

## RESEARCH ARTICLE

# Rapid Detection of PCB Defects Based on YOLOx-Plus and FPGA

YAJING PAN<sup>1</sup>, LEI ZHANG<sup>2</sup>, AND YUJIE ZHANG<sup>2</sup><sup>1</sup>Hangzhou Normal University, Hangzhou 311121, China<sup>2</sup>Hebei University of Technology, Tianjin 300401, China

Corresponding author: Lei Zhang (houyupeng@tbea.com)

**ABSTRACT** During the production process, Printed circuit boards (PCBs) may encounter many defect issues which can severely affect the functionality of the circuit. However, existing PCB defect detection methods suffer from low detection accuracy and slow detection speed. Considering the requirements of PCB factories for detection accuracy and real-time performance This paper first made structural improvements to the existing YOLOx defect detection algorithm, introducing PAN+FPN, SimAM, and SIoU modules to improve the detection accuracy of the algorithm, and named it YOLOx-Plus. Then, algorithm acceleration is achieved by quantifying network parameters and designing FPGA accelerators. In the experiment, the average detection accuracy of YOLOx-Plus is 93.2%, the network loss is reduced by 1.094, the model size is compressed by 64%, detection speed is improved by 68.1%, and the FPS reaches 72.6. The experimental results show that the proposed PCB defect detection method based on YOLOx-Plus and FPGA can efficiently detect typical defects in PCB boards, overcome the limitations of existing methods, and have a wide range of practical applications.

**INDEX TERMS** PCB, defect detection, FPGA, YOLOx-Plus.

## I. INTRODUCTION


A printed circuit board (PCB) is an important component in the electronics industry, as its quality significantly impacts the functionality and lifespan of electronic products. With the development of electronic manufacturing, electronic products are becoming more intelligent and smaller in size, which makes PCB product design and wiring increasingly complex [1]. PCB production usually involves over ten different processes, and although strict control measures are implemented, it cannot guarantee zero defects. Common defects may include short circuits, open circuits, burrs, and missing holes [2]. These defects are mostly found within the intricate PCB circuits, and their colors are similar to the background, with various types. This not only increases the difficulty of detection but also makes it easy to experience false positives or missed detections during the testing process.

However, the production process of PCBs is complex and prone to defects such as soldering flaws, open circuits, short circuits, and imperfections. The characteristics of these

defects are as follows: (1) PCB designs vary, leading to a complex background environment for PCB defect detection; (2) Defects are often small in size and have similar colors to the background, increasing the difficulty of detection [3]; (3) There are multiple types of defects, and their features overlap, making it challenging to differentiate them [4]. These characteristics contribute to increased difficulty in detection, resulting in higher chances of false positives and false negatives.

### A. COMPUTER VISION DETECTION METHODS BASED ON ARM/GPU

Computer vision technology encompasses knowledge in mathematics, computer science, electronics, and other fields. It is a significant branch in the domain of deep learning. It involves various areas such as image processing, PC applications, pattern recognition, signal processing, and artificial intelligence. It serves as a crucial tool in the field of industrial inspection today [5]. Due to the limitations of the aforementioned defect detection methods, utilizing visual inspection technology has become the current main-stream approach for PCB defect detection and also represents the developmental

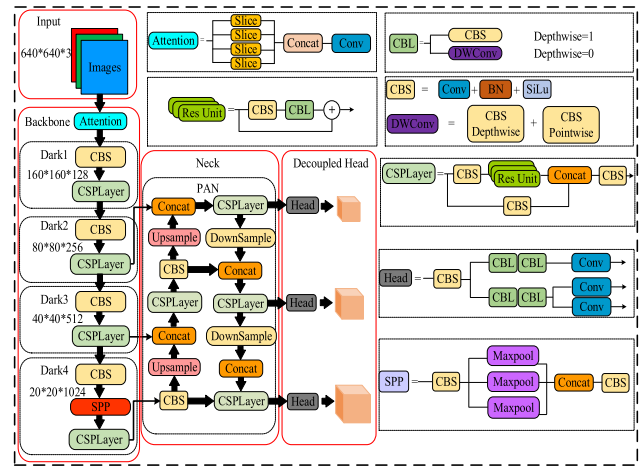
The associate editor coordinating the review of this manuscript and approving it for publication was Mario Donato Marino .

trend in PCB research. Computer vision is mainly implemented through neural network algorithms. However, neural network algorithms usually involve a huge number of calculations and require detection equipment to have powerful computing capabilities. ARM-based embedded devices have weak image processing capabilities, slow detection speeds, and are prone to lags during batch detection, making it difficult to meet the needs of rapid on-site detection. With the continuous development of neural networks, traditional ARM devices have been unable to meet the computing needs of today's highly complex neural networks. Although high-end GPU can support high throughput of convolution operations and has fast detection speed [6], it has problems such as excessive power consumption and poor portability, making it difficult to meet the actual inspection requirements of large-volume inspection sites in PCB factories.

**B. COMPUTER VISION DETECTION METHODS BASED ON FPGA**

FPGA has developed rapidly in recent years and attracted much attention. It is currently a popular direction for using hardware to implement deep learning. Computer vision has the characteristics of complexity, high speed, and gridding, which are very similar to the characteristics of FPGA for image processing. In addition, structurally, FPGA is equipped with freely changeable logic blocks, and the wiring of connection points can be changed through programs to achieve all functions. required logical functions. The FPGA defect detection method uses FPGA as the detection platform, and completes the deployment of the detection algorithm in the FPGA through a series of logic designs [7]. By matching relevant peripherals, the detection task can be realized, with the characteristics of fast detection speed and high detection accuracy. With the continuous development of EDA technology, FPGA can exert its flexibility and hardware parallelism to a greater extent, which plays a very important role in seizing new product markets and shortening the development cycle [8]. In recent years, more and more methods have been used to implement defect detection using FPGA, and they have dominance that traditional detection methods do not have. With its flexible programmable configuration features, powerful parallel computing capabilities, ultra-low power consumption, and portability, FPGA has become an ideal choice for completing modern industrial defect detection tasks.

Among the mainstream algorithms for computer vision, the Mask R-CNN algorithm is more effective in terms of detection accuracy, but has a larger amount of computation and slower detection speed. The SSD algorithm is more effective in terms of detection speed, but is less effective in terms of detection accuracy and is less effective in terms of deployment at the FPGA side. The YOLO algorithm is widely used, improvable, and adaptable, and has the potential to achieve highly efficient detection [9]. However, although the previous YOLO PCB defect detection methods based on FPGA met the basic requirements in detection speed, there



**FIGURE 1. The network structure of YOLOx.**

were significant defects in detection accuracy, so they could not meet the actual detection needs. The method proposed in this paper can overcome the difficulties currently faced in this field, which improves the YOLOx algorithm and designs an FPGA accelerator to achieve accurate and rapid detection of defects on industrial PCB boards. It solves the problems faced by PCB board defect detection in the current market, reduces detection costs, and improves detection efficiency.

**II. INTRODUCTION OF ALGORITHMS**

**A. YOLOx**

YOLOx is a deep learning algorithm in the YOLO series, introduced in 2021. YOLOx is the target detection network of YOLO series with better detection effect, which is the product of inheritance and innovation on the basis of YOLOv3, YOLOv4, and YOLOv5 [10]. It differs from other algorithms by adopting an anchor-free approach, which reduces the number of predicted results and completes initial screening. The network structure of YOLOx is shown in FIGURE 1.

YOLOx consists of three parts: Backbone Network, Neck Feature Enhancement Network and Classification Prediction Network. FIGURE 1 shows the structure of YOLOx, the main modules in the backbone network of the YOLOx are Attention, CBS, CSPLayer, and SPP, which are stacked together. Among them, the main purpose of the Attention module is to reduce the number of parameters in the network, improve the speed of forward and backward propagation, and at the same time it can retain more image information modules. The CBS module is the basic constituent unit of YOLOx, which consists of the Conv2D convolution function, the BN (Batch Normalization) algorithm, and the SiLU activation function, and it can sequentially carry out the 2D convolution of the input target, regularization and activation operations on the input target in order to realize basic image processing [11]; The CSPLayer module is mainly composed of the CSPNet network structure, which splits the residual block into 2 parts, one part as the residual backbone for residual stacking, and the other part as the residual edges, which are

finally spliced together after a small amount of processing, and used to alleviate the problem of disappearing gradient and improve the training effect of the model [12]; the SPP module is responsible for performing convolution and pooling operations on the input to perform convolution and pooling operations, the convolution layer uses the basic convolution unit, and the pooling layer uses the maximum pooling method with different sizes of pooling kernels for feature extraction to improve the sensory field of the network.

## B. YOLOx-PLUS

Although YOLOx has strong advantages in the field of target detection, there are still challenges for targets such as PCB defects, which are difficult to distinguish with the naked eye. In order to solve the existing problems of PCB defect detection difficulty and low accuracy, the paper proposes a PCB defect detection algorithm based on YOLOx-Plus network, which has three main contributions:

- In order to reduce the error of the network input data, the data enhancement is optimized by improving the weak enhancement.
- In order to solve the problems of PCB image background similarity, texture replication leading to feature extraction difficulties and large loss of feature information in network propagation, and to strengthen the ability of effective feature recognition, a generalized parameter-free attention SimAM (Simple Attention Mechanism) module is inserted into the backbone network. The SimAM method utilizes the energy function to identify valid features and suppress irrelevant features [13].
- In order to obtain more higher-order semantic information, improve the discriminative power and increase the fusion interaction ability of the feature fusion network, the CSPHB module is used in the feature fusion network, which enhances the fusion of different scales and improves the spatial interaction of the downstream tasks.

