International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 17 (2025) pp. 1-20

https://doi.org/10.70917/ijcisim-2025-0184
Article

PCB Surface Defect Detection Algorithm Based on Multi-Scale Feature Enhancement and Lightweight

Xiaoyan Xu [1,2] and Jennifer C. Dela Cruz [1,*]

[1] School of Electrical, Electronics and Computer Engineering, Mapúa University, Manila, 1002, Philippines
[2] School of Engineering, Lishui University, Lishui, Zhejiang, 323000, China; jcdelacruz@mapua.edu.ph

Abstract: With the continuous development of electronic technology, printed circuit boards (PCBs) are evolving toward higher density and precision. However, traditional surface defect detection methods face issues such as low accuracy and poor real-time performance, making it difficult to meet modern production requirements. This paper uses YOLOv8 as the base network and addresses the challenges of model deployment in resource-constrained scenarios and slow detection speeds. It proposes a lightweight PCB defect detection method, PDE-YOLO, based on partial convolution. Additionally, considering the challenges of identifying small defects, insufficient accuracy due to insignificant features, and false positives and false negatives in PCB defect detection, we propose another method, YOLOv8-MPSW, which combines multi-scale feature enhancement and attention mechanisms. Based on the proposed PDE-YOLO and YOLOv8-MPSW algorithms, we have developed a simple and efficient PCB surface defect detection system. Experimental results show that the PDE-YOLO algorithm achieves an mAP of 95.8%, with recall rates improved by 3.2% and 4.76% compared to Faster R-CNN and YOLOv8, respectively, reaching 90.4%, demonstrating significant reliability. The YOLOv8-MPSW algorithm achieved an mAP of 97.79%, which is 3.34% higher than the detection accuracy of YOLOv8s, meeting the high-precision requirements of industrial detection applications.

Keywords: PCB surface defects; YOLOv8; multi-scale features; attention mechanism

1. Introduction

Printed circuit boards (PCBs) serve as the primary connection medium for electronic components and are often referred to as the “mother of electronic products.” As the foundation of electronic devices, PCBs provide robust mechanical support for various components within circuits, ensuring the realization of intended circuit functions. They find widespread application across industries such as communications, electronics, and computing [1-2]. Additionally, the development level of the PCB industry directly reflects a nation or region's manufacturing technology capabilities and information technology industry strength [3]. According to relevant statistics, the global PCB industry's total output value reached 73.565 billion U.S. dollars in 2024. China, as the world's largest PCB producer, accounted for 56% of the global market. With the rapid development of artificial intelligence and the Internet of Things, the importance of PCBs in electronic devices has become increasingly evident. The PCB manufacturing industry has become a vital pillar of the electronics and information technology sector, playing an immeasurable role in driving the progress and development of the entire electronics industry [4-6].

With the rapid development of the electronics manufacturing industry and continuous advancements in packaging technology, electronic products are evolving toward miniaturization. This trend has led to a continuous reduction in the surface area of PCBs, thereby imposing higher demands on PCB production processes [7-9]. During PCB production, a series of complex processes must be undergone, including substrate selection, pattern transfer, and soldering. Each production stage requires precise process control; otherwise, various defects may appear on the PCB surface, including but not limited to via defects,

copper contamination, burrs, open circuits, and short circuits [10-12]. Due to factors such as material quality, equipment performance, and production environment, some PCBs inevitably have defects. Once these defective PCBs are put into use, they may cause equipment malfunctions, and in severe cases, even trigger safety accidents [13]. Therefore, PCB surface defect detection is not only an important part of quality control but also an indispensable key step in the PCB production process.

Traditionally, PCB defects were manually inspected by professionals, a method that was extremely costly and inefficient. As a result, electrical testing methods were adopted, primarily relying on electrical performance to identify surface defects on PCBs. However, this approach was difficult to implement due to the high cost of fixtures required for testing, as well as the lengthy programming and debugging times involved [14]. In recent years, Automatic Optical Inspection (AOI) systems have gradually replaced traditional PCB defect detection methods. Although AOI has many advantages in PCB defect detection, Chen et al. [15] pointed out that AOI has a high false positive rate, and manual secondary quality inspection increases costs. In contrast, PCB defect detection using deep learning algorithms achieves an accuracy rate of approximately 95%, with a single image detection time of 0.027 seconds. Additionally, AOI detection has poor robustness, and the investment and maintenance costs of AOI equipment are relatively high, particularly for small-batch PCB manufacturing companies, where it may not be cost-effective [16].

