IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 February 2024, accepted 13 March 2024, date of publication 18 March 2024, date of current version 22 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3378999

RESEARCH ARTICLE

LCG-YOLO: A Real-Time Surface Defect Detection Method for Metal Components

JIANGLI YU, XIANGNAN SHI, WENHAI WANG, AND YUNCHANG ZHENG
Hebei University of Architecture, Zhangjiakou, Hebei 075000, China

Corresponding author: Yunchang Zheng (zyc2023@hebiace.edu.cn)

This work was supported by the Basic Scientific Research Business Fund Project of Universities in Hebei Province under Grant 2022CXTD08.

ABSTRACT Surface defect inspection of metal components plays a critical role in ensuring product quality, enhancing production efficiency, and reducing costs, with particular emphasis on the detection of small-sized surface defects to ensure the safety and reliability of metal components during their usage. Existing detection methods for small size defects on the surfaces of metal components have some shortcomings, such as low precision and poor real-time performance. To solve these two problems, this paper proposes a real-time defect detection method based on improved YOLO. Firstly, the LSandGlass (LSG) module is used to replace the residual module in the backbone network, which reduces information loss, eliminates the low-resolution feature layer, and minimizes semantic loss. The network then uses a lightweight Ghost convolution at the neck to extract the network features. In addition, the convolutional block attention mechanism (CBAM) module is added to improve the detection precision of small-size defects. Finally, soft intersection over union (SIoU) is used to further enhance target detection capability. The experiment was carried out a self-made hexagonal bolt data set of typical commonly used metal components. The experimental results show that compared to the original YOLOv5, the mAP (0.5) is improved by 5.7% to 95.50%, and the reasoning FPS is improved by 21 fps to 95 fps. These results indicate that the proposed LCG-YOLO improves the real-time detection performance of metal component surface defects.

INDEX TERMS Surface defect detection, LSandGlass, CBAM, ghost, SIoU, YOLOv5.

I. INTRODUCTION

The detection of surface defects in metal parts is important for quality control [1]. In industrial production, metal surface defects have a serious impact on the performance of parts; it is necessary to detect the surface in metal parts to ensure quality. Through the detection of defects on the metal surface, defects can be found early, avoiding the loss of subsequent processing, testing and other links, and reducing production costs [2]. The more automated the detection of surface defects of metal parts, the higher the detection efficiency, which helping to improve the production efficiency and productivity. This inspection requires high-precision, high-efficiency and non-contact inspection of metal surfaces to detect defects in a timely manner and requires the ability to distinguish between different types of defects. Therefore, the detection of metal components is essential for quality control in industrial production.

Traditional metal surface defect detection technology relies on manual detection, which has some disadvantages, such as low efficiency, low precision and reliance on expert experience. To overcome these shortcomings, automated defect detection technologies have been rapidly developed, such as optical inspection, magnetic particle inspection, ultrasonic inspection and computer vision [3]. However, optical detection can easily be influenced by the external light intensity and stability of the light source, potentially leading to erroneous judgments. Magnetic testing can only identify ferromagnetic materials that are susceptible to magnetic field interference, and that is incapable of detecting three-dimensional defects. Ultrasonic testing is susceptible to the form and quality of the object being evaluated, ultrasound

The associate editor coordinating the review of this manuscript and approving it for publication was Wen-Sheng Zhao.

interference, and other variables. It may not be able to identify internal imperfections for some diverse materials, such as cast iron. Computer vision detection capabilities may not identify objects with unique shapes and surface colors.

With the development of deep learning technology, deep learning-based computer vision defect detection technology has been widely used. The technology to detect defects using deep learning can be categorized into two parts, such as two-stage and one-stage series. For two-stage detection, researchers have proposed algorithms such as CNN [4], Fast-RCNN [5], Faster RCNN [6] and R-CNN [7]. The detection method based on R-CNN focuses on sensor, deep learning and target-related problems, but there are some problems such as long running time, large environmental impact and target size difference. However, this method is not suitable for real-time detection in different research environments. To efficiently detect targets in images while generating high-quality segmentation masks for each instance, researchers invented Mask R-CNN [8], which extends Faster R-CNN and adds a branch to predict target masks to the existing boundary box recognition branch. In 2016, researchers proposed a series of single-lens multi-box detector (SSD) [9] algorithms for one-stage detection. SSD uses prior frames with different scales and aspect ratios and extracts feature maps with different scales for detection. Large-scale feature maps can be used to detect small objects, whereas small-scale feature maps can be used to detect large objects. In 2019, an improved small target detection algorithm FA-SSD [10] based on SSD was proposed, which is divided into two structures: F-SSD and A-SSD. F-SSD is used to fuse feature layers of different sizes to enhance the feature information in the context. A-SSD is used to establish feature relationships in the feature mapping space. The SSD is similar to the YOLO in run speed and faster RCNN in terms of detection precision. However, the debugging process in the network is highly dependent on experience and there are some problems such as a low degree of feature convolution and insufficient feature extraction. The You Only Look Once (YOLO) [11] algorithm proposed by Redmon et al. has a high detection speed. By dividing the grid, the object and its class boundary box can be returned directly, and object recognition is treated as a regression problem, which increases the detection speed. The YOLOv5 proposed by the Ultralytics team in 2020 has different variants [12] (such as YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x and YOLOv5n) to represent models of different sizes and levels of complexity. These variants offer different trade-offs between speed and precision to accommodate different computing powers and real-time requirements. GSConv is a lightweight convolutional technique proposed in 2022 that can reduce the complexity of models while maintaining accuracy in the implementation object detection tasks [13]. In 2023, YOLOv8's one of the key characteristics is scalability [14]. The new backbone network, anchorless detection head and loss function also provide good algorithm conditions for improving detection precision. The bottleneck design paradigm aims to improve the cost-effectiveness of the detectors. Shift-ConvNets were proposed in 2024 by scientists from Shenzhen University as a convolutional neural network architecture. It replaces traditional convolution operations with shift operations, thus reducing the model parameters and improving computational efficiency. This network structure introduces shift offsets to control the receptive field of the convolutional kernel, enabling better capture of features and exhibiting good spatial and translational invariance [15]. However, one of its disadvantages is that it requires higher computing resources to achieve high precision and high detection speed, and another disadvantage is that the detection effect of small targets is not ideal, and there are some problems in our detection of metal parts.

In recent years, new methods for defects have been proposed in deep learning metal surface defect detection methods. For metal surface defects, Wenfang et al. [16] conducted simulation experiments to verify the ability of the support vector machine to identify the first and second echoes of the defects. By combining ultrasonic signal propagation theory with an RBF neural network, Wenluan et al. [17] proposed a new method to locate and detect near-surface micro hole defects in metal alloy components using four waveforms. Han et al. [18] used the DIoU-NMS method to replace the traditional NMS algorithm and improve the recognition of repeatedly blocked targets.

FIGURE 1. Some surface defects of metal components.

However, as shown in FIGURE 1, for small-size surface defect detection methods for metal parts [19], the following arise as the problems arise:

(1) Low detection precision: for small surface defects, many traditional detection methods have difficulty meeting high-precision detection requirements, and they are easy to miss or misdirect.

(2) Slow detection speed: Traditional metal surface defect detection methods require a certain amount of time for scanning and analysis, and require manual intervention, resulting in a slow detection speed.

(3) High detection cost: Some traditional metal surface defect detection methods require professional operation and high-end detection equipment, which is expensive and unsuitable for large-scale production lines.

(4) Struggles in adapting to complex working environments: When detecting defects in metal surfaces; it is often necessary to meet specific working conditions, such as temperature and humidity, so it cannot adapt to complex working conditions.

(5) It is difficult to achieve automated detection: Because the traditional method requires human intervention and

operation is more complex, it is difficult to achieve automated detection and cannot meet the needs of large-scale production lines [20].

To solve these problems, it is necessary to develop more efficient, convenient and automated small-size surface defect detection methods for metal parts to achieve more accurate, high-speed and low-cost detection. In this study, the original YOLOv5 algorithm is improved to enhance the detection precision and the recall rate of metal targets while supporting the detection speed, which can achieve real-time detection in different research environments.

The main contributions of our work can be summarized as follows.

(1) The LSandGlass (LSG) [21] module was utilized to replace the residual module in the backbone, which can extract features from different scales and enhance the feature table through a cross-layer connection. It can reduce semantic loss and has low computational complexity, which is helpful in improving detection precision.

(2) Our method utilizes the lightweight Ghost model [22], and the adjustments we made include modifying the number of network channels, changing the size and number of convolutional cores, and optimizing the structure. The grouping convolution method eliminates the correlation between channels so that the current channel features are only relevant to themselves. On the one hand, the mode of redundant feature generation is simulated, and on the other hand, the number of parameters and calculations are significantly reduced. These advances have improved the precision and speed of object detection.