YOLOx-Plus is improved in YOLOx, and its network structure consists of several parts such as Input, Backbone, Neck and Prediction using Decoupled Head.

- **Input:** The role of the input side of YOLOx-Plus is to input images, data enhancement and some other preprocessing operations. In this part, it mainly improves the data enhancement, and the improved data enhancement adopts the optimized data enhancement, the weakened data enhancement to exclude the error and improve the detection effect.

- **Backbone network:** The improved backbone network Backbone consists of Attention, Darknet53 that adds parameter-free attention, SPP, and other structures. The attention structure is proposed in YOLOv5. The attention layer slices the original input image of size  $640 \times 640 \times 3$  into smaller segments, performs a splicing operation on them, followed by a convolution operation, resulting in a final feature map of size  $320 \times 320 \times 64$ . This method can reduce the information loss caused by down-sampling. Adding Parameter-Free Attention. The Darknet53 backbone consists of CBL, SimAM parameter-free attention and CSPLayer. The

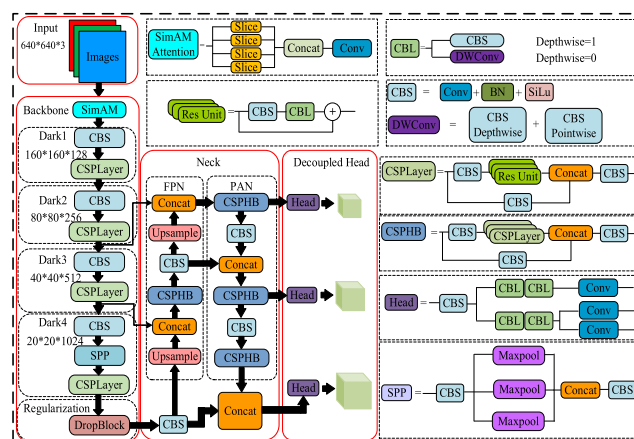


FIGURE 2. The network structure of YOLOx-Plus.

CSPLayer structure is mainly composed of the components CBS (convolutional layer + normalization layer + activation function) and residual component (Containing both CBS and CBL). This network structure helps to improve the feature extraction and fusion capabilities of the network.

In addition, in the process of training neural networks, overfitting scenarios often occur, and it is necessary to regularize the neural network to suppress overfitting. DropBlock is a regularization method for convolutional neural networks. DropBlock zeros adjacent elements in the feature map, allowing the model to extract information from non-adjacent regions, making it more suitable for convolutional layer regularization. The purpose of zeroing is to reduce the situation where certain neurons are only effective in the presence of special neurons, that is, to reduce the complex co-occurrence relationships between neurons and enhance the robustness of the network. DropBlock defines the size of the feature map zeroing block as  $B_s$ . If the number of activation units to be deleted is defined as  $y$ , the survival probability of activation units is  $p$ , and the feature map size is  $F_s$ . As is shown in Equation (1).

$$y = \frac{1-p}{B_s} \frac{F_s}{(F_s - B_s + 1)^2} \quad (1)$$

- **Neck and Decoupled Head:** The feature fusion network of the model fuses strong semantic features and strong localization features by using the feature pyramid FPN+PAN structure. The FPN (Feature pyramid network) structure passes deep semantic features to shallow layers in a top-down manner to enhance semantic expression at multiple scales, while the PAN (Path aggregation network) structure passes shallow localization information to deep layers to enhance localization ability at multiple scales, combining both FPN and PAN will be better for the feature enhancement of the network. In the final prediction part, the number of prediction results is reduced by the anchorless way to complete the preliminary screening, and then the SimOTA algorithm is used to refine the screening of the prediction results to get the final prediction results.

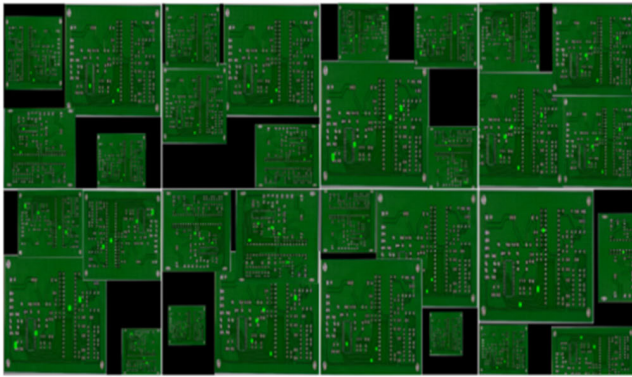


FIGURE 3. Mosaic enhanced images in YOLOx.

YOLOx-Plus has a more complex network structure than YOLOx. Firstly, YOLOx-Plus introduces the SimAM module in the Backbone, which allows the network to improve model training without increasing the number of parameters. Secondly, its neck becomes a complex structure composed of two feature enhancement networks, FPN and PAN, which can significantly improve the feature fusion and feature extraction capability. Finally, YOLOx-Plus improves the existing feature enhancement network by adding the CSPHB module, which can be used to acquire higher-order features and improve the spatial interaction of the downstream tasks, fusing features of different scales, and improving the predictive classification results of the model.

### III. ALGORITHM IMPROVMENT

#### A. OPTIMIZED DATA ENHANCEMENT

The YOLOx data enhancement method uses Mosaic and Mixup. Mosaic selects four different images and performs data augmentation operations such as flipping, scaling, color gamut change, etc., and arranges the images to splice them together, so that the new images contain labeled boxes. Mixup is an algorithm that performs mixing augmentation on images, and the core of the algorithm lies in mixing the images between different classes to achieve the effect of expanding the training dataset, and to a certain extent, the batch size is also enlarged, which can make the model training achieve better results. achieve the effect of expanding the training dataset, to a certain extent, it also expands the batch size, and the amount of input data is larger, which can make the model training achieve better results.

The last 15 rounds of training in YOLOx turn off data augmentation to remove the large number of inaccurate images introduced due to Mosaic, while Mixup changes the distribution of the training data. Turning off the last 15 epochs allows the detector to accomplish final convergence on real-world data without the errors introduced by Mosaic.

However, it is found that YOLOx setting Mosaic enhancement still introduces a large number of blank images without detectors, as shown in FIGURE 3.

In order to solve this problem, this paper optimizes the data enhancement by using weakening data enhancement, that sets

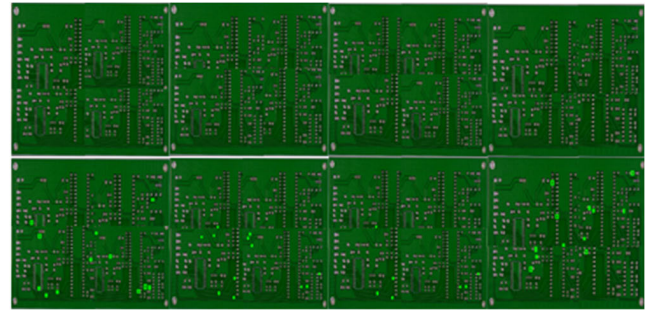


FIGURE 4. Weakening data enhancement in YOLOx.

the Mosaic enhancement mode in the Mosaic image. This way can reduce the rotation angle and scaling size of the Mosaic image in the process of generating the image, and also reduce the number of blank images without detectors, as shown in FIGURE 4.

#### B. PARAMETER-FREE ATTENTION MODULE

PCB defects suffer from the problem of small and diverse targets, which requires a certain feature recognition capability of the algorithm. SimAM is a parameter-free SimAM is a parameter-free 3D attention module. SimAM module does not need to add parameters to the original network, compared to channel attention (1D attention) which focuses only on channel importance, and spatial attention (2D attention) which focuses on channel importance. Compared to channel attention (1D attention), which focuses only on channel importance, and spatial attention (2D attention), which focuses only on spatial location importance, 3D attention can focus on each channel simultaneously. In contrast to channel attention (1D attention), which focuses only on channel importance, and spatial attention (2D attention), which focuses only on spatial location importance, 3D attention focuses on the importance of each channel and spatial location feature simultaneously. The 3D attention can focus on the importance of each channel and spatial location feature at the same time, and the 3D attention weights are inferred by analyzing the feature map-ping (Both channel and spatial location importance) [13].

The implementation of SimAM three-dimensional weights is based on a theory of visual neurology: in visual neurology, information-rich neurons usually have different firing patterns from their surrounding neurons and can produce spatial depression on neighboring neurons. Neurons that show spatial depression effects in visual processing tasks should then be given higher weights, and the simplest implementation to find these neurons is to measure the linear separability between a target neuron and other neurons.

Based on these scientific findings, define an energy function for each neuron number. Input features of the input image  $R, X \in R^N(C \times H \times W)$ , existence of  $C$  passages.  $M=H \times W$  neurons ( $H, W$  are the height and width on the feature map,  $C$  is the number of channels on the feature map). Then theoretically, each channel has menergy functions, which can

be calculated by Equation (2) and (3) for a single channel of all neurons  $x_i$  with average mean  $\mu$  and variance  $\sigma$ , which reduces the iterative computation, avoiding the calculation of mean and variance at each position.

$$\hat{\mu} = \frac{1}{M} \cdot \sum_{i=1}^M x_i \quad (2)$$

where,  $\hat{\mu}$  is average value,  $M$  is the number of energy functions, and  $x$  is neurons.

$$\hat{\sigma}^2 = \frac{1}{M} \sum_{i=1}^M (x_i - \hat{\mu})^2 \quad (3)$$

where,  $\hat{\sigma}^2$  is variance,  $M$  is the number of energy functions,  $x$  is neurons, and  $\hat{\mu}$  is average value.