As deep learning technology continues to mature, an increasing number of researchers are beginning to explore its applications in PCB defect detection. Compared to traditional algorithms, deep learning offers several significant advantages. First, deep learning can automatically learn defect features without the need for manual extraction, thereby addressing the issue of human intervention required in traditional algorithms. Second, deep learning can handle high-dimensional data and exhibits excellent generalization performance, enabling it to adapt more effectively to various scenarios [17]. Additionally, by leveraging end-to-end learning through neural network structures, the overall efficiency of PCB surface defect detection can be enhanced [18]. Zhou et al. [19] developed a lightweight object detection model for detecting small defects on PCB surfaces using an adaptive training sample selection strategy, a detail-enhancing convolutional module, a shared lightweight detail-enhancing detection head, and a loss function. The model achieved detection accuracy and precision of 97.8% and 99.6%, respectively, while reducing model size and computational complexity. Lei et al. [20] aggregated multiple convolutional kernels to form a reliable and lightweight adaptive convolutional network, optimized the network through tensor decomposition, and processed dataset noise under collaborative learning and confidence assessment. This method achieved excellent feature representation capability, visual interpretability, and robustness in PCB surface defect detection. Shen et al. [21] utilized object detection technology and character recognition technology based on deep convolutional neural networks to jointly construct a PCB defect detection system, improving overall detection accuracy while maintaining real-time performance. However, the use of multiple algorithms can lead to a significant increase in computational load, which not only consumes a large amount of time and computational resources but may also affect the accuracy of the processing results. Aiswarya [22] developed a deep convolutional neural network based on improved fault features in PCB images for detecting surface defects, achieving an accuracy rate of 95.70% and a precision rate of 96.88%. Zhao et al. [23] proposed a lightweight PCB surface defect detection model under a multi-branch streamlined convolutional structure, a simplified GDLite architecture, and a lightweight coordinate attention module, achieving a detection accuracy of 98%. However, deep learning algorithms also have some issues. As the number of convolutional layers increases, the model's parameter count becomes very large and the model structure becomes complex, leading to longer training and prediction times, which are difficult to meet the real-time industrial detection requirements. Additionally, deep learning models often face issues such as overfitting and underfitting, which require further improvement [24-25].

To address the aforementioned issues, researchers have turned their attention to improving YOLO as a foundational algorithm. The YOLO series is widely applied in the field of PCB surface defect detection due to its high detection accuracy and fast speed [26]. Zhang and Xi [27] improved YOLOv4 and detected PCB surface defects with the assistance of K-means++ clustering, achieving an accuracy rate of up to 99.34% and reducing false negatives and false positives. Additionally, relevant researchers introduced attention mechanisms to optimize defect detection. Hu et al. [28] developed a real-time detection model for PCB surface micro-defects based on residual networks and a novel attention mechanism, improving accuracy and precision. Subsequently, attention mechanisms were utilized or combined to improve the YOLO algorithm for defect detection. Zhao et al. [29] introduced a ShuffleNetV2-YOLOv5 model incorporating attention mechanisms and multi-source information fusion for rapid detection of PCB surface defects, achieving improvements in accuracy and computational efficiency. Wang et al. [30] utilized the C2f_SHSA attention mechanism, C2f_IdentityFormer structure, and PloU to improve the YOLOv8n baseline model, achieving a PCB surface defect detection algorithm

with higher accuracy, efficiency, and real-time performance.

Currently, there are three major development directions for multi-scale feature enhancement technology: feature pyramid optimization, dynamic receptive field mechanisms, and lightweight design, which inject new momentum into PCB surface defect detection. In the development of multi-scale feature enhancement technology, researchers initially relied on manual feature design to achieve multi-scale analysis, such as using the SIFT (Scale-Invariant Feature Transform) algorithm to extract key point features from images. While this method can capture multi-scale information to some extent, it depends on human experience and has high computational complexity [31-33].

