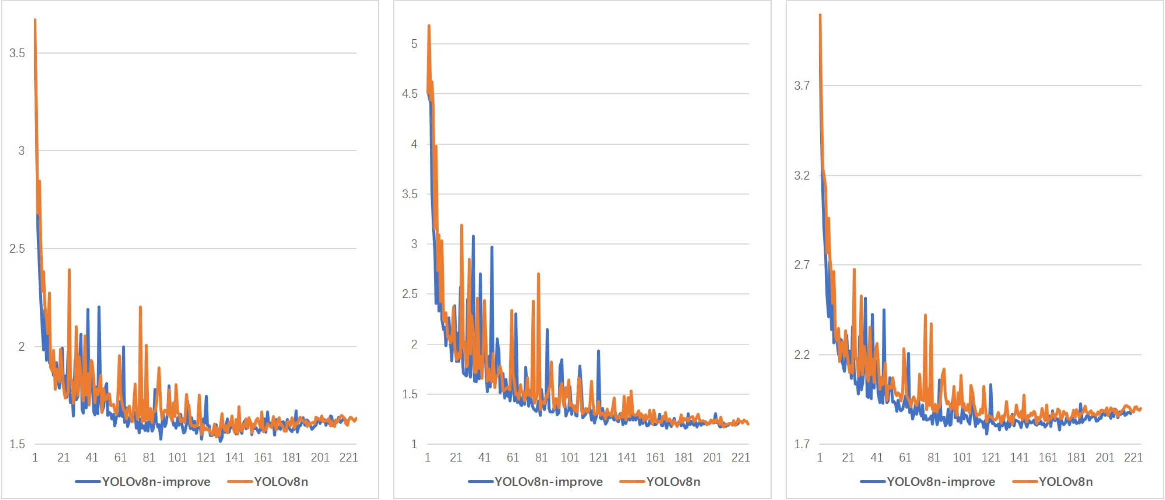


Fig. 16. *Scatterplot of the model's parameters and mAP metrics*

COMPARATIVE EXPERIMENTS WITH DIFFERENT MODELS

Target detection is one of the important applications in the field of computer vision, which can accurately identify and locate objects in images, reduce labor costs, and have a wide range of applications. For example, it is important in the field of public safety, industry and agriculture. However, there



(a) box_loss (b) cls_loss (c) dfl_loss

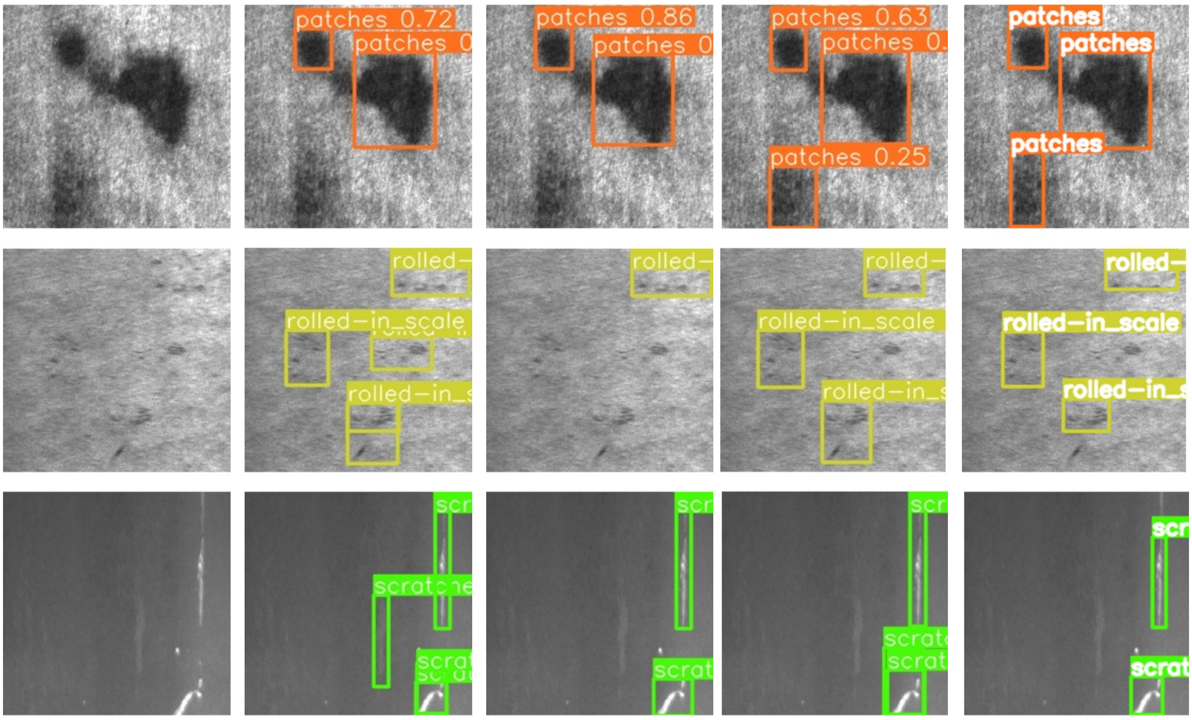
Fig. 17. *Comparison of Loss value in validation phase before and after improvement*