Therefore, the final minimum energy function is shown in Equation (4).

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \quad (4)$$

where,  $e_t^*$  is the minimum energy function,  $t$  is the target neuron, and  $\lambda$  is the hyperparameter. From Equation (4), it is clear that SimAM can define the energy function for each neuron by defining the linear divisibility of each neuron of the network and calculating the minimum energy  $e_t^*$ . The lower the  $e_t^*$ , the lower the energy, and the more different the neuron  $t$  is from the surrounding neurons, and the higher the importance of the neuron  $t$  is. The lower the energy of  $e_t^*$ , the more different the neuron  $t$  is from the surrounding neurons and the higher its importance.

Finally, based on the gain effect of neuron response, scaling operations are applied for weighting. At the same time, the sigmoid function can also limit the excessive value of (features grouped according to the importance of neurons), without affecting the relative importance of each neuron, as shown in Equation (5).

$$\hat{X} = \text{sigmoid}\left(\frac{1}{E}\right) \odot X \quad (5)$$

where,  $\hat{X}$  is the gain effects of neuronal responses, “*sigmoid*” is activation function,  $E$  is the importance grouping of the minimum energy  $e_t^*$ , and  $\odot$  represents multiplication calculation by elements [14].

For image feature recognition, both channel and spatial location features are quite important, in order to effectively and comprehensively evaluate the importance of channel and spatial location features, this paper introduces the parameter-free attention SimAM into the backbone network Darknet53, which can evaluate the features extracted from the Backbone through the energy function without increasing the parameters of the model, and find out the neurons that are rich in information of higher importance. information-rich neurons, neurons with higher importance. The improved model can effectively find out the important features, inhibit the interference of irrelevant features, and improve the feature representation ability and modeling ability of the network. The improved model can effectively identify important features, suppress irrelevant feature interference, and improve

the feature representation ability of the network and the target localization ability of the model.

### C. SIOU LOSS FUNCTION

The loss function is a mathematical function used to compute the difference between predicted values and actual values. In the YOLOx, the role of the loss function is to calculate the error between the predicted results obtained during each iteration of the training process and the ground truth results. The loss function is commonly used to guide the subsequent training of neural networks in a more accurate direction. The quality of the loss function largely determines the accuracy of defect localization. Traditional loss functions for defect detection mainly include IoU (Intersection over Union) and its variants, such as DIOU, CIOU, GIOU, and ICIOU [15]. These loss functions aggregate regression metrics of bounding boxes, such as the ratio of intersection to union, overlapping area, and distance, to achieve accurate localization.

YOLOx selects CIOU as its loss function, which has a certain improvement in accuracy compared to previous loss functions. However, CIOU does not consider the issue of misalignment between predicted boxes and ground truth boxes. This shortcoming can lead to reduced learning efficiency, slower model convergence speed, and ultimately poorer model performance. Therefore, this paper chooses to replace the original CIOU loss function with the SIOU loss function. The SIOU loss function computes the angle between the required regressions and redefines the penalty metric. By applying SIOU to the neural network of YOLOx, training speed and inference accuracy can be further improved. The SIOU loss function consists of three cost functions: IoU Cost, Angle Cost, and Distance Cost. SIOU primarily includes three components of loss: IoU loss, angle loss, and distance loss.

The IoU loss refers to the intersection over union ratio between the predicted box and the ground truth box. The calculation method of this loss is shown in Equation (6).

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \quad (6)$$

where,  $B$  is prediction bounding box,  $B^{gt}$  is true bounding box.

The function of angle loss is to make the prediction box return to the same horizontal or vertical line as the real box, reducing this loss can accelerate the convergence of the network model. The formula for calculating angle loss is shown in Equation (7).

$$\Lambda = 1 - 2 * \sin^2\left(\arcsin(x) - \frac{\pi}{4}\right) \quad (7)$$

where,  $\Lambda$  is angle loss, “*sin*”, “*arcsin*” is trigonometric function.

The distance loss considers the angle loss parameter, and its function is to reduce the loss by controlling the minimum center distance between the prediction box and the target box to move towards the target box when the target box and

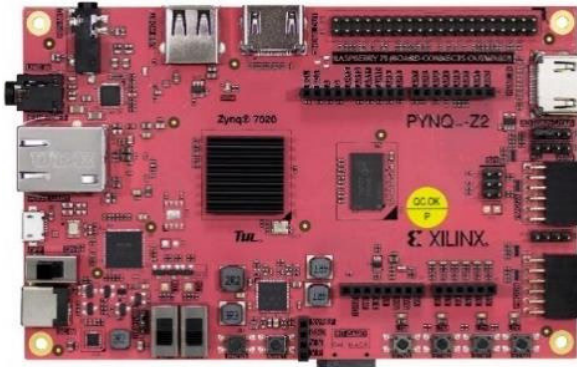


FIGURE 5. The platform of PYNQ-Z2 FPGA.

prediction box do not overlap [16]. The calculation method for distance loss is shown in Equations (8) ~ (13).

$$\Delta = \sum_{t=x,y} (1 - e^{-\gamma \rho^t}) \quad (8)$$

$$\gamma = 2 - \Delta \quad (9)$$

$$\rho_x = \left( \frac{b_{cx}^{gt} - b_{cx}}{c_w} \right) \quad (10)$$

$$\rho_y = \left( \frac{b_{cy}^{gt} - b_{cy}}{c_h} \right) \quad (11)$$

$$c_w = \max(b_{cx}^{gt}, b_{cx}) - \min(b_{cx}^{gt}, b_{cx}) \quad (12)$$

$$c_h = \max(b_{cy}^{gt}, b_{cy}) - \min(b_{cy}^{gt}, b_{cy}) \quad (13)$$

where,  $\Delta < 2$ ,  $b$  is the center point coordinates of the predicted box,  $b^{gt}$  is the center point coordinates of the real box,  $\rho$  is the distance between the center points of the two boxes,  $\gamma$  is an intermediate parameter,  $c$  is the square of diagonal length of minimum bounding rectangle of the two boxes, and  $w, h$  is the width and height of the rectangular box.

Based on the above losses, the calculation method of SIoU is shown in Equation (14).

$$SIoU = 1 - IoU + \frac{\Delta}{2} \quad (14)$$

The newly obtained SIoU defines a new penalty metric and incorporates the vector angle between the required regressions into the calculation range. Therefore, using the SIoU loss function can enable YOLOx-Plus to have better performance in defect detection tasks than using the CIoU loss function.

#### IV. ALGORITHM ACCELERATION

##### A. ACCELERATOR DESIGN PLATFORM

FPGA is a new type of programmable logic device following the advent of programmable array logic (PAL), general array logic (GAL), complex programmable logic (CPLD) and other devices. As a semi-customized circuit, it not only inherits the advantages of a fully customized application-specific integrated circuit (ASIC), but also improves and optimizes it to further increase the number of logic gates and can be used to design more complex circuits [17].

TABLE 1. The hardware resource of PYNQ-Z2.

PYNQ-Z2 Resources	Model/Interface
Master Chip	ZYNQ 7020
Memory	512 MB DDR3
Storage	Micro SD
Video Interface	HDMI IN & HDMI OUT
Audio Interface	ADAU1761 Codec with HP
Network Interface	10/100/1000M Ethernet
Expansion	USB Host

Neural networks require a huge amount of computation and are difficult to run on embedded devices without optimization. PYNQ-Z2 is the first FPGA hardware development platform compatible with Python. It has rich logic resources and supports the deployment of deep learning. With PYNQ-Z2, developers can use hardware libraries to import low-level drivers and programs through its application programming interface (API) [18]. Developers can directly complete the top-level design of the algorithm on the Jupyter Notebooks integrated development environment and call the underlying IP core to achieve the required design tasks.

##### B. PARAMETER QUANTIFICATION OF ALGORITHMIC NETWORKS

In order to reduce the resource requirements of the YOLOx-Plus algorithm network for PYNQ-Z2 and improve the detection speed of the algorithm, this paper performs fixed-point quantification of network parameters after the algorithm improvement is completed. Since conventional quantization methods are difficult to maintain a balance between accuracy and data volume, this section improves conventional quantization methods to ensure that the algorithm network reduces the amount of data and resource requirements without significant loss in accuracy, thereby further improving the performance of the algorithm on FPGA [19]. This paper uses the quantization-after-retraining method. This method uses a linear model to quantify the floating-point data in the algorithm network. First, the zero-point offset and scale scaling factor are calculated based on the maximum difference method to complete the approximation of the floating-point numbers [20]. Secondly, the algorithm network is continuously fine-tuned through re-training. Parameters such as weights and biases; finally, the quantification is completed and the accuracy loss is controlled within the ideal range. The calculation formulas of this quantification method are shown in Equations (15) and (16).

$$F = S(Q - Z) \quad (15)$$

$$S = \frac{\max - \min}{2^n} \quad (16)$$

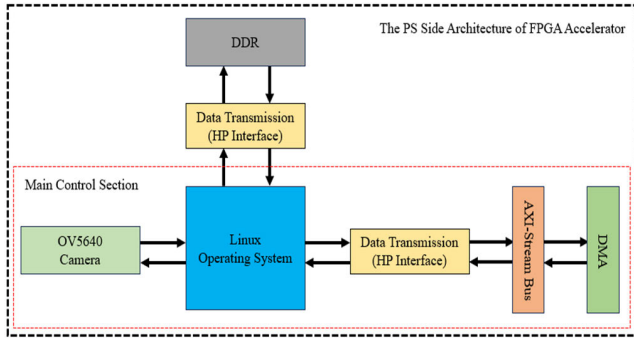


FIGURE 6. The PS architecture of FPGA accelerator.

where,  $F$  is the floating point data before quantization,  $S$  is the quantization scale factor, and the calculation method is the maximum difference method,  $Q$  is the quantized fixed-point data, and  $Z$  is the zero-point offset [21].  $\max$  and  $\min$  represent the maximum and minimum values in the sample respectively, and  $n$  is the quantization bit width. For zero-point offset  $Z$ , its calculation method is shown in Equations (17) and (18).

$$Z = f\left(-\frac{\min}{S}\right) \quad (17)$$

$$f(x) = \begin{cases} 0, & x < 0 \\ [x], & 0 < [x] < 2^n - 1 \\ 2^n - 1, & x > 2^n - 1 \end{cases} \quad (18)$$

where,  $[x]$  is rounding  $x$ ,  $n$  is the quantization accuracy.