Since the beginning of the 21st century, with the rise of deep learning technology, multi-scale feature enhancement methods based on convolutional neural networks have gradually become the mainstream [34-35]. In recent years, researchers have further explored fusion methods combining multi-scale features with attention mechanisms to better highlight important features and suppress noise. These methods have achieved significant results in tasks such as object detection and image segmentation, effectively enhancing the representational capabilities of multi-scale features and driving the development of multi-scale feature enhancement algorithms [36-38]. Wang et al. [39] developed a few-shot PCB surface defect detection model based on feature enhancement and multi-scale fusion, effectively balancing detection accuracy and efficiency. Hou and Zhang [40] established a lightweight PCB defect detection model based on YOLOv8, which integrates an advanced feature fusion module, an enhanced advanced filtering feature fusion pyramid network, an improved detection head, and a multi-scale convolutional block attention module, thereby reducing computational complexity and improving detection speed. Liu and Wen [41] introduced the k-means clustering algorithm, MobileNetV2, spatial pyramid pooling structure, and feature pyramid network to respectively obtain defect anchor boxes, backbone networks, and enhanced image perception ranges from PCB surface images, and perform multi-scale feature fusion to rapidly detect PCB surface defects. This approach reduces model size and offers real-time performance and portability. Zhu et al. [42] combined a receptive field expansion network, multi-scale features, a coordinate information interaction mechanism, Varifocal Loss, Complete Intersection over Union Loss, and a multi-dimensional dynamic convolutional prediction head based on a multi-dimensional attention mechanism to construct a fine-grained defect detection network capable of more accurately detecting minute defects on lightweight PCB surfaces. Chen et al. [43] designed a lightweight CenterNet with enhanced local and global context by utilizing a dual-branch lightweight visual transformer module, coordinate attention, dual-level wiring attention, path aggregation network feature fusion structure, and a lightweight deep convolutional prediction head. Compared to YOLOv8s, this method achieves more precise PCB surface defect detection, simplifies parameters, and offers superior real-time performance and robustness. Lang and Lv [44] designed a PCB defect detection algorithm named SEPDNet, which uses a re-parameterizable convolutional network as the backbone and a neck network optimized by a feature pyramid network. With a single detection head, the detection accuracy is improved, and the model parameters are further simplified compared to YOLOv9u-s. Zhang et al. [45] enhanced PCB defect image features using a lightweight feature extraction network and downsampling methods. They employed a multi-scale aggregation network and its attention module for feature map information sharing and interference removal, and used a lightweight decoupled head for detection, forming a high-precision, high-speed, and computationally balanced PCB defect detection network that performs excellently on edge systems or resource-constrained embedded devices. Yu et al. [46] developed a lightweight PCB defect detection algorithm based on YOLO, utilizing GhostNet (backbone network), deep separable convolutions (neck network optimization), the C3STR module (SwinTransformer + neck network C3 module to address image background and classification interference), and a bidirectional feature pyramid network (multi-scale feature fusion). The model parameters were reduced, and accuracy was improved.

In recent years, the development of the electronics manufacturing industry has entered a fast-track phase, with printed circuit boards (PCBs) playing an increasingly important role in industrial production. As a result, defect detection for PCB products has become an indispensable component of the electronics manufacturing process. To address the challenges of model deployment in resource-constrained environments and the issue of slow detection speeds, this paper proposes a lightweight PCB defect detection method based on partial convolutions, named PDE-YOLO. This method uses YOLOv8n as its base network, combining the partial convolution PConv with the C2f module in YOLOv8n to introduce the C2f_PConv_Block, effectively reducing the network's computational load and parameter count, thereby achieving network lightweighting; Additionally, a lightweight upsampling operator, DySample, is introduced as the network's upsampling method. By utilizing point sampling, it addresses the poor adaptability of traditional upsampling operators, thereby compensating for the accuracy degradation caused by network lightweighting. Finally, EIoU_Loss replaces CIoU_Loss as the network's loss function, further accelerating model convergence and improving overall detection accuracy. When

addressing the challenge of detecting small defects in PCB surface defect detection, this paper proposes a PCB defect detection method based on multi-scale feature enhancement and attention mechanisms, named YOLOv8-MPSW. This method replaces the C2f module with a mixed aggregation network (MANet) to enhance the feature extraction capabilities of the backbone network; by combining hollow convolutions with different sampling rates and the CBAM attention mechanism, a multi-scale pyramid fusion module (MPAF) is constructed to replace the SPPF module, thereby enhancing the expression capability of multi-scale features; in the neck network, the SE attention mechanism is added to enhance the perception of small defects and the model's generalization performance; simultaneously, the efficient WIoUv3 loss function is adopted to further optimize detection performance. Finally, based on the PDE-YOLO and YOLOv8-MPSW algorithms, we combined Python programming, the PyCharm IDE environment, and the graphical design tool Qt Designer in PyQt5 to design a human-machine interface, thereby constructing a PCB surface defect detection system. We then tested the system's defect detection performance and defect classification performance.