are various problems that need to be solved in complex scenes, such as a large number of objects, different backgrounds and lighting conditions, occlusion and overlapping, etc., so it is a challenging task to realize the detection of complex scenes. The accuracy of the YOLOv8n-improved model has been validated on the VOC dataset, and in order to further validate the robustness of the improved model proposed in this paper, experiments are conducted on the publicly available steel surface defects dataset NEU-DET as its application to complex scenarios, whose diverse crack shapes and sizes, as well as the existence of different lighting conditions in real production environments are part of the reasons for its use as a complex scenario. The dataset contains six main types of defects on steel surfaces. The respective defects are Crazing, Inclusion, Patches, Pitted_surface, Rolled_in_Scale and Scratches. There are 300 images for each defect, totaling 1800 images. The dataset is divided into training set, validation set and test set in the ratio of 8:1:1. We trained on this dataset with epoch set to 300 and batch size set to 16. In addition to this, current state-of-the-art target detection algorithms such as YOLOv8-GAM-Wise (Xiong *et al.*, 2024), YOLOv5s, YOLOv7-tiny, YOLOv8n, NanoDet-Plus-m (Huan *et al.*, 2024), and DCNN (Zhang *et al.*, 2022a) algorithm for surface defect detection, as well as CenterNet (Nazir *et al.*, 2021) based on Anchor-Free, are compared to the improved model proposed in this paper on the NEU-DET dataset Comparison experiments are conducted, and the experimental results are shown in Table 5, where the FPS metric is calculated based on the inference time per image, which reflects the average number of inference images per second. In order to make the data more intuitive, we plotted the scatter plots of the parameters and mAP metrics of each model as shown in Fig. 16, and the comparison plots of the box loss, classification loss and distribution focus loss between the improved model proposed in this paper and the original model in each epoch verification process are shown in Fig. 17. The detection effect of the model in the NEU-DET dataset is shown in Fig. 18, where (a) is the original image, (b) is the detection effect of YOLOv8n, (c) is the detection effect of YOLOv8-GAM-Wise, (d) is the detection effect of YOLOv8n-improve, and (e) shows the ground truth sample box.

Table 5. *Comparative Experimental Results of Different Models*

Model	mAP50(%)	Parameters(M)	GFLOPs	F_1(%)	FPS
CenterNet	73.4	32.67	70.2	71.5	45.9
YOLOv8-GAM-Wise	80.2	42.45	92.4	75	54.7
YOLOv5s	78.7	7.03	15.8	74.1	75.6
YOLOv7-tiny	76.6	6.02	13.1	73.8	87.2
DCNN	82.6	40.9	89.8	78.2	63.1
NanoDet-Plus-m	79.3	2.44	3	74.6	124.5
YOLOv8n	79.1	3.01	8.2	74.3	112.4
YOLOv8n-improve	80.5	2.55	7.4	74.5	117.6