The algorithm network will continue to reduce quantization errors and improve detection accuracy through the retraining process. The parameter calculation during the retraining process is mainly done through the loss function. This process continuously updates the parameters, that is, by derivation of the loss function, the gradient is reduced to a stable value, and finally the optimal weight and bias parameters can be found. The calculation process is shown in Equations (19), (20) and (21).

$$Y = X_1 W_1 + X_2 W_2 + \dots + X_i W_i + X_n W_n \quad (19)$$

$$y = f(Y) \quad (20)$$

$$Loss = \sum_{i=1}^n (y_i - truths_i)^2 \quad (21)$$

where, in the above formula,  $X_i$  is a single layer feature data,  $W_i$  is a single convolution kernel data,  $Y$  is a single convolution operation result, and  $y$  represents a single inference result.  $f$  is the activation function,  $truths$  is the correct and valid training data, and  $Loss$  represents the network loss [22].

### C. FPGA ACCELERATOR DESIGN

Designing algorithmic network accelerators is an important part of achieving rapid detection of FPGAs. The PS side of the algorithm network FPGA accelerator is one of the two main parts of the rapid defect detection system and is

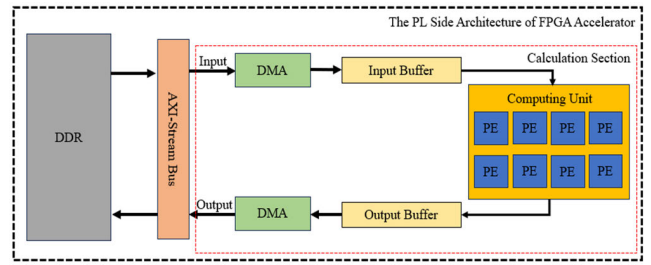


FIGURE 7. The PL architecture of FPGA accelerator.

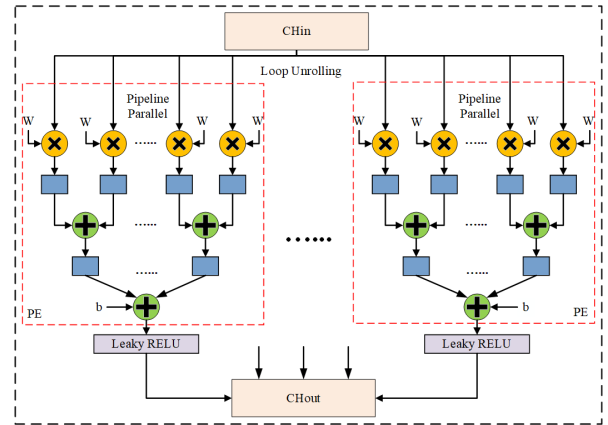


FIGURE 8. The architecture diagram of convolutional parallel computing.

TABLE 2. The pseudocode convolution parallel operation.

Convolution Parallel Operation
Input: Input Feature Data
Output: Output Feature Data
1: On-chip data computation
2: #pragma HLS ARRAY_PARTITION variable = cache_out complete dim = 1
3: #pragma HLS ARRAY_PARTITION variable = cache_in complete dim = 14: #pragma HLS ARRAY_PARTITION variable = kernel complete dim = 1
5: Out_Row: // Output Feature Data Rows
3: L1: for (row = 0; row < C2; row++) do
4: Out_Col: // Output Feature Data Cols
5: L2: for (col = 0; col < C2; col++) do
6: Out_Channel: // Output Feature Data Channels
7: L3: for (m = 0; m < M2; m++) do
8: #pragma HLS PIPELINE
9: In_Channel: // Input Feature Data Channels
9: L4: for (n = 0; n < M1; n++) do
10: #pragma HLS UNROLL
11: Kernel_Row: // Convolutional Kernel Rows
12: L5: for (k_row = 0; k_row < K2; k_row++) do
13: #pragma HLS UNROLL
14: Kernel_Col: // Convolutional Kernel Cols
15: L6: for (k_col = 0; k_col < K2; k_col++) do
16: if (n == 0 && k_row == 0 && k_col == 0)
17: cache_out [m][row][col] = bias[m]; //Bias
18: Convolution calculation // Matrix Operation
19: if (n = M1 - 1 && k_row == K2-1 && k_col == K2-1)
20: Leaky ReLU Activation // Activation function
21: .....

responsible for completing tasks with many control tasks and a small amount of calculation. The PS side consists

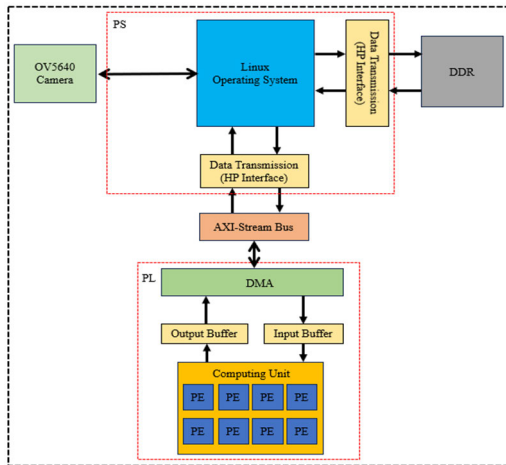


FIGURE 9. Overall architecture of FPGA accelerator.

of Linux system, AXI bus, data transmission interface and image sensor [23]. The PS architecture of FPGA accelerator is shown in FIGURE 6.

The PL end of the algorithm network FPGA accelerator is another main part of the rapid defect detection system and is responsible for completing work with few control tasks and large amounts of calculations. The PL side of the FPGA is responsible for more than 90% of the operations of the defect detection system [25]. The design of the PL side determines the detection speed of the detection system. The PL end of the algorithm network FPGA accelerator is composed of on-chip buffers (including input buffers and output buffers), computing units (Processing Elements, PE), and DMA transfer units [26]. In this section, we will mainly introduce the data cache optimization design and parallel computing design in PL side research and design. The PL side architecture of the algorithm network FPGA accelerator is shown in FIGURE 7.

FPGA accelerators also need to consider data calculation methods when designing. This paper uses convolutional parallel and pooled parallel computing designs. This paper first fuses the convolution module with the activation function to reduce the occupation of cache resources, and then uses pipeline parallelism to implement con-volution parallel operations, as is shown in FIGURE 8.

In the above pseudocode, the `#pragma HLS PIPELINE` pipeline parallel instruction is used to perform circular pipeline optimization on the feature data channel, and the `#pragma HLS UNROLL` instruction is used to perform loop expansion of Channel, Row, and Col [27].

According to the idea of software and hardware co-design, the PS side and PL side of FPGA are combined to form a complete PCB defect detection FPGA accelerator.

FPGA accelerator consists of the following two parts:

- Processing and control part: In the design, the small computational tasks such as processing and control included in the PCB defect detection algorithm are completed by the PS side. The main control system includes the following con-tents: Configuring the image sensor OV5640, and

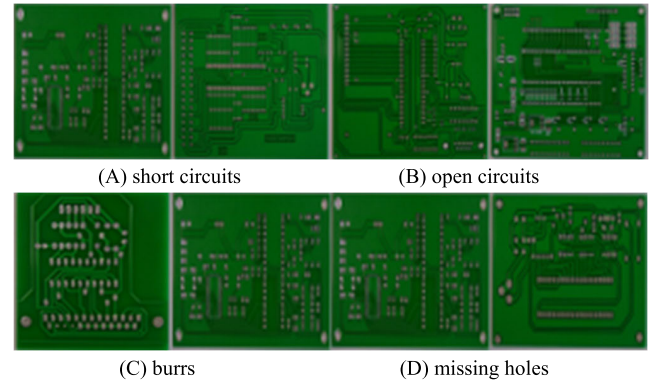


FIGURE 10. The dataset of PCB.

preprocessing the image data collected by the image sensor as input to the algorithm; detection algorithm for-ward reasoning code, as well as related interface configuration, bus configuration, data transmission and Instruction transmission module; image post-processing operations, result display, performance analysis, etc [28].

- Parallel operation part: In the design, the computationally intensive tasks such as convolution and pooling operations included in the defect detection algorithm are completed by the PL side. The computing system first inputs the data that needs to be calculated to the PL side through the AXI bus through DMA and the data buffer area, then uses the logic resources of the PL side to implement fast calculations, and outputs the results to the data buffer area, and then sends them back to the PS side through DMA for processing [29].