2. Lightweight PCB performance defect detection method based on partial convolution

Printed circuit boards (PCBs) are core components of electronic products, and surface defects directly affect product performance. This paper addresses two key challenges in PCB defect detection: lightweight deployment and micro-defect detection. It proposes two improved algorithms based on YOLOv8n.

This chapter addresses the challenges of model deployment and poor detection speed in scenarios with limited computing resources by proposing a lightweight PCB defect detection method based on partial convolution.

2.1. PDE-YOLO

In response to the challenges of model deployment and low detection efficiency caused by limited computing resources in actual industrial scenarios, this chapter proposes a lightweight PCB defect detection method based on partial convolution, named PDE-YOLO.

First, we propose the PConv_Block module, which uses partial convolution (PConv) as its core structure. Then, we replace the Bottleneck structure in the C2f module of YOLOv8n with the PConv_Block module, further proposing the C2f_PConv_Block module, which significantly reduces convolution operations and lowers memory access [47]. In edge computing scenarios, this effectively alleviates memory pressure: simultaneously, all upsampling modules in the network's neck region are replaced with the lightweight upsampling operator DySample, which effectively addresses the limitations of traditional upsampling operators in terms of adaptability using a point sampling method, thereby mitigating the accuracy degradation caused by network lightweighting: Finally, the EIOU loss function is adopted to replace the previous loss function, more accurately measuring the matching degree between bounding boxes and accelerating convergence speed [48].

2.2. Improved C2f module

2.2.1. Partial Convolution

The formula for the number of floating-point operations (FLOPs) for partial convolution PConv is shown in Equation (1):

$$FLOPs = h \times w \times k^2 \times c^2 \tag{1}$$

Among these, h and w represent the width and height of the feature map, k represents the size of the convolution sum, and c represents the number of channels. Typically, the number of channels used is one-quarter of the conventional number of channels. Therefore, the FLOPs of PConv are only one-sixteenth of those of conventional convolution.

The memory access count (MAC) formula for PConv is shown in Equation (2):

$$MAC = h \times w \times 2c + k^2 \times c^2 \approx h \times w \times 2c \tag{2}$$

Among them, h and w are the width and height of the feature map, k is the size of the convolution sum, and c is the number of channels affected by the regular convolution.

PConv's memory access is only 1/4 that of conventional convolution.

2.2.2. C2f module integrating PConv

This paper takes the novel PConv as the main structure and further proposes PConvBlock. Each

PConvBlock module consists of a partial convolution PConv layer and two ordinary convolution layers, which together form an inverted residual structure.

This paper replaces the Bottleneck module in C2f with PConv_Block, proposing C2f_PConv_Block, which makes spatial feature extraction more efficient while reducing redundant calculations and memory accesses, thereby lowering memory access costs.

2.3. Improved upscaling module

2.3.1. Upsampling Module

In convolutional neural networks, transposed convolution is a key operation for implementing upsampling on feature maps. Considering the most basic case, when the stride is set to 1 and no padding is performed, assume that the input feature map size is n×n and the convolution kernel size is k×k. This operation involves the following key steps: First, the convolution kernel slides with a unit stride in the spatial dimension, generating n^2 intermediate feature maps; second, each intermediate feature map is initialized as a k×k zero matrix, and its update rule is to multiply each element of the input feature map with the convolution kernel element-wise and fill the results into the corresponding subregions; finally, the output result is obtained by stacking all intermediate feature maps.

2.3.2. Lightweight Upsampling DySample

YOLOv8 employs the nearest neighbor interpolation method, which achieves upsampling by directly copying the values of adjacent pixels [49]. In contrast, transposed convolution upsampling relies on transposed convolution layers, which expand the feature map size by training the convolution kernel parameters. Although these two traditional upsampling methods are widely used in image processing, they lack the ability to adjust sampling points based on feature map content, which often leads to the loss of some feature information in the image, affecting the accuracy of subsequent recognition. To overcome this limitation, this paper proposes introducing the lightweight upsampling operator Dysample into the YOLOv8n network.