(3) The Convolution block attention module (CBAM) [23] improves the ability of the YOLO network to accurately extract and utilize the information feature. This enhances target perception, which in turn increases the accuracy and robustness of target detection.

(4) Replacement of the generalized intersection over union (GIoU) [24] with a soft intersection over union (SIoU) [25] improves the speed of training and the precision of reasoning.

The remainder of this study is structured as follows. Section II describes the improvements to the original YOLO algorithm. In Section III, we describe the data set and present the experimental results. Finally, we summarize the literature and identify the scope and limitations of future research in Section IV.

II. PRINCIPLE AND METHOD IMPROVEMENT

A. PRINCIPLE OF THE ORIGINAL YOLOv5

The YOLO algorithm is a deep learning-based object detection algorithm that has the characteristics of high speed and high real-time performance and is suitable for applications with high real-time requirements. The defect detection technology based on the YOLO algorithm is also the most widely used [26]; however, it also has some shortcomings, such as poor detection effect on small targets and easy to ignore small targets. Compared to traditional object detection algorithms, YOLO transforms the object detection problem into a regression problem, dividing the image into multiple grid cells, each of which predicts the category and location information of the object [27]. First, the input image is divided into fixed-size grid cells, each of which is responsible for predicting one or more targets. As each cell of is predicted, the output contains information on the target category, location, and confidence level. Filter the final target boxes using non-maximum suppression (NMS) [28], remove overlapping boxes, and select the box with the highest confidence. Finally, the targets are filtered according to the confidence threshold, and only the targets whose confidence is higher than the threshold are retained.

The structure of YOLOv5 follows the classic object detection framework, which is divided into Backbone, Neck and Head. The Backbone of YOLOv5 is responsible for the feature extraction of the input images. The YOLOv5 neck processing module uses a path aggregation network (PANet) [29] to fuse features at various levels through a pyramid structure. The YOLOv5 header module is responsible for predicting the location and category of the target. After the feature passes through multiple convolution layers and fully connected layers, the prediction results are outputted, and each prediction result contains a bounding box and the corresponding class probability. YOLOv5 uses the GIoU loss function.

In general, the YOLOv5 framework has a simple and efficient structure, promotes target detection as a regression problem, realizes end-to-end training and detection, and has a good speed-precision balance. Features are extracted through the backbone network, fused through the neck processing module, and finally, detection results are generated through the head module. This design enables YOLOv5 to achieve a fast-reasoning speed while maintaining precision. Therefore, we chose YOLOv5 as the baseline and made some improvements to meet our requirements for real-time detection of metal surface defects.

B. IMPROVED NETWORK STRUCTURE

1) LSANDGLASS MODEL

Obtain richer information on the combination of gradients and reduce information loss and gradient confusion [30]. This study uses the LSG hourglass module to replace the Resunit [31] residual module of the C3 module in YOLOv5 to further enhance the representation of features.

The LSG module is an improvement over YOLOv5, designed to improve object detection performance and precision. Four LSG modules are used to replace the residual modules in the original YOLOv5 backbone network, which can extract features from different scales and enhance the ability to express features through cross-layer connections. The smaller feature layer has a larger acceptance field and can capture more semantic information but will lead to the loss of local and detailed features. By contrast, shallower convolutional neural networks have smaller acceptance fields and are more concerned with local and detailed information. To reduce semantic loss, we chose to remove

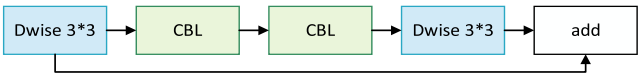

FIGURE 2. The schematic diagram of the LSG module.

the 19 × 19 feature layer from the backbone feature extraction network, while retaining the other two feature layers, which reduces both semantic loss and the number of parameters. FIGURE 2 shows the structure of the LSG model.

The LSG module can capture target information at different scales and fuse features on various levels to improve the precision and robustness of target detection. Experiments show that the LSG module performs better on various target detection data sets and can significantly improve the average precision (mAP) and precision of target detection compared to the traditional YOLOv5 model.

The LSG module has the following advantages over other models in object detection tasks:

(1) Relatively low computational complexity: Compared to some complex target detection models, such as Faster R-CNN and Mask R-CNN, the LSG module has lower computational complexity and can achieve efficient target detection in the case of limited resources.

(2) High precision and robustness: the LSG module can better capture multiscale information, enhance feature expression capabilities, and achieve high precision and robustness in target detection tasks.

(3) Scalability and flexibility: the LSG module can be combined and extended with other object detection models, providing more design space and flexibility to adapt to different scenarios and requirements for object detection tasks.

FIGURE 3 shows a comparison of before and after using the LSG module. The two pictures on the left side of FIGURE 3 show the results of 6 successive convolutions without the LSG module. The edge features of the metal parts are not well obtained, and the loss of information is significant. In the two pictures to the right of FIGURE 3, the edge feature information of the metal parts can be better extracted using the improved LSG feature extraction results, and the difference between the background information and feature information is more obvious.

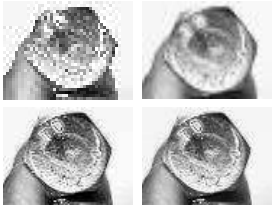

FIGURE 3. Improvement of the image features based on LSG.

2) GHOST CONVOLUTION

To optimize the parameter scale and computing resource consumption of the network, and improve the detection speed, this paper replaces the C3 module in the neck with the C3-Ghost module. In the Ghost module, the input channels are split into two parts, with the backbone network managing one input channel and using a 3 × 3 convolutional kernel for processing, while the Ghost network handles the other input channel. For example, if the input channel number is C, the number of channels processed by the backbone network is $C/2$, and the number of channels processed by the virtual network is $C/2$. The backbone is responsible for performing convolution operations and generating the backbone feature maps. The backbone is typically a standard convolutional layer that receives half of the input channels and outputs the corresponding feature map. A ghost network also performs convolution operations; however, its output is called a ghost feature graph. The Ghost network receives the other half of the input channels and outputs the corresponding feature map. The Ghost module obtains the final output feature map by fusing the backbone feature map with the Ghost feature map.

By setting the stride to 2, we can ensure that the convolution operation is performed with a 2-pixel interval both vertically and horizontally, thus effectively reducing the dimensions of the image. The 640 × 640 input image is processed through the first convolutional layer, resulting in an output size of 319 × 319. Subsequently, the output size of the second convolutional layer was adjusted to 159 × 159. The third convolutional layer further reduces the size to 79 × 79. Finally, after passing through the fourth convolutional layer, the output size is refined to 39×39 pixels. This gradual reduction in size aids in extracting important features from the image, reducing computational complexity and thereby accelerating the training speed of the model. Depending on the network structure and application requirements, fusion can take the form of simple element-level additions or connections. Because the computation is reduced and a feature representation capability comparable to standard convolution is provided, the computation cost is kept low, and the real-time detection effect of the metal-parts network is improved. The Swish function is applied to the feature maps after convolution to introduce nonlinear factors and enhance the feature representation, making the model better simulate and process complex input data, and improve the performance and accuracy of the model. Therefore, it is applied to deep learning object detection and image classification where efficiency needs to be improved, and the Ghost module is shown in FIGURE 4.

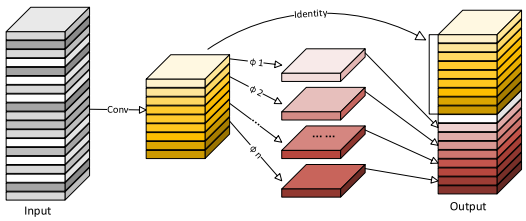

FIGURE 4. The ghost module.

3) CBAM

To improve the detection precision of metal part surface defects, the CBAM module is added in this study, which can enhance the model's attention to specific areas without increasing the amount of calculation, to improve the model's performance in target detection, image segmentation and other tasks. The CBAM attention mechanism is embedded into the network for feature extraction and enhancement. First, the image is input, and then the feature mapping is calculated through two submodules, the channel attention module (CAM) and the spatial attention module (SAM), to obtain the focus weight. The CAM adaptively adjusts the importance of different channels by calculating the attention weights of channel dimensions in the feature graph, so that the network can better focus on useful feature channels and suppress irrelevant channel information. SAM adaptively adjusts the importance of various positions by calculating the attention weight of the spatial dimension of the feature graph, so that the network can better focus on the target region and suppress background noise interference. The weight coefficient of the output is then multiplied by the feature mapping of the input one by one to obtain the new enhanced feature. Finally, experiments are conducted on the data set to verify the detection performance. FIGURE 5 shows the CBAM structure diagram. On this basis, the LCG-YOLO network can more accurately extract and use the feature information, enhance the perception of the target to improve the precision and robustness of target detection, and provide a better representation of the entire network.