As can be seen from the experimental results in Table 5, in terms of steel surface defect detection, the convolutional neural network (CNN) is able to extract a lot of useful features on the picture of steel surface defects, which play an effective role in promoting the detection model to accurately identify and localize the defects. From Fig. 16, it can be seen that the improved model in this paper has certain advantages over some mainstream target detection models in terms of model parameters and mAP indexes combined performance, and the improved algorithm proposed in this paper is better than the original model in terms of model size and inference speed. Although YOLOv8-GAM-Wise has good detection accuracy on the steel surface defects dataset, it is based



(a) Original Image (b) YOLOv8n (c) YOLOv8-GAM (d) YOLOv8n-improve (e) Ground Truth

Fig. 18. *Effectiveness of NEU-DET defect detection*

on YOLOv8m as the baseline model and introduces the Global Attention Mechanism for improvement, and the complex model makes it underperform in inference speed. Although the improved model in this paper falls short of NanoDet-Plus-m in terms of model size and inference speed, the mAP metrics are improved by 1.4% and 1.2% compared to the original model of YOLOv8n and NanoDet-Plus-m, respectively, and the FPS metrics of the three reach more than 100, which can satisfy the real-time demand of most industrial applications.

Compared with the Anchor-Free based CenterNet, the proposed model in this paper has more significant advantages in steel surface defect detection, increasing the mAP by 7.1%, reducing the amount of model parameters by 30.12M, which is only 7.8% of that of CenterNet, and decreasing the computation amount of the model by 62.8GFLOPs, which is only 10.5% of that of CenterNet. Although the mAP metric of DCNN has a 2.1% improvement compared to YOLOv8n-improve, the model uses a dense cross-stage partial Darknet backbone network for feature extraction resulting in a larger computational resource requirement compared to the lightweight model. requirements compared to the lightweight model, and at the same time the inference speed is difficult to meet the real-time performance. Fig. 17 shows that compared with the original YOLOv8n model, the YOLOv8n-improve model fits faster, performs well, and is stable, which further proves that the model improvement in this study is effective. Fig. 18 shows that in the picture of steel surface defects, the influence of various texture shapes, low contrast between the background and defects, and uneven brightness causes the algorithmic model to be difficult to detect the defect types quickly and accurately, and the method proposed in this paper mitigates the adverse effect on the inspection task caused by this problem to a certain extent, and it can effectively reduce the rate of rejects in the production process.

In summary, the lightweight model proposed in this paper has the advantages of small model size and low computation under the guarantee of detection accuracy, which makes it easy to be deployed on some terminal detection devices with weak computational capabilities and limited computational resources.

DISCUSSION

Currently, deep learning based object detectors have achieved great success in the field of target detection. Among them, Transformer-based and CNNs-based approaches are the two main technical tools. Transformer-based detectors are able to capture global contextual information by introducing an attention mechanism, which improves the recognition ability of detectors for complex scenes and occlusions.

However, the high computational complexity of the Transformer model leads to certain latency problems in real-time applications. In contrast, CNN-based detectors are still the preferred choice in the industry due to their simple structure, high computational efficiency, and the combination of convolution and pooling to make the network have certain shift-invariant and equivariance, so CNN-based detection models are still the preferred choice in the industry. In addition, CNN-based detectors can further enhance their performance by introducing some improved techniques. For example, the convolution proposed in this paper combines the advantages of Ghost Convolution and Depthwise Convolution to increase the expressive power of the model so that it can better capture the relevant features in the input data. Think about the application of Lightweight-DepthGhost Convolution in Neck networks in terms of the generalization ability of the model. Neck networks are usually responsible for performing feature fusion and information transfer to extract richer and more abstract features, effectively enhancing the generalization ability of the model. However, the advantages of Lightweight-DepthGhost Convolution may be less obvious in backbone networks. This is because in backbone networks, the main burden of computation comes from dealing with more details and local information, not just the size of the model. Therefore, when choosing Lightweight-DepthGhost Convolution, its advantages and limitations need to be considered comprehensively according to specific application scenarios and requirements.