## V. EXPERIMENT PREPARATION

### A. TRAINING AND DETECTION ENVIRONMENT

The algorithm model training environment in this paper is a workstation using the Windows 10 operating system. The CPU model is i5-12400F, the GPU model is GTX 3080Ti, the video memory size is 16GB, and the video memory size is 16GB. All models used are based on Python 3.9.1, Cuda version 10.1 and CUDNN version 7.6.5. The test platform of this paper is PYNQ-Z2 FPGA. Before board testing, part of the hardware design project needs to be completed in advance on the PC side using the Vivado HLS high-level synthesis tool, and the entire system construction and debugging must be completed on the Vivado 2018.2 software platform [30]. The defect detection algorithm is YOLOx-Plus, the deep learning framework is Pytorch, the top-level design language is Python, and the running and testing environment is Jupyter Notebooks.

### B. THE DATASET OF PCB

The original dataset of PCB defects used in this paper contains 4,000 images. After data enhancement, the effective number of samples in the dataset was expanded to 8,000, with 4 types of defects: short circuits, open circuits, burrs, missing holes. However, in order to further im-prove the training effect of the model and prevent the network from falling into



TABLE 3. The parameters settings of training network.

Parameter	Value
Image size	640×640
Iterations	300
Weight decay	$5 \times 10^{-4}$
Learning rate	$4.250 \times 10^{-5}$
Learning momentum	0.8
Batch size	32

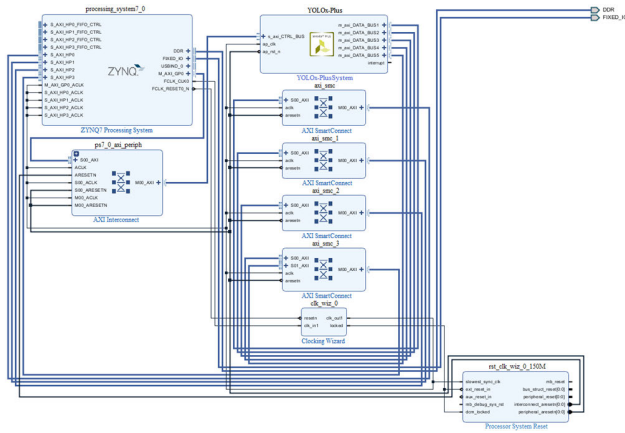


FIGURE 11. The hardware logic circuit of FPGA.

an overfitting state, this article expands the image dataset by rotating, flipping, and other operations on the defect images in the dataset to further increase the data volume. The final data is randomly divided in 9:1 ratio between the training set and the test set, with 7200 images in the training set and 800 images in the test set. As shown in FIGURE 10.

C. MODEL TRAINING

To ensure the fairness and reliability of the experiment, a uniform input size of  $640 \times 640$  is adopted. The stochastic gradient descent (SGD) optimizer is used for parameter optimization, with 300 epochs of model training. The weight decay of the optimizer is set to  $5 \times 10^{-4}$ , the learning rate is set to  $4.250 \times 10^{-5}$ , and the learning momentum is set to 0.8. The batch size is set to 32 according to the experimental platform [31]. The specific parameters settings of the training network are shown in TABLE 3.

D. FPGA LOGIC VERIFICATION

The design verification of the rapid defect detection system is mainly divided into three parts: parallel operation syntax verification in Vivado HLS; Rationality verification of system layout and wiring in Vivado; Verification of algorithms in the integrated development environment. The specific design verification steps are as follows: First, use HLS to verify the parallel operation syntax, including Testbench, logic verification and Modelsim timing verification [32]. Secondly,

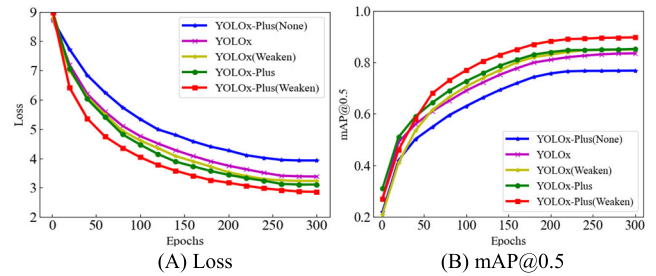


FIGURE 12. The comparison of data enhancement effects.

TABLE 4. The comparative performance of different YOLO methods.

Method	mAP@0.5	Loss	Epochs
YOLOx-Plus(None)	0.768	3.931	300
YOLOx	0.834	3.384	300
YOLOx(Weaken)	0.850	3.231	300
YOLOx-Plus	0.855	3.115	300
YOLOx-Plus(Weaken)	0.897	2.867	300

using Vivado software to verify the design rationality of the overall logic circuit of the system, analyze the correctness of the system paths after layout and routing, the usage of logic resources and the system power consumption, and completing the file configuration, and output the bit stream of the relevant projects files and hardware logic circuits. The system hardware logic circuit generated by Vivado is shown in FIGURE 11.

E. EVALUATION INDEX

The main evaluation index in this paper is mean average precision (mAP), which is formulated as Equations (22) ~ (27).

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% \tag{22}$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \tag{23}$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100\% \tag{24}$$

$$AP = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{25}$$

$$mAP = \frac{1}{n} \sum_{i=0}^n AP_i \tag{26}$$

$$FPS = \frac{\text{FigureNumber}}{\text{TotalTime}} \tag{27}$$

where,  $TP$  (True Positive) is the number of positive samples predicted correctly,  $FP$  (False Positive) is the number of negative samples predicted correctly,  $FN$  (False Negative) is the number of positive samples predicted incorrectly, “Precision” is the ratio of all correctly predicted frames to all predicted frames, “Recall” is the ratio of all correctly predicted frames to all true frames,  $F1$  is the harmonic mean of “Precision” and “Recall”,  $AP$  is the percentage of correctly

**TABLE 5. (A) The comparative results of different methods. (B) The comparative results of different methods.**

(A)				
Method	mAP	Loss	MB	Epochs
YOLOx	0.834	3.583	8.951	300
YOLOx-CBAM	0.855	3.129	9.032	300
YOLOx-SimAM	0.872	2.925	8.965	300
YOLOx-CSPHB	0.881	2.896	9.143	300
YOLOx-Plus	0.897	2.867	9.180	300

(B)					
Method	AP (accuracy and pieces)				Epochs
	Short	Open	Burrs	missing	
YOLOx	0.930 (900)	0.920 (900)	0.905 (900)	0.944 (900)	300
YOLOx-CBAM	0.959 (900)	0.943 (900)	0.893 (900)	0.948 (900)	300
YOLOx-SimAM	0.962 (900)	0.952 (900)	0.924 (900)	0.953 (900)	300
YOLOx-CSPHB	0.965 (900)	0.957 (900)	0.931 (900)	0.960 (900)	300
YOLOx-Plus	0.973 (900)	0.969 (900)	0.955 (900)	0.976 (900)	300

recognized samples in the network to the total number of recognized samples, *mAP* is the average of the *AP* of all defective categories, which is often used to evaluate the recognition rate of the overall network, and *mAP* is usually used as the final index to evaluate the performance of the model [33]. *FPS* (Frames Per Second) refers to the number of pictures that the system can process per second, and is often used to measure the detection speed of the system [34].

**VI. EXPERIMENT RESULT**

**A. COMPARISON OF DIFFERENT DATA ENHANCEMENT METHODS**

This paper used a weak data enhancement method for YOLOx-Plus. FIGURE 12 represents the model training effect of different algorithms, by observing the changes in the evaluation indexes.

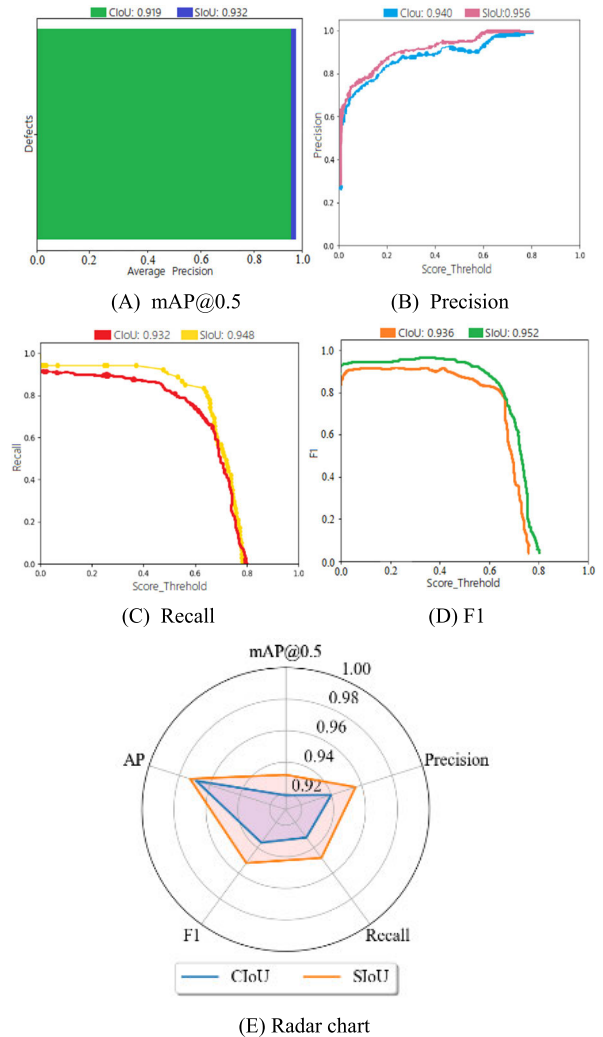
From FIGURE 12, it can be seen that the use of the weakened data enhancement algorithm can effectively improve the detection effect of the algorithm. The dataset of YOLOx-Plus is enhanced by weak data enhancement, and from the comparison experiments, it can be seen that the YOLOx-Plus after weak data enhancement has a better training effect compared with the regular YOLOx and YOLOx-Plus without data enhancement. By observing the changes in the evaluation indexes, it can be seen that the weakened data enhancement algorithm chosen in this paper has a better detection effect.