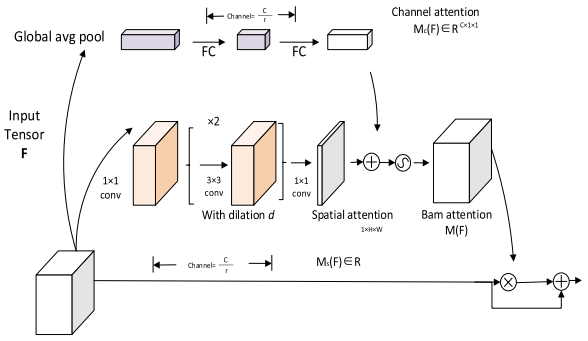

FIGURE 5. The CBAM structure diagram.

4) SIOU

In the original YOLOv5 network, the predicted bounding box is processed using GIoU as a loss function, which successfully resolves the disjoint issue between the predicted and actual bounding boxes. However, GIoU is unable to accurately ascertain the relationship between two boxes and to indicate the intersection of two boxes when the two boxes are contained within each other or have different aspect ratios. Therefore, to overcome the GIoU deficiency, this study replaced GIOU in the original network with SIOU.

The loss function of the IoU, for example, GIoU, distant Intersection over Union Loss (DIoU) [32] and Completed Intersection over Union (CIoU), does not consider the direction between the real and predicted boxes, resulting in a slow convergence rate. In this regard, SIoU introduces a vector Angle between the real box and the predicted box and redefines the relevant loss function, which includes four parts: angle, distance, shape cost and IoU costs.

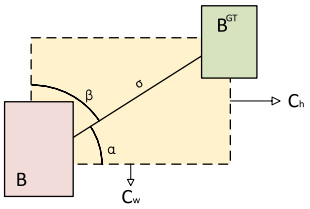

FIGURE 6. The scheme for calculation the contribution of the angle cost to the loss function.

As shown in FIGURE 6, the model first attempts to make predictions on either the X-axis or the Y-axis (the closest) and then continues along the relevant axis.

The angle cost can be defined as

$$\begin{aligned} \wedge &= 1 - 2^* \sin^2\left(\arcsin\left(\frac{c_h}{\sigma}\right) - \frac{\pi}{4}\right) \\ &= \cos\left(2^*\left(\arcsin\left(\frac{c_h}{\sigma}\right) - \frac{\pi}{4}\right)\right) \end{aligned} \tag{1}$$

where c_h is the height difference between the center point of the real box and the prediction box, σ is the distance between the center point of the real box and the prediction box, and $\arcsin(c_h/\sigma)$ is equal to Angle α.

$$x = \frac{c_h}{\sigma} = \sin(\alpha) \tag{2}$$

$$\sigma = \sqrt{\left(b_{c_x}^{gt} - b_{c_x}\right)^2 + \left(b_{c_y}^{gt} - b_{c_y}\right)^2} \tag{3}$$

$$c_h = \max\left(b_{c_y}^{gt}, b_{c_y}\right) - \min\left(b_{c_y}^{gt}, b_{c_y}\right) \tag{4}$$

$\left(b_{cx}^{gt}, b_{c_x}\right)$ is the center coordinate of the real frame, and $\left(b_{cx}^{gt}, b_{c_x}\right)$ is the center coordinate of the prediction frame, and it can be noted that when α is $\pi/2$ or 0, the angle loss is 0, and during training if $\alpha < \pi/4$, α is minimized, β is minimized.

The distance cost can be defined as

$$\Delta = \sum_{t=x,y}\left(1 - e^{-\gamma\rho t}\right) = 2 - e^{-\gamma\rho x} - e^{-\gamma\rho y}$$

$$\rho_x = \left(\frac{b_{c_x}^{gt} - b_{c_x}}{c_w}\right)^2, \quad \rho_y = \left(\frac{b_{c_y}^{gt} - b_{c_y}}{c_h}\right)^2, \quad \gamma = 2 - \wedge \tag{5}$$

Thereinto, (c_w, c_h) is the width and height of the smallest external rectangle of the rea and prediction boxes.

During $\alpha \to 0$, the contribution of the distance cost is reduced. Conversely, the larger Δ is as α approaches $\pi/4$. Thus, as the Angle increased, the problem became increasingly difficult. As the Angle increases, γ is given time priority

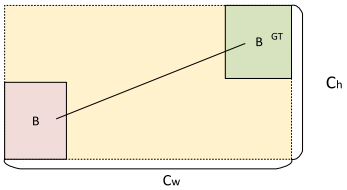

FIGURE 7. The scheme for calculation of the distance between the ground truth bounding box and the prediction of it.

based on the distance value. It is important to note that as $\alpha \rightarrow 0$, distance costs become routine. The distance cost calculation is shown in FIGURE 7.

The shape cost can be defined as

$$\Omega = \sum_{t=w,h} \left(1 - e^{-\omega t}\right)^{\theta} = \left(1 - e^{-\omega_w}\right)^{\theta} + \left(1 - e^{-\omega_h}\right)^{\theta}$$
$$\omega_w = \frac{\left|w - w^{gt}\right|}{\max\left(w, w^{gt}\right)}, \quad \omega_h = \frac{\left|h - h^{gt}\right|}{\max\left(h, h^{gt}\right)} \tag{6}$$

Thereinto, θ defines the shape cost, and its value is unique for each data set. We define it on a scale of 2 to 6. θ is an essential term in this equation, that the amount of attention required for the cost of the shape. If the θ value is set to 1, the shape is immediately optimized, affecting the free movement of the shape. The common algorithm on each data set computes the θ, whose experimental value is close to 4, while the numerical value is within the specified range, indicating that it has a good effect.

The final SIoU loss function is defined as follows.

$$Loss_{SIOU} = 1 - IOU + \frac{\Delta + \Omega}{2}$$
$$IOU = \frac{\left|B \cap B^{GT}\right|}{\left|B \cup B^{GT}\right|} \tag{7}$$

C. THE IMPROVED YOLO MODEL (LCG-YOLO)

The overall algorithm proposed in this paper is shown in FIGURE 8. First, the LSG sandglass module is used to replace the Resunit residual module and remove the 19×19 feature layer from the feature extraction network of the backbone [33]. The C3 structure in the neck is replaced by C3Ghost, reducing the amount of convolution computation. At the same time, when the neck is used for feature fusion, this paper adds the CBAM attention mechanism module to the neck to enhance the characterization ability of deep neural networks. Finally, in the output phase, GIoU is replaced by SIoU.

III. EXPERIMENTS AND DISCUSSIONS

A. EXPERIMENTAL SETTING

Prior to network training, to maximize model performance and avoid overfitting, the GPU is RTX 3090, the video memory is 24GB, and the CPU is a 15vCPU AMD EPYC 7642 48-Core Processor. The CUDA version is 10.1, and the compiled language is Python3.8. The deep learning framework used in the experiment is PyTorch and the optimizer is Adam. The batch size is 32 and the learning rate is set to 0.001. A total of 300 epochs are trained.

B. DATASET AND PREPROCESSING

Based on our research cooperation with local part-processing factories, a reasonable and comprehensive dataset is necessary. The images in the experimental data set are sourced from two main sources. One part consists of 1557 internal data pictures taken from a local factory according to specific needs. These images capture the real-world conditions and variations encountered in manufacturing environments. The other part consisted of 592 images related to metal components obtained from the Internet, providing additional diversity to the data set.

To improve the balance and diversity of the self-made dataset [34], various preprocessing techniques, such as slice-up, grayscale, saturation, filtering and mirroring operations, were applied to the original images. As illustrated in FIGURE 9, these operations aimed to increase the data set and increase its robustness by simulating different environmental conditions and potential variations. Following image preprocessing, a ratio of 7: 2: 1 was assigned for the training, validation, and testing sets, respectively.

The data set contains 2149 bolt surface patterns, which were further expanded to 8569 instances through the application of a series of operations, ensuring a rich and diverse representation of real-world defects and variations in the data. The data set was annotated and defects were classified into five descriptive aspects: circle, crack, damage, scratch, and void, as shown in Table 1, the annotations were created using the LabelImg annotation software and stored in XML files following the PASCAL VOC standard, ensuring compatibility and ease of use with a wide range of deep learning frameworks and tools. To perform defect detection tasks under different environmental conditions and operational scenarios, while working with limited computational resources, we chose a specific input image size of 640×640. This choice allows us to capture finer details of smaller defects efficiently.