CONCLUSIONS

In this paper, a lightweight convolution called Lightweight-DepthGhost Convolution is proposed for optimizing the Neck network part of the YOLOv8 model. This convolution allows the network to significantly reduce the computational effort and the number of parameters while effectively capturing the target features at different scales. In addition, we improve the bottleneck module using the lightweight convolution proposed in this paper and introduce the Coordinate Attention mechanism. This attention mechanism helps the model to better process the important information in the input sequence by focusing on the information and spatial features between the channels. In order to verify the effectiveness and robustness of Lightweight-DepthGhost Convolution, experiments are conducted on different public datasets. The experimental results show that the improved model reduces the amount of computation and the number of parameters by about one-tenth while ensuring better accuracy than the original model. This improved model can be applied not only in the field of public transportation, but also in the detection of defects or anomalies in industrial products. On the NEU-DET dataset, the improved model proposed in this paper also performs well, overall outperforming other detection models, and realizes high-precision and low-computation target detection, thus reducing labor costs and improving productivity.

ACKNOWLEDGEMENTS

This work was partially supported by the Fujian University of Technology under Grant GY-Z20016 and GY-Z19005, and partially supported by the National Science and Technology Council under grant NSTC 113-2221-E-224-056-.

REFERENCES

Alhichri H, Alswayed AS, Bazi Y, Ammour N, Alajlan NA (2021). Classification of remote sensing images using efficientnet-b3 cnn model with attention. IEEE Access 9:14078–94.

Bastidas AA, Tang H (2019). Channel attention networks. In: Proc Eur IEEE Conf Comput Vis Pattern Recog.

Cao M, Fu H, Zhu J, Cai C (2022). Lightweight tea bud recognition network integrating ghostnet and yolov5. Math Biosci Eng 19:12897–914.

Carreira J, Madeira H, Silva JG (1998). Xception: A technique for the experimental evaluation of dependability in modern computers. IEEE Trans Softw Eng 24:125–36.

Fan Q, Brown L, Smith J (2016). A closer look at faster r-cnn for vehicle detection. In: Proc Eur IEEE Intelligent Vehicles Symp. IEEE.

Fu H, Song G, Wang Y (2021). Improved yolov4 marine target detection combined with cbam. Symmetry 13:623.

Gai R, Chen N, Yuan H (2023). A detection algorithm for cherry fruits based on the improved yolo-v4 model. Neural Comput Appl 35:13895–906.

Hou Q, Zhou D, Feng J (2021). Coordinate attention for efficient mobile network design. In: Proc Eur IEEE Conf Comput Vis Pattern Recog.

Huan Z, Zhou J, Xie Y, Xu J, Wang H, Ma W, Li X, Zhou W, Luo T (2024). Automated droplet manipulation enabled by a machine-vision-assisted acoustic tweezer. Sensor Actuat B chem 418:136352.

Humphreys GW, Sui J (2016). Attentional control and the self: The self-attention network (san). Cogn Neurosci 7:5–17.

Jiang S, Xu T, Li J, Huang B, Guo J, Bian Z (2019). Identifynet for non-maximum suppression. IEEE Access 7:148245–53.

Li B, He Y (2018). An improved resnet based on the adjustable shortcut connections. IEEE Access 6:18967–74.

Li F, Bai H, Zhao Y (2020). Learning a deep dual attention network for video super-resolution. IEEE Trans Image Process 29:4474–88.

Li Y, Huang H, Xie Q, Yao L, Chen Q (2018). Research on a surface defect detection algorithm based on mobilenet-ssd. Appl Sci 8:1678.

Liu M, Wu W, Gu Z, Yu Z, Qi F, Li Y (2018). Deep learning based on batch normalization for p300 signal detection. Neurocomputing 275:288–97.

Ma N, Zhang X, Zheng HT, Sun J (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proc Eur Conf Comput Vis.

Meng R, Rice SG, Wang J, Sun X (2018). A fusion steganographic algorithm based on faster r-cnn. CMS Comput Mater Con 55.

Mg WHE, Zin TT, Tin P, Aikawa M, Honkawa K, Horii Y (2025). Automated system for calving time prediction and cattle classification utilizing trajectory data and movement features. Sci Rep 15:2378.

Michele A, Colin V, Santika DD (2019). Mobilenet convolutional neural networks and support vector machines for palmprint recognition. Procedia Comput Sci 157:110–7.