In order to have a more intuitive understanding of the effectiveness of the performance of the network model enhanced with weaker data, this paper also compares it with other methods in the form of a table, and the comparison results are shown in TABLE 4.

As shown in TABLE 4, when the number of network iterations of the algorithm reaches 300, the detection accuracy

**TABLE 6. The comparison of two different loss functions performance.**

Method	mAP	AP	Recall
CIoU	0.919	0.969	0.932
SIoU	0.932	0.973	0.948



**FIGURE 13. The comparison of different loss functions performance.**

of YOLOx-Plus (Weaken) is improved by about 16.8% compared with that of the YOLOx-Plus (None). Compared with the YOLOx and YOLOx-Plus algorithms using conventional data enhancement methods, the YOLOx and YOLOx-Plus algorithm with weakened data enhancement improves the detection accuracy by about 1.9% and 5.0%, respectively, and the algorithm’s network loss decreases by about 0.153 and 0.248, respectively, which is a significant improvement in the performance metrics. Therefore, the weakened data enhancement method is more effective in improving the performance of the algorithm.

**TABLE 7. The comparative analysis of regularization.**

Method	mAP@0.5 (Train)	mAP@0.5 (Validate)	Loss (Train)	Loss (Validate)
YOLOx	0.934	0.834	3.245	3.583
YOLOx (DropBlock)	0.829	0.831	3.487	3.556
YOLOx-Plus	0.952	0.932	2.571	2.867
YOLOx-Plus (DropBlock)	0.940	0.932	2.641	2.741

**B. COMPARATIVE OF ARCHITECTURE IMPROVEMENTS**

In order to more comprehensively analyze the effectiveness of each module on YOLOx improvement, this paper adds SimAM module and CSPHB module into YOLOx respectively, and the original YOLOx is used as a control group for the ablation experiments, and the experimental results are shown in TABLE 5.

YOLOx-SimAM is added to the backbone network Darknet53 after the dark3, dark4, dark structure without parameter attention SimAM, from the comparative experiments of TABLE 5, The YOLOx-SimAM algorithm achieved a mAP@0.5 of 87.2%, an improvement of 3.8 percentage points. Comparison with the YOLOx-CABM, which adds the classical channel space attention mechanism at the same location, shows that YOLOx-SimAM is superior in terms of both mAP and parameters. YOLOx-SimAM can enhance the network characterization ability while highlighting key features, effectively solving the problem of similar PCB image background and large loss of feature information in network propagation.

**C. COMPARATIVE ANALYSIS OF DIFFERENT LOSS FUNCTIONS**

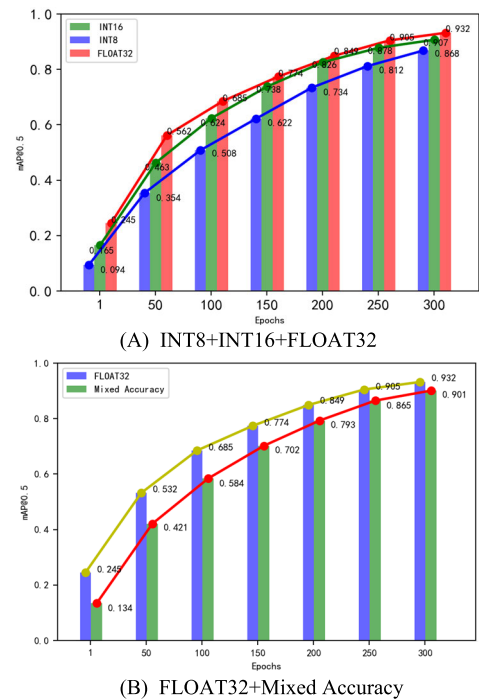
In order to verify the superiority of the SIOU loss function chosen in this paper, one of the most widely used loss functions, CIOU, is added to the YOLOx-Plus network as a control in this paper, and a comparison table of several performance metrics is obtained, as shown in TABLE 6.

As can be seen in TABLE 6 and FIGURE 13, the optimization of the network’s loss function results in a certain improvement in all performance metrics. Since SIOU defines new penalties and includes the vector angles between the required regressions in the computation, the use of the SIOU loss function enables the YOLOx-Plus to perform better than the CIOU loss function in the defect detection task.

**D. COMPARATIVE ANALYSIS OF REGULARIZATION**

In order to investigate the impact of DropBlock regularization method on YOLO neural network, this paper compares and analyzes the training process before and after algorithm improvement. The experimental results are shown in TABLE 7.

From TABLE 7, it can be seen that although the DropBlock regularization method has little impact on the accuracy of the YOLO neural network, it will increase some network



**FIGURE 14. The comparison of two different loss functions performance.**

**TABLE 8. The comparison of various schemes performance.**

Programme	Model Size/Mb	Compression Ratio	Precision Loss	FPS
FLOAT32	42.8	/	/	32.9
INT8	11.6	72%	6.9%	112.7
INT16	23.7	45%	2.7%	54.3
Mixed Accuracy	15.4	64%	3.4%	72.6

loss. From the comparison of data between the training and validation sets, it can be seen that when the YOLO network introduces the DropBlock regularization method, the difference in network loss decreases, indicating that the DropBlock method has a certain effect on overfitting.

**E. QUANTITATIVE COMPARATIVE ANALYSIS OF NETWORKS**

When using a single-precision quantization scheme, since the convolutional layer with the weakest anti-noise performance will limit the quantization accuracy, other convolutional layers with strong anti-noise performance will have a large amount of quantization redundancy, resulting in a more bloated network [35]. According to relevant literature, it can be seen that the noise resistance of different convolutional layers in neural networks is different, and using different precision quantization schemes for different convolutional layers will not significantly affect the output of the network.

Based on the above characteristics, in order to improve the accuracy loss caused by the single-precision quantization scheme and reduce the amount of network data as much as possible, this paper will optimize the single-precision

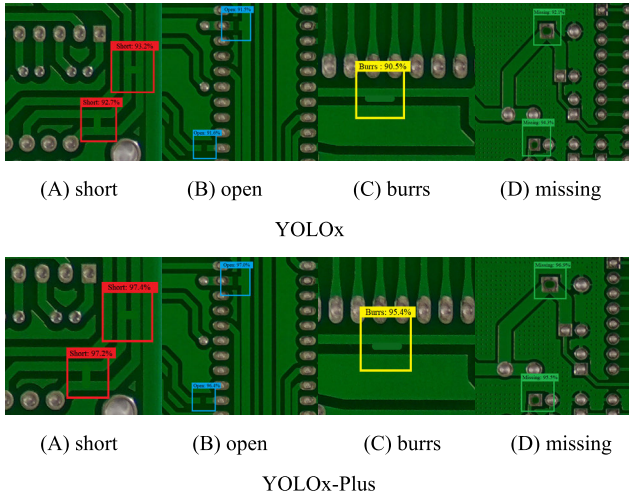


FIGURE 15. The comparison of experimental effects between YOLOx and YOLOx-Plus.

TABLE 9. The comparison of various schemes performance.

Method	mAP	Loss	AP	MB
YOLOx	0.834	3.583	0.930	8.951
YOLOx-CBAM	0.855	3.129	0.959	9.032
YOLOx-SimAM	0.872	2.925	0.962	8.965
YOLOx-CSPHB	0.881	2.836	0.965	9.143
YOLOx-Plus	0.919	2.672	0.969	9.180
YOLOx-Plus-SIoU	0.932	2.547	0.973	9.227

quantization scheme, that is, use a mixed-precision fixed-point quantization scheme to improve the accuracy and network data volume [6]. The specific steps of the mixed-precision fixed-point quantization scheme are: first, set the weight parameters to a quantization scheme with a precision of INT8; then, set the output data of each convolution layer to a quantization scheme with a precision of INT16; finally, the quantization process in section performs network data quantification. When the number of iterations reaches 300, the iterative training is stopped, the preliminary quantification accuracy of each layer of the network is recorded, and the weight and parameter information are saved [36]. After the preliminary quantization accuracy of each layer is determined, the network is retrained with mixed accuracy. First, you need to perform a fixed-point quantization with an accuracy of INT16 and record the mAP value in real time; then, based on the last quantization, requantize with an accuracy of INT8 starting from the first layer of the network, and record the completion of quantization at each layer in turn. The final detection accuracy. When quantizing to a certain layer, if the mAP value of the network is lower than 60% of the INT16 fixed-point quantization, stop quantization and perform retraining until the mAP value loss after retraining

TABLE 10. The comparison of final detection results of various methods.

Types	YOLOx mAP	YOLOx-Plus mAP	YOLOx FPS	YOLOx-Plus FPS
Cortex-A9 ARM	0.834	0.930	42.5	39.2
GTX 1060 GPU	0.834	0.932	65.1	58.3
PYNQ-Z2 FPGA	0.798	0.901	76.9	72.6

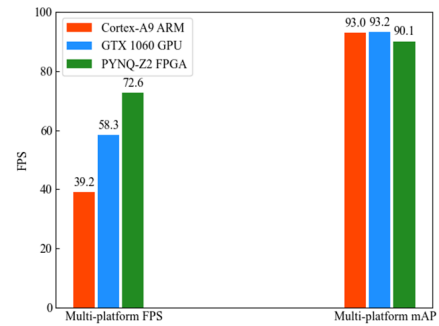


FIGURE 16. The comparison of experimental effects between YOLOx and YOLOx-Plus.

is controlled within 10% of the INT16 quantization scheme, and then end the retraining process. The comparison results of mAP values of the three quantization schemes are shown in FIGURE 14.