C. EVALUATION INDICATORS

Five key indicators were used to measure the effectiveness of the YOLOv5 algorithm Precision, Recall, F1 score, mAP and FPS. Precision is the ratio of the number of objects correctly detected by the model to the total number of objects detected by the model. High precision indicates that the model has a low error rate for the detected target. Precision can be denoted as

$$Precision = \frac{TP}{(TP + FP)} \tag{8}$$

Recall refers to the ability of a category to recall positive classes that should have been correct. The Recall can be denoted as:

$$Recall = \frac{TP}{(TP + FN)} \tag{9}$$

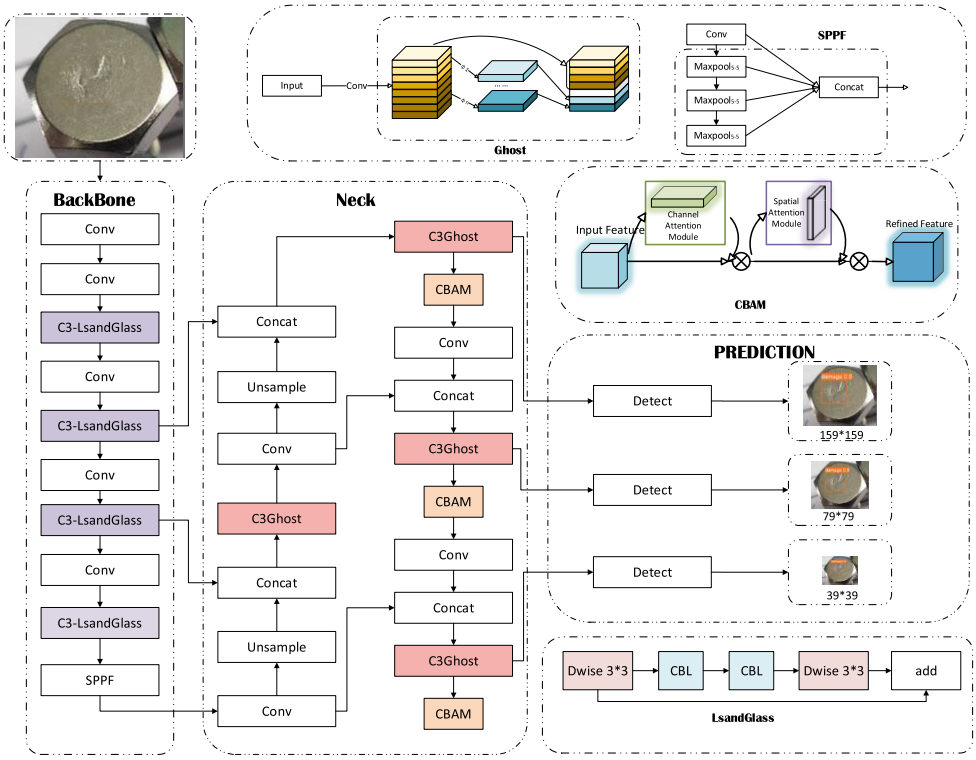

FIGURE 8. The improved YOLO Model (LCG-YOLO).

TABLE 1. Defect sample diagram.

Name of Defect	Description	Image Style
Circle	Hex bolt with a raised circle in the center.	
Crack	Hex bolt with a single crack on the surface.	
Damage	Hex bolt with a deep damage on the surface.	
Scratch	Hex bolt with scratched areas on the surface.	
Void	Hex bolt with one small hole right in the center.	

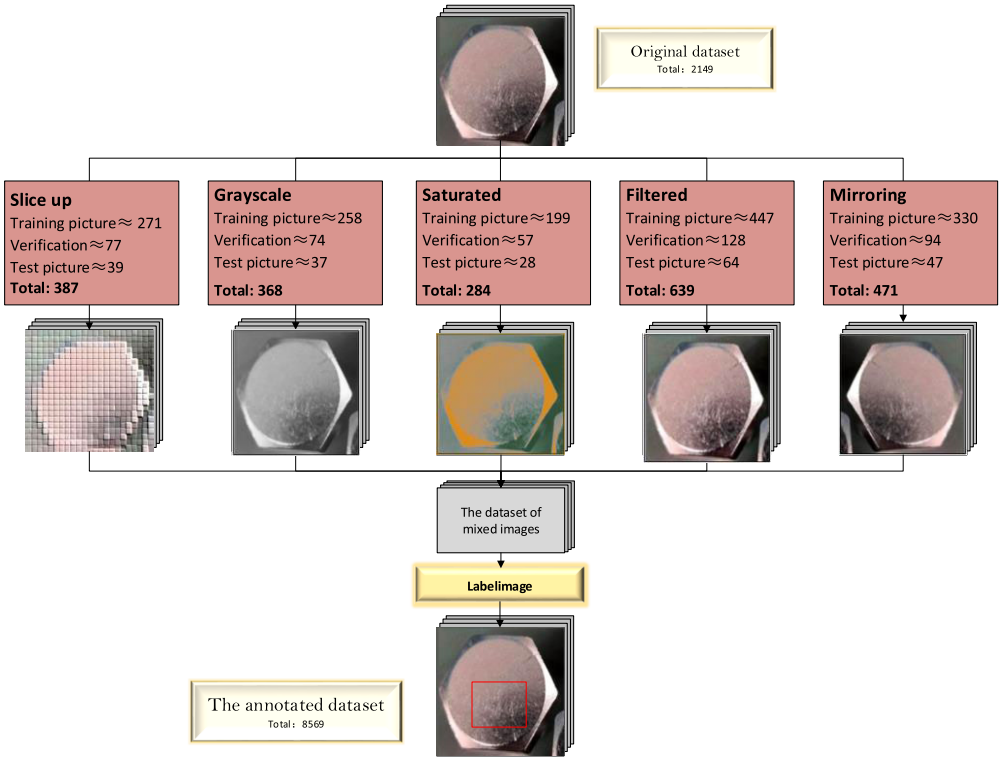

FIGURE 9. Data set and preprocessing.

TP refers to the classification that the classifier correctly predicts as a positive example; FN refers to the classification that the classifier incorrectly predicts as a negative example; and FP refers to the classification that the classifier incorrectly predicts as a positive example.

The F1 score is the harmonic average of precision and recall. Results related to specificity and impact can be obtained by calculating the following formulas. The F1 score can be denoted as

$$F1\text{score}=2 \times \frac{Precision \times Recall}{Prcision + Recall} \tag{10}$$

mAP is the average value of the area under the PR curve for each category. The mAP can be denoted as:

$$mAP = \sum_{C=1}^{C} \frac{Average\ Precision\ (C)}{C} \tag{11}$$

where C represents the number of categories.

To better evaluate the performance of the algorithm in terms of processing speed, we used the FPS as the evaluation index. The FPS can be obtained by calculating the total time required to detect the target and the number of frames processed. The FPS can be denoted as:

$$FPS = \frac{1}{Total\ time/frame\ count} \tag{12}$$

In addition, mAP(0.5) and mAP(0.5:0.95) referring to the area enclosed by mAP after Precision and Recall are used as two axes; represents the average, the number in parentheses represents the threshold for determining IoU as positive and negative samples, and (0.5:0.95) represents the mean value after the threshold is set at 0.5: 0.95.

D. COMPARATIVE EXPERIMENTS

To evaluate our proposed training model, we compared several popular one-stage methods, such as FA-SSD, YOLOv5, YOLOv5-Shift-ConvNets, YOLOv8 and YOLOv8-GSConv. FA-SSD is more prominent in real-time target detection, whereas YOLOv5, YOLOv5-Shift-ConvNets YOLOv8 and YOLOv8-GSConv are optimized and improved on the basis of YOLO series algorithms, providing higher detection precision and faster speed. Our design followed these detection aspects.

The network training process is illustrated in FIGURE 10. After conducting a thorough analysis of our model's training process over 300 epochs, we observed a distinct pattern in the behavior of the loss function value, which served as a critical indicator of the model's learning progress and optimization efficiency. This analysis is structured into three stages based on the observed trends in the loss function values throughout the epochs.

1) THE INITIAL RAPID DECLINE STAGE (0–30 EPOCHS)

During the initial 0-30 epochs, the loss function value of the model showed a sharp decline. This stage is characterized by rapid learning in which the model efficiently captures the underlying patterns in the data. This steep reduction indicates

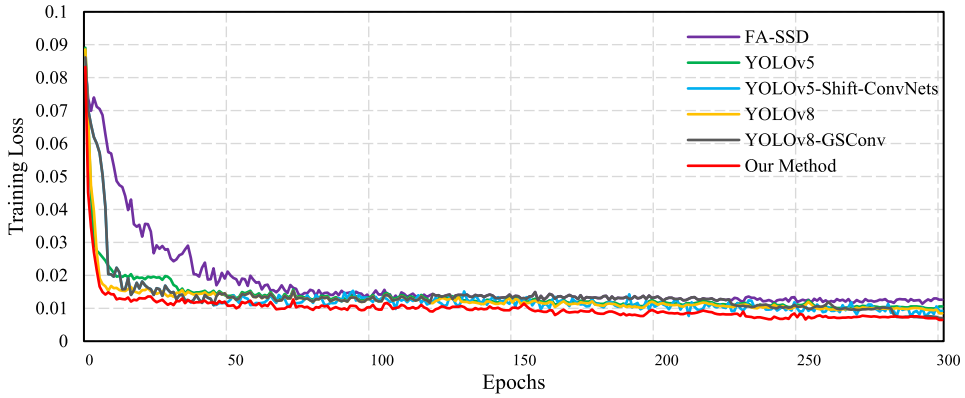

FIGURE 10. The training loss of different models.