DySample is a dynamic upsampler with universal and fast characteristics. It uses a grid sampling function to perform bilinear interpolation sampling on the input X, thereby generating an upsampled feature map X' of size $C \times H_2 \times W_2$. As shown in Equation (3):

$$X' = gridsample(X, S) \tag{3}$$

By rearranging pixels, offset O is reshaped into the form of $2 \times sH \times sW$. Finally, offset O is added to the original grid sample g to obtain sample set S, thereby completely realizing the sample set generation process, as shown in Equation (4).

$$S = g + O \tag{4}$$

By performing a linear projection on the input features, a point-by-point “dynamic range factor” is generated, with the aim of enhancing the flexibility of the offset. In terms of dynamic range settings, the sigmoid function is employed. The key role of this dynamic range factor lies in its ability to dynamically adjust the offset based on the characteristics of the input image, thereby improving the effectiveness of upsampling. This addresses the limitation of poor adaptability in traditional upsampling operators, thereby enhancing the detection accuracy of PCB images. The calculation formula is shown in Equation (5):

$$0 = 0.5 sigmoid(linear_1(X)) \cdot linear_2(X) \tag{5}$$

2.4. Improved loss function

2.4.1. Common loss functions

The Intersection over Union (IoU) loss function is a metric used to evaluate the degree of alignment between a model's predicted results and the true targets. Although the IoU loss function is widely used in object detection tasks, it has significant limitations. To address this, this paper proposes the Generalized Intersection over Union (GIoU) loss function. This method builds upon the traditional IoU calculation by introducing the minimum bounding rectangle as a penalty term. As shown in Equation (6), the GIOU loss function not only considers the proportion of the overlapping area but also adjusts the loss value by calculating the proportion of the non-overlapping area relative to the bounding rectangle:

$$L_{GIoU} = 1 - \frac{|A \cap B|}{|A \cup B|} + \frac{\left| \frac{C}{A \cup B} \right|}{|C|} \tag{6}$$

Although GIoU effectively addresses some of the limitations of IoU, its penalty mechanism becomes ineffective when the ground truth box and the predicted box exhibit an inclusion relationship, causing GIoU to degrade into a standard IoU form. To address this issue, researchers have proposed the Distance Intersection over Union (DIoU) loss function. This method builds upon the traditional IoU evaluation framework by simultaneously considering the overlapping area between the predicted box and the ground truth box, as well as the spatial distance between their center points. As shown in Formula (7), DIoU can more accurately characterize the spatial differences between predicted and ground-truth bounding boxes:

$$L_{DIoU} = 1 - \frac{|A \cap B|}{|A \cup B|} + \frac{\rho^2(b, b^{gt})}{c^2} \tag{7}$$

Building upon the DIoU method, this paper further proposes the complete intersection over union (CIoU) loss function. The innovation of CIoU lies in its integration of three key geometric features. The CIoU calculation framework consists of three core components: the overlap area term evaluates the coverage of the bounding box, the center distance term measures spatial position deviation based on Euclidean distance, and the aspect ratio term quantifies shape similarity through parameter v. Here, w and h represent the width and height of the predicted box, respectively, while the balancing coefficient α adjusts the weight ratios of each term:

$$L_{CIoU} = 1 - \frac{|A \cap B|}{|A \cup B|} + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{8}$$

$$\alpha = \frac{v}{[1 - IOU(A, B)] + v} \tag{9}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right) \tag{10}$$

2.4.2. EIoU loss function

The original loss function used by YOLOv8 is CIoULoss. Although CIoULoss has achieved significant results in bounding box regression tasks by comprehensively considering geometric features such as overlap area, center point distance, and aspect ratio, there is still room for improvement in its handling of aspect ratio differences. Specifically, CIoULoss mainly focuses on changes in aspect ratio and fails to fully consider the independent effects of width and height on target confidence. To address this issue, this paper adopts EIoULoss as an alternative to the original loss function. Compared to CIoU, the innovation of EIoU lies in treating width error and height error as independent optimization items rather than simply coupling them through the aspect ratio. The EIoU loss function expression is shown in Equation (11):

$$\begin{aligned} L_{EIoU} &= L_{IoU} + L_{dis} + L_{asp} \\ &= 1 - IoU + \frac{\rho^2(b, b^{gt})}{(W^c)^2 + (h^c)^2} \\ &+ \frac{\rho^2(w, w^{gt})}{(W^c)^2} + \frac{\rho^2(h, h^{gt})}{(h^c)^2} \end{aligned} \tag{11}$$

2.5. Experiments and Analysis of Results

2.5.1. Setting up the experimental environment

The hardware and software configuration used in the training environment for this study is shown in Table 1. The experimental platform configuration is as follows: 12th Gen Intel(R) Core(TM) i5-12500H CPU, NVIDIA GeForce RTX 3060, and 32 GB of RAM. The software environment includes Python 3.8, PyTorch 1.13.0, and CUDA 11.7. Further details will not be repeated in the subsequent experiments.