Mnih V, Heess N, Graves A, Kavukcuoglu K (2014). Recurrent models of visual attention. Adv Neural Inf Process Syst 27.

Nazir T, Nawaz M, Rashid J, Mahum R, Masood M, Mehmood A, Ali F, Kim J, Kwon HY, Hussain A (2021). Detection of diabetic eye disease from retinal images using a deep learning based centernet model. Sensors 21:5283.

Rabbi MF, Sultan MN, Hasan M, Islam MZ (2023). Tribal dress identification using convolutional neural network. J Inf Hiding Multim Signal Process 14:72–80.

Redmon J, Divvala S, Girshick R, Farhadi A (2016). You only look once: Unified, real-time object detection. In: Proc Eur IEEE Conf Comput Vis Pattern Recog.

Redmon J, Farhadi A (2017). Yolo9000: better, faster, stronger. In: Proc Eur IEEE Conf Comput Vis Pattern Recog.

Ren M, Zhang X, Chen X, Zhou B, Feng Z (2023). Yolov5s-m: A deep learning network model for road pavement damage detection from urban street-view imagery. Int J Appl Earth Obs Geoinf 120:103335.

Ren S, He K, Girshick R, Sun J (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Adv Neur In 28.

Tammina S (2019). Transfer learning using vgg-16 with deep convolutional neural network for classifying images. Int J Sci Res 9:143–50.

Tan M, Le Q (2021). Efficientnetv2: Smaller models and faster training. In: Proc Eur Int Conf Mach Learn. PMLR.

Tawfeeq LA, Hussein SS, Mohammed MJ, Abood SS (2021). Predication of most significant features in medical image by utilized cnn and heatmap. J Inf Hiding Multim Signal Process 12:217–25.

Tian Y, Yang G, Wang Z, Wang H, Li E, Liang Z (2019). Apple detection during different growth stages in orchards using the improved yolo-v3 model. Comput Electron Agr 157:417–26.

Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh JW, Yeh IH (2020a). Cspnet: A new backbone that can enhance learning capability of cnn. In: Proc Eur IEEE Conf Comput Vis Pattern Recog.

Wang J, Yang L, Huo Z, He W, Luo J (2020b). Multi-label classification of fundus images with efficientnet. IEEE Access 8:212499–508.

Wang Q, Zhang X, Chen G, Dai F, Gong Y, Zhu K (2018). Change detection based on faster r-cnn for high-resolution remote sensing images. Remote Sens Lett 9:923–32.

Wu W, Liu H, Li L, Long Y, Wang X, Wang Z, Li J, Chang Y (2021). Application of local fully convolutional neural network combined with yolo v5 algorithm in small target detection of remote sensing image. Plos One 16:e0259283.

Xiong C, Zayed T, Abdelkader EM (2024). A novel yolov8-gam-wise-iou model for automated detection of bridge surface cracks. Constr Build Mater 414:135025.

Yang A, Jiang T, Han Y, Li J, Li Y, Liu C (2022). Research on application of on-line melting in-situ visual inspection of iron ore powder based on faster r-cnn. Alex Eng J 61:8963–71.

Zhang D, Hao X, Liang L, Liu W, Qin C (2022a). A novel deep convolutional neural network algorithm for surface defect detection. J Comput Des Eng 9:1616–32.

Zhang K, Shen H (2021). Solder joint defect detection in the connectors using improved faster-rcnn algorithm. Appl Sci 11:576.

Zhang X, Zeng H, Guo S, Zhang L (2022b). Efficient long-range attention network for image super-resolution. In: Proc Eur Conf Comput Vis. Springer.

Zhao X, Song Y (2023). Improved ship detection with yolov8 enhanced with mobilevit and gsconv. Electronics 12:4666.

Zhou K, Tong Y, Li X, Wei X, Huang H, Song K, Chen X (2023). Exploring global attention mechanism on fault detection and diagnosis for complex engineering processes. Process Saf Environ 170:660–9.

Zhu L, Lee F, Cai J, Yu H, Chen Q (2022). An improved feature pyramid network for object detection. Neurocomputing 483:127–39.