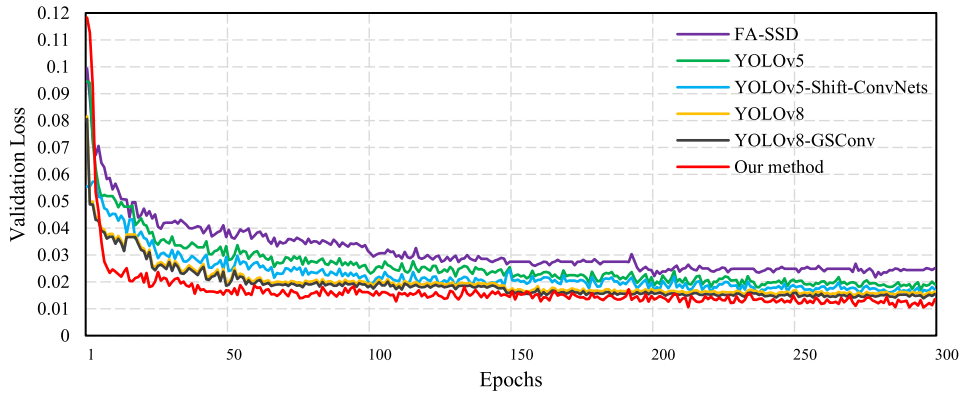

FIGURE 11. The validation loss of different models.

that the model's initial weights are significantly updated towards optimal values, leading to a quick improvement in model performance.

2) THE GRADUAL DECLINE STAGE (30–200 EPOCHS)

Following the initial stage, between 30 and 200 epochs, the rate of decline in the loss function value gradually decreased. This slowdown is indicative of the model entering a more refined tuning stage, in which the adjustments to the weights are smaller and more precise. During this stage, the model continued to learn, but at a slower pace, optimizing its parameters to better fit the training data without overfitting.

3) THE STABILIZATION STAGE (AFTER 200 EPOCHS)

After reaching the 200th epoch, the loss function values tended to stabilize, showing minimal fluctuations. This stabilization suggests that the model has reached its best state, and that further training does not significantly improve the model's performance. At this point, the model probably converges to an optimal or near-optimal solution, and the benefits of additional training epochs are marginal.

In comparison, when analyzing the training process of the other models, we observed that within the first 85 epochs, the loss function decreased to a smaller value. This indicates that the initial learning rate of the other models is slower than that of our model. However, from epochs 85 to 250, the rate of decrease significantly slows down and eventually levels off after 250 epochs. This pattern suggests that our model achieves rapid initial improvements, and that the optimization process tends to stabilize earlier than that of other models, potentially indicating an efficient fine-tuning stage.

In the validation loss chart shown in FIGURE 11, in the first 150 rounds, the losses of our model decreased rapidly and stabilized. It levelled off in 150-250 rounds and eventually stabilized below 0.02.

This comparison highlights the importance of understanding the dynamics of the loss function between the different models and training stages. This demonstrates a balanced learning approach, with significant initial improvements followed by a stable fine-tuning process, eventually reaching a stable optimal state.

Furthermore, to better verify the precision of our model, we add mAP (0.5) as a comparison indicator, which is mAP (0.5). For 300 epochs, our strategy is compared with alternative methods during the training stage. In FIGURE 12, the mAP (0.5) of the individual models grew rapidly in the first 150 rounds. Slow growth in 150-250 epochs with less volatility. Finally, the FA-SSD model performed the worst

TABLE 2. The mAP (0.5) of different methods through 0 to 300 epochs.

Epochs	100	150	200	250	300
FA-SSD	0.795421	0.833019	0.850178	0.863833	0.876125
YOLOv5	0.814912	0.867659	0.868178	0.888347	0.902128
YOLOv5-Shift-ConvNets	0.877765	0.896465	0.913355	0.926489	0.936338
YOLOv8	0.861356	0.888660	0.912051	0.934637	0.945900
YOLOv8-GSConv	0.882974	0.897543	0.912657	0.932635	0.950632
Our method	0.882948	0.910843	0.930248	0.946639	0.955017

TABLE 3. Comparison experiments of different models.

Model	Params (M)	FPS (fps)	Precision (%)	Recall (%)	mAP (0.5) (%)	mAP (0.5:0.95) (%)
FA-SSD	9.3	54	92.5	82.6	87.6	72.3
YOLOv5	**7.2**	74	91.2	82.7	90.0	75.3
YOLOv5-Shift-ConvNets	8.1	68	95.9	89.4	92.5	76.4
YOLOv8	11.2	87	95.4	90.6	94.6	**76.9**
YOLOv8-GSConv	12.1	90	96.9	92.4	95.3	75.2
Our method	10.9	**95**	**98.3**	**93.8**	**95.7**	74.7

TABLE 4. The F1 score for five types of defects in the detection process.

Model	Circle (%)	Crack (%)	Damage (%)	Scratch (%)	Void (%)
FA-SSD	87.451	85.452	89.541	87.451	86.954
YOLOv5	86.364	87.425	86.645	87.645	84.615
YOLOv5-Shift-ConvNets	90.326	94.621	91.976	93.742	94.743
YOLOv8	94.652	89.451	93.451	**96.322**	94.954
YOLOv8-GSConv	95.082	96.034	93.459	95.143	**96.963**
Our method	**96.465**	**98.451**	**94.525**	95.155	96.014

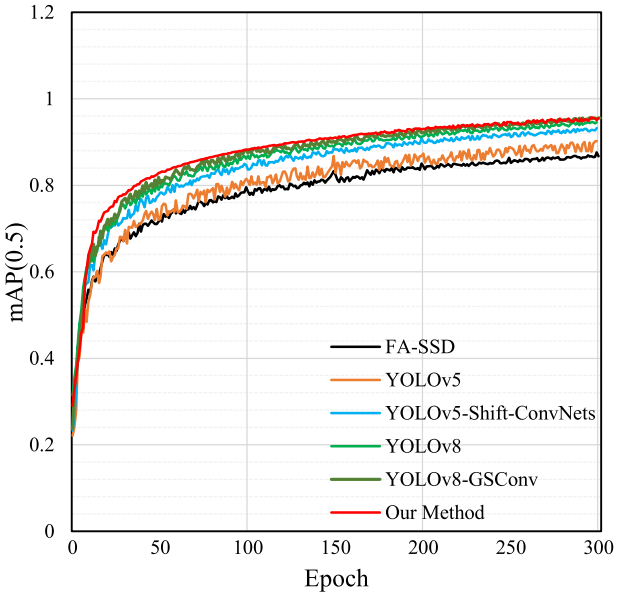

FIGURE 12. The mAP (0.5) iteration diagram for different models.

and the final value stabilized at about 0.88, while the value of YOLOv5-Shift-ConvNets and YOLOv8 grew faster, and the final value stabilized at approximately 0.94. Comparison with various models, our model achieves stability faster at mAP (0.5), and the stable value is better, thus achieving better results. Table 2 shows the mAP (0.5) values for different models at 100, 150, 200, 250, and 300 epochs. Shows that mAP (0.5) increases rapidly during the first 50 epochs and stabilizes after the number of epochs reaches 200 and reaches a value of 0.957 in the 300th epoch. During the training process, the curve of our method performs better than that of other methods, indicating that the network has achieved higher detection precision. In addition, the improved model curves show smoother progress, indicating improved stability. Lastly, our method's optimal mAP (0.5) is 0.95, which is almost 5.7% higher than the original YOLOv5 model's 0.90, indicating the efficacy of the improvement.

As shown in Table3, Table 4, FIGURE 13 and FIGURE 14 show that our method improves detection precision and recall rate.