Table 1. Hardware and software environment.

Environmental configuration	Name	Information
Hardware and software environment	CPU	12th Gen Intel(R) Core(TM) i5-12500H
	GPU operating system	NVIDIA GeForce RTX 3060 Windows11
	Python version	3.8
	Pytorch version	1.13.0
	CUDA version	11.7

2.5.2. Dataset and Expansion

The dataset used in this paper is the PCB defect dataset (PCB_DATASET) publicly available from the Beijing University of Technology Intelligent Robotics Open Laboratory. It contains six types of PCB defects: The experiment utilized a PCB defect dataset provided by an open laboratory at a certain university, comprising 705 images covering six defects: missing holes, rodent bites, open circuits, short circuits, burrs, and excess copper. To simulate complex scenarios, the dataset was expanded to 3,576 images using techniques such as random brightness adjustment, noise addition, rotation, cropping, scaling, and mirroring.

2.5.3. Analysis of ablation experiment results

To validate the impact of the improved modules proposed in this paper on PCB surface defect detection in the YOLOv8n model, ablation experiments were conducted under identical experimental parameters. The results are shown in Table 2. In the table, "√" indicates that the method was used to improve the model in the experiment, while "×" indicates that it was not used. The ablation test results indicate that the improved modules proposed in this study (C2f_PConv_Block, DySample, EIOU) significantly optimize the overall performance and computational efficiency of the YOLOv8n model, achieving optimal model performance. When all three innovative modules are fully integrated into the YOLOv8n model, the recall rate R reaches 0.904 (an improvement of 5.73% compared to the unmodified model), mAP@0.5=0.968 (an improvement of 2.7%), and GFLOPs remains stable at 7.1.

Table 2. Ablation experiments.

Experimental serial number	YOLOv8n	DCNv2	MultiSEAM	EIoU	GFLOPs	R(%)	mAP@0.5
1	√	×	×	×	8.2	0.855	0.931
2	√	√	×	×	8.1	0.864	0.934
3	√	×	√	×	7.5	0.873	0.938
4	√	×	×	√	8.2	0.892	0.941
5	√	√	√	√	7.1	0.904	0.958

The detection performance of the model on the test set was quantified using the P-R curve, as shown in Figure 1. The curve of the improved model in this paper shows a balanced characteristic of high accuracy and high recall rate, with an area under the curve (AUC) of 0.958. This proves that the improved model can balance lightweight and detection accuracy.

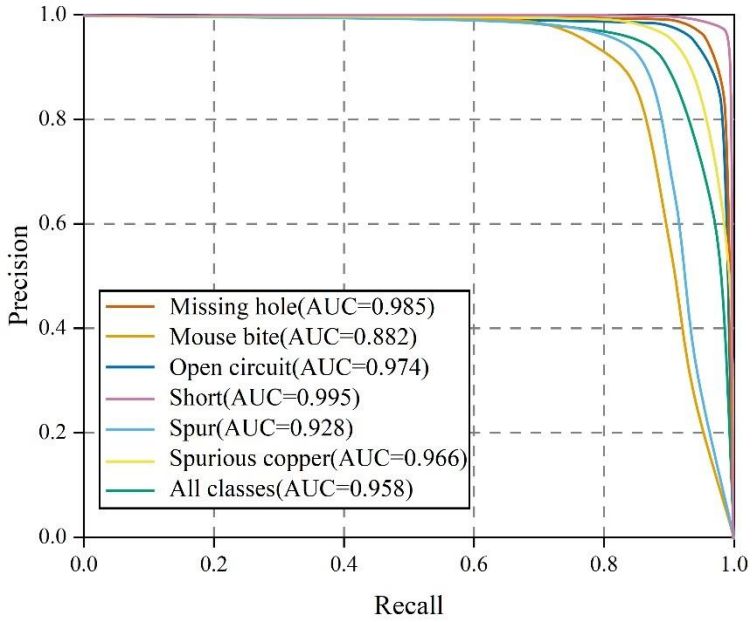


Figure 1. The P-R graph of the improved model.

Multivariate validation of PR curves and ablation experimental data shows that the optimization strategy proposed in this study significantly improves the generalization ability and practicality of the model in PCB defect detection tasks.

2.5.4. Analysis of comparative experimental results

1) Loss Function Comparison Experiment

To validate the superiority of the EIOU loss function, a comparison experiment was conducted between the EIOU loss function and several mainstream loss functions, including IoU, GIoU, DIoU, and CIoU. While keeping other training conditions unchanged, only the loss function was altered in the experiment. The comparison results of different loss functions are shown in Figure 2. As shown in the figure, the EIOU loss function exhibits the fastest convergence rate and stable regression performance, providing more reliable target localization accuracy for PCB defect detection tasks.