As can be seen from FIGURE 14(A), compared with the INT8 single-precision quantization scheme, the mixed-precision quantization scheme improves the detection accuracy of the network by approximately 3.3%. Compared with the INT16 single-precision quantization scheme, the accuracy reduction of the mixed-precision quantization scheme is not large, about 0.7%. As can be seen from FIGURE 14(B), the final detection accuracy of the YOLOx-Plus improved algorithm after mixed-precision quantization is about 90.1%. Compared with the unquantized (FLOAT32) solution, the accuracy of the algorithm is reduced by about 3.1%, which is higher than that of INT8 single-precision quantization. The solution has improved by about 3.5%, and is about 0.7% lower than the INT16 single-precision quantization solution.

As can be seen from TABLE 8, after mixed-precision quantization, the storage capacity occupied by network data is reduced by approximately 8.3Mb compared with the single-precision INT16 solution, and the compression ratio is increased by approximately 19%. Compared with the unquantized solution, the compression ratio is approximately 64%.

F. DETECTION EFFECT ANALYSIS

YOLOx-Plus and the original YOLOx are used for validation, and it can be seen from FIGURE 15 and TABLE 9 that the YOLOx-Plus is better for the detection of PCB defects, which

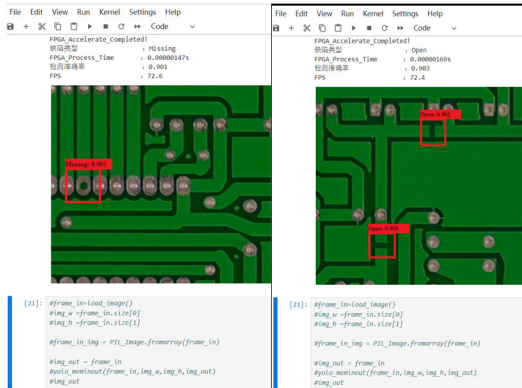


FIGURE 17. The example of FPGA rapid detection results.

TABLE 11. The comparison of final detection results of various methods.

Methods	FPS	mAP@0.5	COCO mAP@0.5
YOLOx-Plus(FPGA)	72.6	0.901	0.785
YOLOx-Plus	58.3	0.932	0.834
YOLOx(FPGA)[34]	76.9	0.901	0.741
YOLOx[16]	65.1	0.932	0.807

proves the superiority of the YOLOx-Plus proposed in this paper.

From FIGURE 15, it can be seen that when YOLOx-Plus and the original YOLOx algorithm are used to detect the same PCB defect targets, the defects detected by the YOLOx-Plus algorithm have higher confidence in both position and type. In addition, as shown in TABLE 9, the final version of YOLOx-Plus algorithm with the introduction of SIOU loss function outperforms other algorithms in terms of various evaluation indicators with little change in parameter quantity. Therefore, the YOLOx-Plus algorithm performs better in detecting various types of defects, verifying that the improvements made to the network in this study are of significant importance.

### G. ACCELERATION EFFECT ANALYSIS

In order to evaluate the acceleration performance of the FPGA-based PCB defect detection accelerator, this paper will also use the YOLOx-Plus defect detection algorithm and the best training model before and after the improvement on the ARM (Cortex-A9) platform, GPU (GTX 1060) platform and FPGA (PYNQ-Z2) platform was deployed and implemented, and the detection performance was compared [37]. The specific comparison plan is: Letting the same algorithm and training model detect 400 images in the PCB verification set, and find the average PCB defect detection accuracy and detection speed under different platforms, and get the results on ARM platform and GPU before and after the algorithm improvement. The average detection accuracy and average detection speed of the platform and FPGA plat-

form. The corresponding comparison results are shown in TABLE 10 and FIGURE 16.

As can be seen from TABLE 10, the FPS value of YOLOx-Plus decreases slightly, but the detection accuracy is significantly improved, with the accuracy increased by about 9.6%. From FIGURE 16, By comparing the detection speed and accuracy of multiple platforms, the results show that the PYNQ-Z2 FPGA platform has the highest average FPS value, with an average accuracy of 90.1% and an accuracy loss of less than 3%. So, YOLOx-Plus deployed on FPGA has best balance.

The FPGA rapid detection of PCB board defects is shown in FIGURE 17.

It can be seen from FIGURE 17 displays the detailed parameters of the detection results: the average FPS value of the two defects is about 72.5, and the average detection accuracy is about 90%. According to TABLE 11, the method proposed in this paper has the best balance of accuracy and performance. The final detection accuracy of the system is 90.1%, FPS is 72.6, and the overall performance is good, superior to other mainstream methods, with strong innovation and application value. The experimental results of this paper are in general agreement with the expected results of the theoretical part of the paper.

## VII. CONCLUSION AND DISCUSSION

### A. CONCLUSION

Aiming at the problems of low PCB defect detection accuracy and slow detection speed, this paper proposes a PCB defect detection method based on FPGA. In terms of improving detection accuracy, this paper improves the YOLOx defect detection algorithm and names it YOLOx-Plus. First, it adopts a weakened data enhancement strategy to reduce the number of inaccurate annotations and unreasonable enhancements caused by Mosaic. This can complete model convergence earlier and improve model accuracy. Secondly, this paper introduces parameter-free attention SimAM into the backbone network and uses the energy function to evaluate its features to obtain more effective data features and improve detection accuracy. Thirdly, this article uses the CSPHB module in the feature fusion network, which enables the original YOLOx network to obtain more semantic information, thereby further improving the resolution of the network and enhancing the interactivity of feature fusion. In addition, a DropBlock regularization module has been added after the backbone network to suppress possible overfitting during the training process. Finally, this paper replaces the CIoU loss function in the network with the SIOU loss function, which further improves the detection accuracy of the network and reduces the network loss, making the final YOLOx-Plus algorithm have better performance in PCB defect detection accuracy [38]. In terms of improving detection speed, this article first quantifies the parameters of the YOLOx-Plus algorithm, integrates two precision quantization schemes, INT8 and INT16, and implements a hybrid quantization based on INT8+INT16.

TABLE 12. Glossary.

Number	Name	Symbol
1	Simple Attention Mechanism	SimAM
2	Spatial Pyramid Pooling	SPP
3	Sigmoid Linear Unit	SiLU
4	A Data Enhancement Method	Mosaic
5	A Data Enhancement Method	Mixup
6	Conv + BN + SiLU	CBS
7	A Structure in CSPDarknet Network	CSPLayer
8	You Only Look Once	YOLO
9	Nonlinear Activation Function	Sigmoid
10	Intersection over Union	IoU
11	Complete Intersection over Union	CIoU
12	Distance Intersection over Union	DIoU
13	Generalized Intersection over Union	GIoU
14	Improved Complete Intersection over Union	ICIoU
15	Shape Intersection over Union	SIoU
16	Batch Normalization	BN
17	A Regularization Technique	DropBlock
18	mean Average Precision	mAP
19	Average Precision	AP
20	Integer Quantization	INT
21	Positive Sample Matching Technique	SimOTA
22	Feature Pyramid Network	FPN
23	Path Aggregation Network	PAN
24	Field Programmable Gate Array	FPGA
25	Stochastic Gradient Descent	SGD
26	Convolutional Block Attention Module	CBAM
27	Programmable Array Logic	PAL
28	Generic Array Logic	GAL
29	Complex Programmable Logic Device	CPLD
30	A Method of Attention Mechanism	CSPHB
31	Xilinx Zynq-7000 FPGA	PYNQ-Z2
32	Direct Memory Access	DMA
33	Xilinx FPGA Integrated Development Environment	Vivado
34	Python Integrated Development Environment	Jupyter Notebooks
35	Advanced eXtensible Interface	AXI
36	FPGA Simulation Tool	Modelsim
37	Frames Per Second	FPS
38	High Level Comprehensive Tools	HLS
39	Deep Learning Framework	Pytorch
40	CUDA Deep Neural Network Library	CUDNN
41	Application Specific Integrated Circuit	ASIC
42	Advanced RISC Machines	ARM
43	Graphics Processing Unit	GPU

Combined with the post-quantization retraining method, the quantized YOLOx-Plus network retraining process was completed, and a network model with a compression ratio of 64% was finally output. The detection accuracy loss was less than 3%, and the quantification effect was ideal. Then, for the PYNQ-Z2 platform, this paper divides the FPGA accelerator into PS and PL parts according to the software and hardware co-design concept, and integrates them to design the FPGA accelerator. The computing speed of YOLOx-Plus is increased by about 68.1%, and the acceleration effect is relatively significantly.

In the end, YOLOx-Plus outperformed YOLOx by 9.8% in terms of mAP@0.5, reaching 93.2%, the accuracy of YOLOx-Plus on FPGA can also reach over 90%, reducing network loss by 1.036, and the average AP of each type of defect increased accordingly. The detection speed reached 72.6 FPS, and the overall performance was good. In future work, the author will conduct in-depth research on real scene detection to further improve the detection performance in real scenes.

## B. DISCUSSION

The theoretical part of the paper over-idealizes the experimental results. In the actual experimental process, it is difficult to achieve, and there is a gap between the experimental results and the theoretical study. However, this paper also has certain limitations. For instance, due to time constraints, there is a dearth of additional algorithm scheme demonstrations pertaining to algorithm enhancement. In the design of FPGA accelerators, a certain degree of precision is compromised due to the excessive pursuit of computational velocity. Although a suitable equilibrium has been achieved in terms of algorithmic accuracy and computational speed, there is scope for further enhancement. In the future, the author will persist in conducting comprehensive research to further refine the algorithm and optimize the FPGA accelerator.