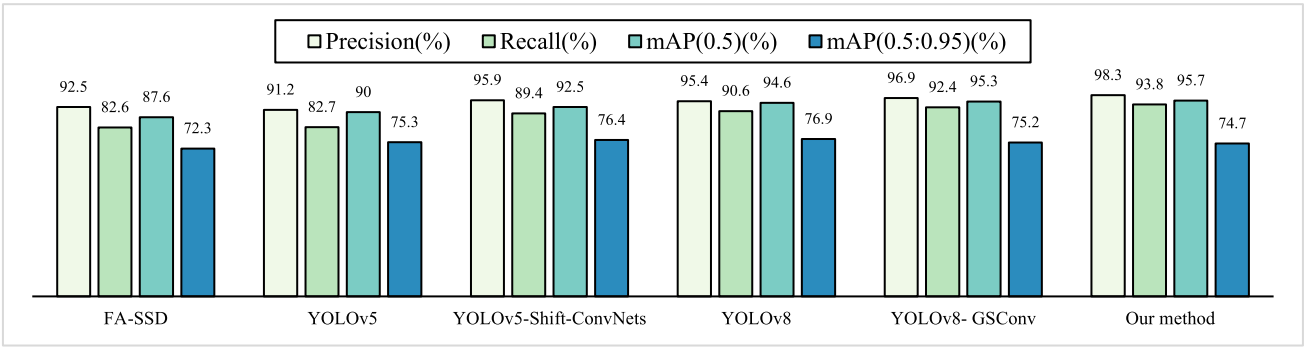

FIGURE 13. The Precision, Recall and mAP of different models.

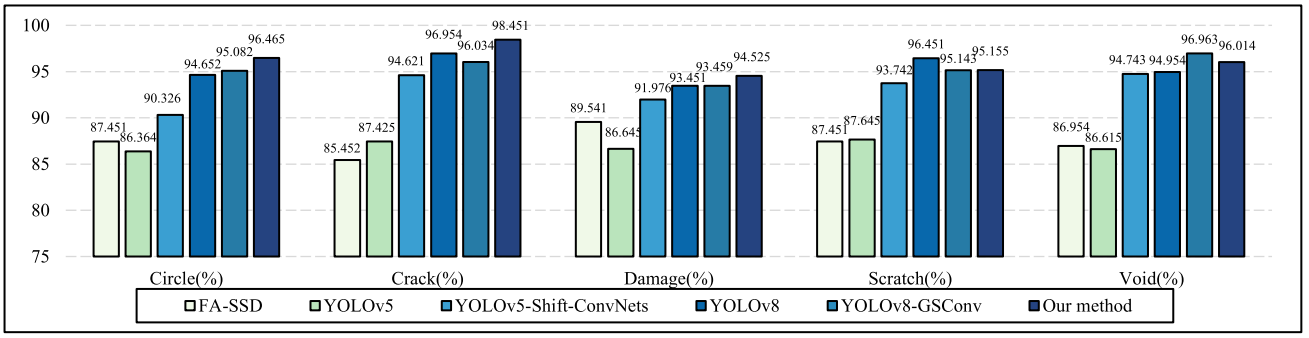

FIGURE 14. The F1 score for five types of defects in the detection process.

In the rapidly developing field of target detection, a model that strikes the best balance among accuracy, speed, and computational efficiency is required. This article presents a comprehensive comparative analysis of a novel object detection model against established benchmarks such as FA-SSD, YOLOv5, YOLOv5-Shift-ConvNets, YOLOv8 and YOLOv8-GSConv. The evaluation focused on key performance metrics, including parameter size, frames per second (FPS), precision, recall, mean Average Precision (mAP), and F1 scores across specific object classes.

Enhanced performance and Efficiency: Our model shows a remarkable improvement in both performance and efficiency, as evidenced by its superior FPS rates and precision-recall metrics when compared to its counterparts. In particular, against FA-SSD, the model not only increased the FPS by 41 but also enhanced the precision and recall by 5.8% and 11.2%, respectively. Such improvements are indicative of the model's ability to process video feeds in real-time while maintaining a high accuracy in object detection.

Parameter optimization: Although our model exhibits an increase in parameter size compared to the FA-SSD and YOLO variants, this increase is justified by significant gains in detection metrics. For example, compared to YOLOv5, the model sees a parameter increase of 3.7%, but delivers a substantial 5.7% increase in mAP (0.5). This trade-off between model complexity and performance enhancement suggests thoughtful optimization of the network architecture to achieve better detection outcomes without excessively burdening computational resources.

F1 scores improvements: The analysis also revealed notable improvements in F1 scores to detect specific object classes such as circle, crack, damage, scratch, and void. These enhancements are particularly prominent when comparing the model with YOLOv5, where F1 scores for circles and cracks show double-digit percentage increases. Such class-specific advancements underscore the model's refined capability to detect and classify various objects with higher precision, making it highly suitable for specialized tasks in industrial inspection or quality control.

Comparative challenges: Despite its strengths, the model faces certain challenges, particularly compared to the latest YOLOv8. A slight decrease in parameter size and a minor reduction in certain F1 scores suggest areas where the model could be further optimized. The decrease in mAP (0.5: 0.95) compared to the benchmarks indicates that there is room for improvement to achieve consistent accuracy across different IoU thresholds.

Taking into account the intricacy of every model and the real detection results, although our method increases the parameters of the model, it also promotes the number of detection frames and map (0.5), which proves that the model has a good detection effect and is superior to other methods in metal part defect detection.

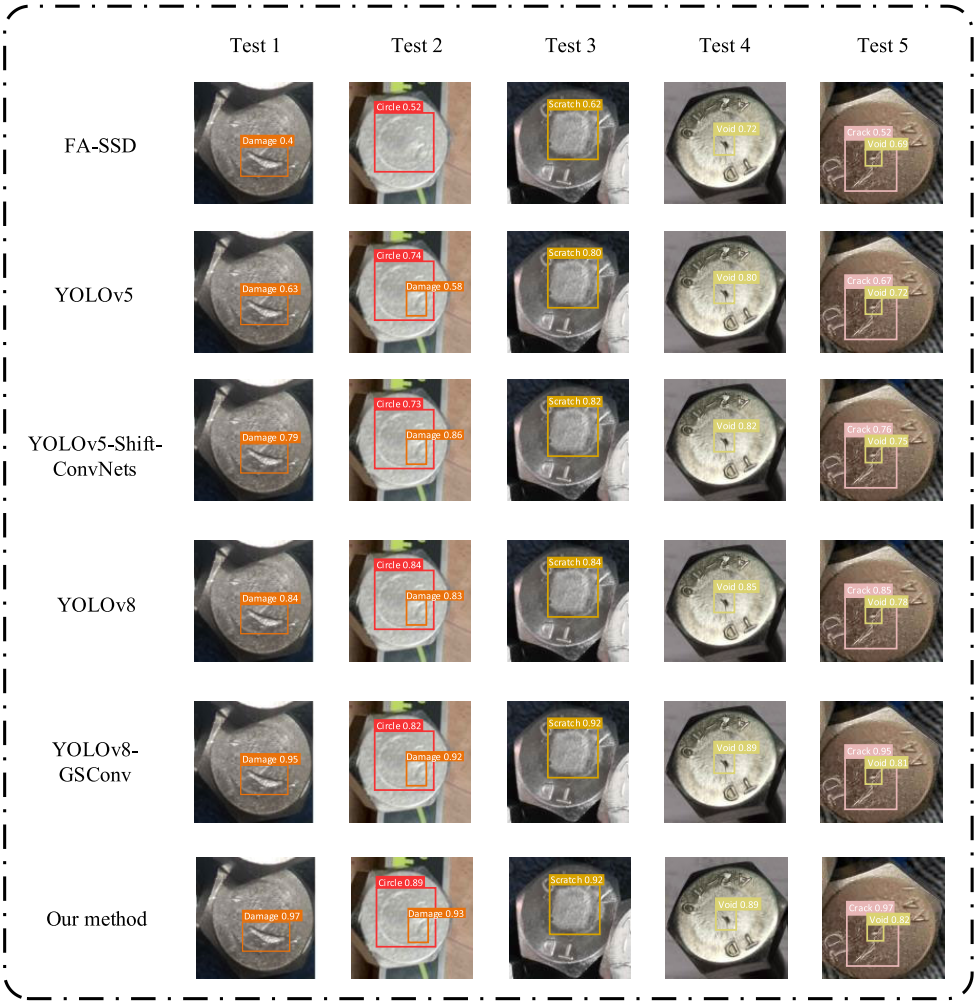

FIGURE 15. Comparison between other algorithms and our method detection.

To evaluate the detection performance of different methods more intuitively from a visual point of view, our method randomly selects a subset of photos from the data set for detection and compares five other detection methods with the improved method.

As shown in FIGURE 15, to compare the detection effects of various methods more intuitively, we conducted multiple groups of comparative experiments for different scenes.

In a series of comprehensive tests designed to benchmark the performance of various object detection algorithms under different environmental conditions, our algorithm consistently outperformed established methods such as FA-SSD, YOLOv5, and their variants. These tests were meticulously designed to simulate real-world scenarios that pose significant challenges to object detection systems, including variations in lighting, occlusions, and surface conditions. The results of these tests provide valuable information on the robustness and accuracy of our method compared to its counterparts.

(1) Enhanced performance under adverse lighting conditions (in Test 1): The first test, set in a low-light environment, revealed a notable disparity in confidence levels between our algorithm and others such as FA-SSD and YOLOv5. Our algorithm demonstrated superior confidence and precision in detecting defects, underscoring its effectiveness in challenging lighting conditions. This suggests that our model has advanced features or mechanisms that enable it to maintain high detection accuracy even when the illumination is suboptimal.