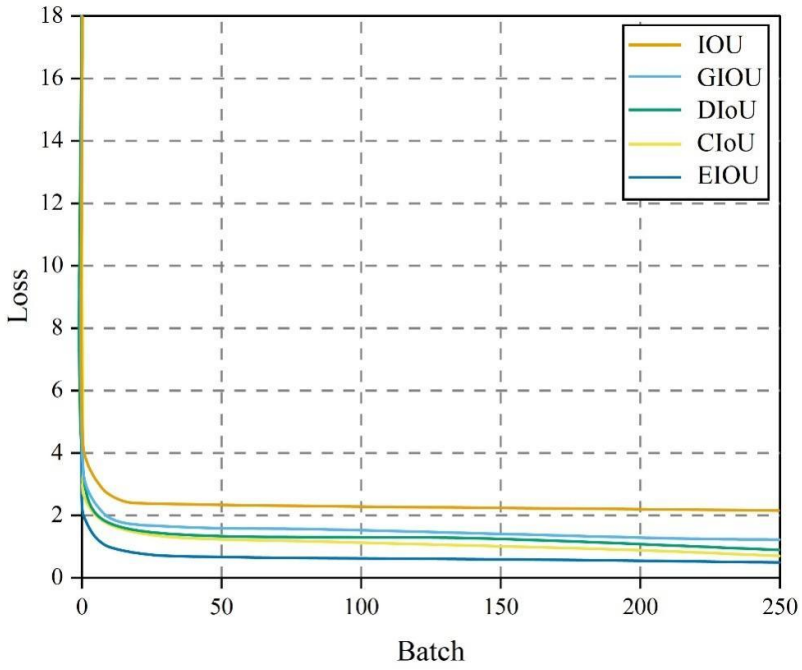


Figure 2. Comparison of different loss function results.

2) Algorithm Comparison Experiment

To validate the performance advantages of the algorithm proposed in this paper for PCB surface defect detection tasks, six mainstream object detection models were selected for comparison experiments. All models were trained and tested on the same training and testing datasets, using uniform parameter settings and hardware configurations to ensure the fairness and reproducibility of the evaluation. The experimental results are summarized in Table 3. In terms of overall performance, Improved algorithm mAP@0.5~95.8%, significantly outperforming other algorithms. In terms of recall rate, the proposed algorithm achieved a high of 90.4%, outperforming Faster R-CNN and YOLOv8 by 3.2% and 4.76%, respectively. The proposed algorithm PDE-YOLO achieves good detection performance while saving computational resources.

Table 3. Comparison test.

Model	mAP@0.5(%)	Recall(%)
Faster R-CNN	93.94	87.2
SSD	87.72	84.44
YOLOX	90.35	85.64
YOLOv5	93.23	86.5
YOLOv7	92.67	86.25
YOLOv8	93.4	86.44
YOLOv10	93.77	86.96
Improved algorithm in this paper	95.8	90.4

3. PCB surface defect detection method based on multi-scale feature enhancement

This preliminary study achieved lightweight improvements to the model by using partial convolution and dynamic upsamplers. However, in actual production, PCB defects are mostly small targets with low contrast between defects and the background, making it difficult for the model to extract features effectively and thus affecting detection accuracy. Therefore, this chapter will focus on enhancing the model's performance in multi-scale object detection, strengthening its ability to extract features from multi-scale objects, and proposing the YOLOv8-MPSW algorithm to improve the detection accuracy of small objects.

3.1. MANet Backbone Network

In the field of PCB defect detection, most defects are very small in size, which poses a significant challenge to detection algorithms when locating defects. Inspired by the Hyper-YOLO architecture proposed by Tsinghua University, this section innovatively integrates the Mixed Aggregation Network (MANet) into the backbone network of YOLOv8. This architecture synergistically fuses three typical convolutional variants: 1×1 bypass convolutions for channel-level feature recalibration, Deep Separable Convolutions (DS Conv) for efficient spatial feature processing, and the C2f module for enhancing feature hierarchy integration. During training, this fusion generates more diverse and abundant gradient flows, significantly enhancing the semantic depth of the base features in each stage of PCB defect feature extraction. The MANet formula is expressed as Equation (12):

$$\begin{cases} X_{mad} = Conv_l(X_m) \\ X_1 = Conv_2(X_{mid}) \\ X_2 = DSConv(Conv_j(X_{mid})) \\ X_3, X_4 = Split(X_{mid}) \\ X_5 = Bottleneck_1(X_4) + X_4 \\ X_6 = Bottleneck_2(X_3) + X_3 \\ \cdots \\ X_{4+n} = Bottleneck_n(X_{3+n}) + X_{3+n} \end{cases} \tag{12}$$