## APPENDIX

See Table 12.

## REFERENCES

- [1] S. Wen, Y. Yuan, and J. Chen, "A vision detection scheme based on deep learning in a waste plastics sorting system," *Appl. Sci.*, vol. 13, no. 7, p. 4634, Apr. 2023.
- [2] D. Alves, V. Farias, I. Chaves, R. Chao, J. P. Madeiro, J. P. Gomes, and J. Machado, "Detecting customer induced damages in motherboards with deep neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Padua, Italy, Jul. 2022, pp. 1–8.
- [3] B. Hu and J. Wang, "Detection of PCB surface defects with improved faster-RCNN and feature pyramid network," *IEEE Access*, vol. 8, pp. 108335–108345, 2020.
- [4] J. P. Nayak, "PCB fault detection using image processing," in *Proc. IOP Conf. Ser., Mat. Sci. Eng.*, May 2017, pp. 205–218.
- [5] A. D. Santoso, "Development of PCB defect detection system using image processing with YOLO CNN method," *Jour. Art. Int. Res.*, vol. 18, no. 6, pp. 121–135, Feb. 2022.
- [6] J. Tang, S. Liu, D. Zhao, L. Tang, W. Zou, and B. Zheng, "PCB-YOLO: An improved detection algorithm of PCB surface defects based on YOLOV5," *Sustainability*, vol. 15, no. 7, p. 5963, Mar. 2023.
- [7] G. Ras, N. Xie, M. Van Gerven, and D. Doran, "Explainable deep learning: A field guide for the uninitiated," *J. Artif. Intell. Res.*, vol. 73, pp. 329–397, Jan. 2022.
- [8] M. Humayun, F. Ashfaq, N. Z. Jhanjhi, and M. K. Alsadun, "Traffic management: Multi-scale vehicle detection in varying weather conditions using YOLOV4 and spatial pyramid pooling network," *Electronics*, vol. 11, no. 17, p. 2748, Sep. 2022.
- [9] D. A. Ranjan, "Comparing YOLOV8 and mask RCNN for object segmentation in complex orchard environments," *Sensors*, vol. 22, no. 13, p. 4720, Mar. 2022.
- [10] P. Daogang, G. Ming, W. Danhao, and H. Jie, "Anomaly identification of critical power plant facilities based on YOLOX-CBAM," in *Proc. Power Syst. Green Energy Conf. (PSGEC)*, Shanghai, China, Aug. 2022, pp. 649–653.
- [11] W. Shi, Z. Lu, W. Wu, and H. Liu, "Single-shot detector with enriched semantics for PCB tiny defect detection," *J. Eng.*, vol. 2020, no. 13, pp. 366–372, Jul. 2020.
- [12] M. Zhang, "Surface defect detection of solar cells based on SimAM YOLOV5," *Elec. Meas. Tech.*, vol. 46, no. 22, pp. 17–25, Jul. 2023.
- [13] C. Zhu, "Hybrid defect detection model based on SimAM module and ResNet34 network," *Mod. Manu. Eng.*, vol. 2, no. 13, pp. 1–9, 2023.
- [14] J. Su, "PCB defect detection algorithm based on YOLO-G," *Micro Comp.*, vol. 12, no. 17, pp. 1–10, Mar. 2024.
- [15] Z. Deng, "Real time detection technology and system design for non-woven fabric defects," *Ima. Vis. Comp.*, vol. 47, no. 21, pp. 583–593, Dec. 2021.
- [16] W. Ding, Z. Huang, Z. Huang, L. Tian, H. Wang, and S. Feng, "Designing efficient accelerator of depthwise separable convolutional neural network on FPGA," *J. Syst. Archit.*, vol. 97, pp. 278–286, Aug. 2019.
- [17] H. Wang, "Improved mosaic: Algorithms for more complex images," in *Proc. Jour. Phy. Conf. Ser.*, Feb. 2020, pp. 120–137.
- [18] G. Ran, X. Lei, D. Li, and Z. Guo, "Research on PCB defect detection using deep convolutional neural network," in *Proc. 5th Int. Conf. Mech., Control Comput. Eng. (ICMCCE)*, Dec. 2020, pp. 1310–1314.
- [19] J. Yuan, "Defect detection method of PCB based on improved YOLOV5," *Int. Jour. Fron. Eng. Tech.*, vol. 4, no. 10, pp. 28–33, Mar. 2022.
- [20] M. Glučina, N. Ančelić, I. Lorencin, and Z. Car, "Detection and classification of printed circuit boards using YOLO algorithm," *Electronics*, vol. 12, no. 3, p. 667, Jan. 2023.
- [21] W. Xuan, G. Jian-She, H. Bo-Jie, W. Zong-Shan, D. Hong-Wei, and W. Jie, "A lightweight modified YOLOX network using coordinate attention mechanism for PCB surface defect detection," *IEEE Sensors J.*, vol. 22, no. 21, pp. 20910–20920, Nov. 2022.
- [22] Y. Chen, Y. Tang, H. Hao, J. Zhou, H. Yuan, Y. Zhang, and Y. Zhao, "AMFF-YOLOX: Towards an attention mechanism and multiple feature fusion based on YOLOX for industrial defect detection," *Electronics*, vol. 12, no. 7, p. 1662, Mar. 2023.
- [23] L. Kang, Y. Ge, H. Huang, and M. Zhao, "Research on PCB defect detection based on SSD," in *Proc. IEEE 4th Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT)*, Dali, China, Oct. 2022, pp. 1315–1319.
- [24] A. Tomoki, "Improvement of PCB defect detection algorithm based on FPGA," *Sensors*, vol. 21, no. 15, pp. 45–57, Feb. 2021.
- [25] X. Zhang, "Design of industrial PCB component detection system based on FPGA," *Iran. Jour. Com. Sci.*, vol. 11, no. 37, pp. 83–93, May 2022.
- [26] P. Rohit, "Automated detection and classification of pavement distresses pavement surface," *Jour. Trans. Res. Board.*, vol. 13, no. 21, pp. 1440–1456, Jul. 2021.
- [27] A. Raimundo, J. P. Pavia, P. Sebastião, and O. Postolache, "YOLOX-ray: An efficient attention-based single-staged object detector tailored for industrial inspections," *Sensors*, vol. 23, no. 10, p. 4681, May 2023.
- [28] Z. Gevorgyan, "SiOU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
- [29] J. Xue, F. Cheng, Y. Li, Y. Song, and T. Mao, "Detection of farmland obstacles based on an improved YOLOV5S algorithm by using CiOU and anchor box scale clustering," *Sensors*, vol. 22, no. 5, p. 1790, Feb. 2022.
- [30] W. Deng and Z. Wang, "SN-YOLO: Improved YOLOV5 with softer-NMS and SiOU for object detection," in *Proc. 5th Int. Conf. Artif. Intell. Pattern Recognit.*, Sep. 2022, pp. 57–62.
- [31] S. Bhardwaj, "Detection of bare PCB defects by image subtraction method using machine vision," in *Proc. Wo. Cong. Eng.*, Feb. 2019, vol. 2, no. 11, pp. 879–892.
- [32] S. Mittal, "A survey of FPGA-based accelerators for convolutional neural networks," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 1109–1139, Feb. 2020.

- [33] A. G. Blaiech, K. Ben Khalifa, C. Valderrama, M. A. C. Fernandes, and M. H. Bedoui, "A survey and taxonomy of FPGA-based deep learning accelerators," *J. Syst. Archit.*, vol. 98, pp. 331–345, Sep. 2019.
- [34] A. Santoso, "Development of PCB defect detection system using image processing with YOLO CNN method," *Jour. Art. Int. Res.*, vol. 6, no. 13, pp. 11–22, Nov. 2023.
- [35] D. Pestana, P. R. Miranda, J. D. Lopes, R. P. Duarte, M. P. Véstias, H. C. Neto, and J. T. De Sousa, "A full featured configurable accelerator for object detection with YOLO," *IEEE Access*, vol. 9, pp. 75864–75877, 2021.
- [36] C. Chen, "An improved-YOLOX model for detection of fabric defects," *Int. Jour. Com. Sci.*, vol. 51, no. 1, pp. 28–36, Feb. 2024.
- [37] Q. He, A. Xu, Z. Ye, W. Zhou, and T. Cai, "Object detection based on lightweight YOLOX for autonomous driving," *Sensors*, vol. 23, no. 17, p. 7596, Sep. 2023.
- [38] Y. Hou, L. Zhang, Y. Wang, X. Zhao, G. Feng, and Y. Zhang, "Field rapid detection method of wind turbine blade fixing bolt defects based on FPGA," *Optoelectronics Lett.*, vol. 18, no. 9, pp. 541–546, Sep. 2022.



**LEI ZHANG** is currently a Graduate Tutor with the School of Artificial Intelligence and Data Science, Hebei University of Technology, with a research direction of artificial intelligence. He has published more than ten SCI and EI articles.



**YAJING PAN** is currently pursuing the bachelor's degree with the School of Mathematics, Hangzhou Normal University, with a research direction of AI.



**YUJIE ZHANG** is currently pursuing the master's degree with the School of Artificial Intelligence and Data Science, Hebei University of Technology, with a research direction of artificial intelligence, mainly engaged in the research of image detection algorithms. He has published a number of academic articles.

...