(2) Superior detection accuracy in different scenarios (in Test 2): In high-light scenes, as observed in Test 2, FA-SSD exhibited instances of missed detection, which is a critical flaw that was not present in our algorithm. Although YOLOv5, YOLOv5-Shift-ConvNets, and YOLOv8 were able to identify all defects on the metal surfaces, they were weak in terms of precision compared to our method. This indicates that our algorithm not only ensures comprehensive detection, but also maintains a higher standard of precision, making

TABLE 5. Ablation experiments.

Model	+LSandGlass	+Ghost	+CBAM	+ SIoU	Precision	Recall	mAP (0.5)	mAP (0.5:0.9)
Original					91.2	82.7	90.0	**75.3**
-	√				96.5	83.6	90.2	74.3
-		√			95.8	90.8	91.3	74.1
-			√		96.2	89.9	94.9	74.2
-				√	96.6	90.7	95.0	73.8
-	√	√			96.1	92.6	95.3	74.5
-	√	√	√		97.5	93.4	95.5	74.6
Our Method	√	√	√	√	**98.3**	**93.8**	**95.6**	74.7

it particularly reliable for applications where accuracy is paramount.

(3) Robustness against partial occlusion and surface dirt (in Test 3 and Test 4): The third test scenario involved images with partial occlusion, which is a common challenge in real-world applications. Unlike several other algorithms, our method did not show missed detections and maintained a higher detection confidence. Similarly, in scenarios where the surface of the object was dirty (Test 4), our algorithm again proved its mettle by accurately detecting all surface defects without omissions and showing higher confidence levels than its competitors. These results highlight the robustness of our algorithm against common obstacles, such as occlusion and surface impurities.

(4) Consistent performance in low-complexity scenes (Test 5): Finally, Test 5, conducted in environments with high saturation, further affirmed the accuracy of the algorithm. Despite the simplicity of the scene, our method continues to exhibit high accuracy, reinforcing its versatility under a variety of conditions.

Collective findings of these tests demonstrate the exceptional performance of our algorithm under various challenging conditions. Its ability to maintain high confidence and precision under low-light conditions, coupled with its robustness against occlusion and surface anomalies, sets it apart from existing algorithms. Moreover, its consistent accuracy in different environmental settings underscores its potential for diverse applications, from industrial quality control to surveillance. The superiority of our algorithm can be attributed to its sophisticated architecture and optimization strategies, which probably include advanced feature extraction capabilities and effective handling of environmental noise. Future work could explore further enhancements to improve its adaptability to even more diverse conditions and its efficiency in processing speed and computational resource utilization.

Through observation and in different scenarios, our method can obtain the best detection performance on the surface of metal parts with low complexity, reduce missed detection defects and false detection, improve the detection precision of our method, and detect small defects. The detection results are more reliable and accurate and contribute to the progress of metal part surface defect detection methods.

E. ABLATION EXPERIMENTS

To evaluate the function of different components in the proposed method, ablation experiments are carried out for verification.

As shown in Table 5, to refine object detection models to identify surface defects in metal parts, this study systematically integrates and evaluates several advanced architectural enhancements on a modified YOLOv5 framework. The incremental additions of the LSG model, Ghost's convolutional module, CBAM attention mechanism, and SIOU metric contribute uniquely to the model's performance, culminating in a comprehensive solution that significantly surpasses traditional approaches.

The initial step involved replacing the backbone network with the LSG model, focusing on reducing the semantic loss to enhance the model's precision and recall capabilities. This modification resulted in a 1% decrease in mAP(0.5:0.9), but a modest increase in recall (0.9%), a 0.2% increase in mAP (0.5) and a substantial improvement in precision (5.3%). The LSG model facilitates richer information about gradient combinations, effectively minimizing information loss and mitigating gradient confusion. This enhancement underscores the importance of a robust backbone to capture detailed features essential for accurate defect detection.

By optimizing the replacement of the convolutional module with the Ghost module in the neck, we observed a slight decrease in precision by 0.7% and mAP (0.5:0.9) showed a slight reduction of 0.2%. However, there was a corresponding increase in mAP (0.5) of 1.1% and an increase in the recall rate of 7.2%. This change not only optimized the network parameter scale, but also reduced computing resource consumption, demonstrating the efficacy of lightweight convolutional modules in maintaining, if not enhancing, the model performance while ensuring computational efficiency.

The addition of the CBAM attention mechanism further amplified the model's performance, leading to significant increases in precision (0.4%), mAP(0.5) (3.6%), and mAP(0.5:0.9) (0.1%), but a reduction in recall (0.9%).

CBAM optimizes the focus of the model, enabling more effective feature extraction and improving detection precision. This adjustment illustrates the transformative impact of attention mechanisms on refining the model's ability to discern and prioritize relevant spatial and channel characteristics for superior defect detection.

Integrating the SIOU metric into the model further improved precision (0.4%), recall (0.8%) and mAP(0.5) (0.1%) but reduced the mAP(0.5: 0.9) (0.4%). SIOU offers a more nuanced approach to bounding-box regression, enhancing the target detection capabilities of the model by providing a more accurate representation of object shapes and orientations. This enhancement solidifies the critical role of advanced geometric metrics in increasing the detection accuracy.

Cumulative addition of these four modules resulted in the following remarkable performance metrics: precision (98.3%), recall (93.8%), mAP(0.5) (95.6%), and mAP(0.5:0.9) (74.7%). This comprehensive integration not only showcases the individual contributions of each module but also highlights their synergistic effects, leading to significant improvements in training speed, inference precision, and practical performance. Collective implementation of these enhancements presents a compelling case for adopting multifaceted architectural innovations in object detection models.

The method used in this study to integrate cutting-edge enhancements into a modified YOLOv5 model demonstrates a significant leap forward in the detection of surface defects in metal parts. Each module, LSG, Ghost, CBAM, and SIOU brings different advantages to the table, resulting in a highly efficient, accurate, and practical model. These findings affirm the potential of combining various technological advancements to address the intricate challenges of object detection, paving the way for future research and development in this critical field.

To better verify the functionality of each component, we performed a more intuitive visual comparison, as shown in FIGURE 16.

To advance the capabilities of YOLOv5 to detect surface defects in metal parts, our study systematically incorporated a series of modifications aimed at improving the detection accuracy while optimizing the efficiency of the model. The progression of these modifications and their impact on model performance are meticulously documented, showing a clear trajectory of improvement in identifying defects with greater precision and confidence.

The starting point, depicted in FIGURE 16 (a), illustrates the original YOLOv5's capability, where it managed to detect only one flaw with a relatively moderate confidence level of 0.56, indicating room for substantial improvement, particularly in complex defect detection scenarios.

As shown in FIGURE 16(b), the introduction of the LSG model to replace the C3 network in the backbone marks the first step towards enhancement. This modification allowed the detection of two distinct types of imperfections, namely circles and voids, with improved confidence levels

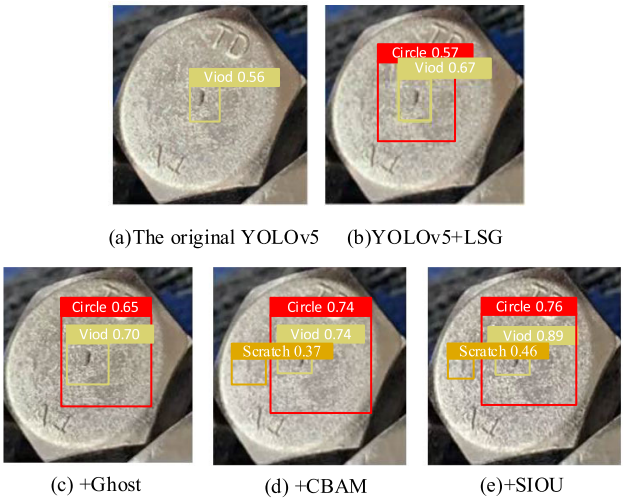

FIGURE 16. The visual results of the ablation study.

of 0.57 and 0.67, respectively. The ability of the LSG model to capture richer gradient information plays a pivotal role in this advancement, reducing semantic loss and enhancing the precision of the model.

Further refinement is achieved by substituting the Ghost module in the neck, as shown in FIGURE 16 (c). This adjustment leads to an increase in the detection precision for both circles and voids to 0.65 and 0.70, respectively, underscoring the effectiveness of the Ghost module in optimizing the network's parameter scale and computational resource consumption without compromising detection accuracy.

The integration of the CBAM attention mechanism, as illustrated in FIGURE 16(d), significantly increased the performance of the model. The detection count increased to three types of defects, with precision scores for circles, voids, and scratches of 0.74, 0.74 and 0.37, respectively. The CBAM improves the focus of the model on relevant features, further improving its ability to discern and accurately classify different types of defect.

Replacement of GIoU with SIoU, as shown in FIGURE 16(e), marks the culmination of our enhancements. This change specifically targets the reduction of loss between true and predicted values, culminating in superior detection accuracies of 0.76, 0.89, and 0.46 for circles, voids, and scratches, respectively. The SIoU metric introduces a more refined approach to bounding-box regression, closely aligning the predicted boxes with actual defect locations and shapes.

The sequential addition of these enhancements not only progressively improves the detection precision but also ensures that the detected defects are identified with increasing accuracy. Our approach significantly increased the performance of the model, achieving a delicate balance between high-precision detection and model efficiency. The results unequivocally validate the effectiveness of our proposed modifications, establishing our model as not only highly precise but also lightweight and efficient. Through strategic

architectural changes, our model achieves unparalleled precision, closely mirroring the true characteristics of the defects it aims to detect, thereby establishing a new benchmark in the field of surface defect detection in metal parts.

IV. CONCLUSION

To improve the efficiency and reliability of defect detection on metal surfaces, this study introduces a significant advance using an improved YOLOv5 model. The main contributions of this study are as follows. (1) Our method integrates the LSG and Ghost models to reduce semantic loss and optimize the network structure for superior performance. (2) The inclusion of CBAM enhances the inferencing capabilities and facilitates the fusion of multiscale feature, while the SIoU metric increases the precision in target detection. By expanding our data set to 8,569 instances using various processing techniques, we ensured comprehensive training and evaluation, covering a diverse range of defect types and scenarios for real-world applicability. The experimental results show that compared to the original yolov5, our proposed method has a significant improvement. Specifically, maps (0.5) increased by 5.7%, whereas FPS increased significantly by 21%. At the same time, LCG-YOLO has advantages over other models proposed in recent years in terms of mAP, detection speed, and model size.

The results of rigorous testing of the six distinct methods and five key indices demonstrate the improved performance of the proposed model. However, there remains substantial room for improvement in the detection of surface defects in metal parts. More research in the field of defect detection is necessary to achieve the following objectives:

(1) Advancing defect detection capabilities: Continuous innovation is essential to keep up with evolving manufacturing processes and the growing complexity of metal parts.

(2) Large-scale data sets and transfer learning: Build more abundant data sets on metal parts defects and use transfer learning and other methods to improve the generalizability and adaptability of the algorithm in different scenarios.

(3) Differential detection of defect types: In addition to common surface cracks, bubbles and other defects, many other types of defects need to be detected, such as deformation and discoloration. Further research can expand the ability of this algorithm to detect various types of defects.

To achieve these goals, it is necessary to significantly expand the comprehensive training of the data set and enhance the detection ability of the algorithm through structural and algorithm innovation. Further research is needed to improve feature extraction, optimize model parameters and accelerate the detection speed. Conducting research on advanced feature extraction techniques and streamlining inspection processes is critical to achieving impeccable accuracy and operational efficiency, in line with the broader goals of sustainable manufacturing and safety assurance in industrial environments.

REFERENCES

[1] L. Jie, L. Siwei, L. Qingyong, Z. Hanqing, and R. Shengwei, "Real-time rail head surface defect detection: A geometrical approach," in *Proc. IEEE Int. Symp. Ind. Electron.*, Seoul, Jul. 2009, pp. 769–774, doi: 10.1109/ISIE.2009.5214088.

[2] W. Wang, C. Mi, Z. Wu, K. Lu, H. Long, B. Pan, D. Li, J. Zhang, P. Chen, and B. Wang, "A real-time steel surface defect detection approach with high accuracy," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022, doi: 10.1109/TIM.2021.3127648.

[3] X. YongHua and W. Jin-Cong, "Study on the identification of the wood surface defects based on texture features," *Optik Int. J. Light Electron Opt.*, vol. 126, no. 19, pp. 2231–2235, Oct. 2015.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[6] S. Ren, K. He, and R. Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–11.

[7] N. Zhang, J. Donahue, and R. Girshick, *Part-Based R-CNNs for Fine-Grained Category Detection*. New York, NY, USA: Springer, 2014, doi: 10.1007/978-3-319-10590-1_54.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision—ECCV 2016*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37.

[10] J.-S. Lim, M. Astrid, H.-J. Yoon, and S.-I. Lee, "Small object detection using context and attention," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Jeju Island, South Korea, Apr. 2021, pp. 181–186, doi: 10.1109/ICAIIC51459.2021.9415217.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[12] C. Zehua and H. Junying, "YOLOv5 masks detection algorithm of lightweight improvement," *Software Tribune*, pp. 1–6, 2023.

[13] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, "Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles," 2022, *arXiv:2206.02424*.

[14] H. Qiang, "Improved YOLOv8 algorithm for small target detection research," Jilin Univ., Changchun, China, Tech. Rep., 2023, doi: 10.27162/dcnki.Gjlin.2023.001647.

[15] D. Li, L. Li, Z. Chen, and J. Li, "Shift-ConvNets: Small convolutional kernel with large kernel effects," 2024, *arXiv:2401.12736*.

[16] L. Wenfang, "Research on ultrasonic defect detection method of metal components based on machine learning," North Central Univ., Minneapolis, MN, USA, 2021, doi: 10.27470/dcnki.GHBGC.2021.000777.

[17] J. Wenluan, "Research on ultrasonic nondestructive testing method and application for defect identification of metal alloys," Lanzhou Jiaotong Univ., Lanzhou, China, Tech. Rep., 2022, doi: 10.27205/dcnki.Gltec.2022.000682.

[18] W. Han, L. Haiming, and S. Yuhong, "Research on metal surface defect detection based on improved YOLOv5 algorithm," *Mech. Sci. Technol.*, pp. 1–6, 2023.

[19] Y. J. Jiang, C. Li, and X. Zhang, "Surface defect detection of high precision cylindrical metal parts based on machine vision," in *Proc. Int. Conf. Intell. Robot. Appl.* Cham, Switzerland: Springer, 2021, pp. 810–820, doi: 10.1007/978-3-030-89098-8_76.

[20] W. Lin, H. Hongyu, and S. You, "Industrial metal surface defect detection based on computer vision review," *J. Automat.*, 2023, p. 24, doi: 10.16383/j.aasc230039.

[21] Y. Juan and L. Shun, "Detection method of illegal building based on YOLOv5," *Comput. Eng. Appl.*, vol. 57, no. 20, pp. 236–244, 2021.

[22] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.

[23] S. Woo, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 1–17.

[24] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[25] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
[26] U. Mittal, P. Chawla, and R. Tiwari, "EnsembleNet: A hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models," *Neural Comput. Appl.*, vol. 35, no. 6, pp. 4755–4774, Feb. 2023.
[27] A. M. Roy, J. Bhaduri, T. Kumar, and K. Raj, "WilDect-YOLO: An efficient and robust computer vision-based accurate object localization model for automated endangered wildlife detection," *Ecol. Informat.*, vol. 75, Jul. 2023, Art. no. 101919.
[28] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, Hong Kong, 2006, pp. 20–24.
[29] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.
[30] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.
[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, vol. 16, Las Vegas, NV, USA, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[32] Z. Zheng, "Distance-IoU Loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1–9.
[33] A. Tatsuma and M. Aono, "Food image recognition using covariance of convolutional layer feature maps," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 6, pp. 1711–1715, 2016, doi: 10.1587/transinf.2015edl8212.
[34] H. Tian-Xuan, X. Xin-Ge, and Z. Li-Zhen, "Coal and rock fracture image recognition method research," *Ind. Automat.*, pp. 1–7, 2023.
[35] L. Li, "Hyperband: Bandit-based configuration evaluation for hyperparameter optimization," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–9.

JIANGLI YU was born in Longhua, Hebei, China, in 1980. He received the master's degree from Beijing University of Civil Engineering and Architecture. He is currently with Hebei University of Architecture. His research interests include building intelligence and smart city.

XIANGNAN SHI is currently pursuing the degree in electrical engineering and automation with Hebei University of Architecture, Zhangjiakou, China. Her research interests include machine learning and artificial intelligence.

WENHAI WANG was born in Yangyuan, Zhangjiakou, Hebei, in 1994. He received the Ph.D. degree from South China Normal University, in 2023. He has been engaged in the study of luminescence properties of carbon quantum dots and the exploration of photoelectronic properties of two-dimensional materials.

YUNCHANG ZHENG received the B.S. degree in electronic science and technology and the M.S. degree in signal and information processing from the University of Electronic Science and Technology of China, in 2013 and 2016, respectively. Since 2017, he has been a Lecturer with the College of Electrical Engineering, Hebei University of Architecture, Zhangjiakou, China. His research interests include image processing, deep learning, and artificial intelligence.