Among these, input processing uses Conv to perform preliminary processing on the input features, generating intermediate feature map X_{mu} . It then uses 1×1 convolutions and depth-separable convolutions to extract features from multiple branches of the intermediate feature map in parallel. Subsequently, multiple Bottleneck modules are used to enhance the branch features, and finally, all

branch features are concatenated and fused and compressed through 1×1 convolutions, generating an output feature map X_{out} with 2c channels. as shown in Formula (13):

$$X_{out} = Conv_o(X_1 \| X_2 \| \cdots \| X_{4+n}) \tag{13}$$

Through the above operations, MANet can effectively extract and fuse feature information from different levels and scales, thereby improving the model's ability to express key features and its level of diversity perception, further enhancing the feature extraction effect and detection performance of the backbone network.

3.2. Multi-scale Pyramid Fusion Module MPAF

In order to further effectively guide the network to focus on key features of PCBs and suppress background interference, thereby improving the network's discrimination ability, the CBAM hybrid attention mechanism is introduced. CBAM combines channel attention and spatial attention mechanisms, aiming to simultaneously consider the spatial location of input data and the relationships between different channels, thereby enhancing the performance and generalization ability of neural networks [50].

The CBAM module enhances key features through a combination of channel attention and spatial attention mechanisms. In the channel attention stage, global average pooling and max pooling operations are first applied to the input feature map along the width and height directions, respectively, yielding two $1 \times 1 \times C$ channel feature maps (where C is the number of channels). This process essentially compresses spatial-dimensional information into a single description for each channel, thereby extracting global channel features.

3.3. Neck network integrating SE attention mechanism

This section proposes the introduction of the SE attention mechanism to enhance the feature processing capabilities and information fusion efficiency of the neck network. The SE attention mechanism is a channel-based attention mechanism whose core idea is to improve the model's performance through two operations: Squeeze and Excitation.

In the object detection network architecture, the neck network plays a central role in connecting the backbone network with the detection head. Its efficiency in processing and fusing features directly impacts the accuracy and efficiency of object detection. Given this, this paper implements optimization measures in the neck network, replacing the original C2f module with the C2fSE module. The C2fSE module leverages the SE attention mechanism to adaptively allocate weights based on the importance of feature channels, effectively reinforcing key channel features and significantly enhancing the network's ability to capture feature information at different scales. This particularly enhances the network's sensitivity and representational capability in perceiving small defects in electrical PCB features. The overall architecture of the improved backbone network and neck network is detailed in Figures 3–7. In these figures, the orange sections clearly indicate the precise locations where the C2fSE module has been added.

3.4. Improvement of the loss function

The design of IoU-based loss functions has become a key element in improving localization accuracy in modern object detection algorithms, and its effectiveness has been verified in multiple benchmark tests. By minimizing the IoU difference between the predicted box and the ground truth box, the detector can gradually optimize its localization capabilities, thereby achieving more accurate object detection results. The IoU definition is expressed as in Equations (14) and (15):

$$IoU = \frac{A \cap B}{A \cup B} \tag{14}$$

$$L_{IoU} = 1 - IoU \tag{15}$$

However, this type of IoU-based loss function has a significant drawback: when the predicted bounding box has no overlapping area with the ground truth bounding box, the IoU value becomes zero, leading to the vanishing gradient problem. To address this issue, we propose the WIoU innovative dynamic focusing mechanism, which reconstructs the bounding box loss function by embedding an attention mechanism.

WIoUv1 combines a distance-based attention mechanism, resulting in a model with two layers of attention mechanisms, defined as in Equations (16) and (17):

$$R_{wloU} = \exp\left(\frac{(x-x_{gt})^2 + (y-y_{gt})^2}{(W_g^2 + H_g^2)^{\cdot}}\right) \tag{16}$$

$$L_{WIoUN1} = R_{WloU} L_{IoU} \tag{17}$$

Here, x and y represent the coordinates of the center point of the detection box, while W_ε and H_ε represent the size of the minimum bounding box. The superscript * indicates that W_ε and H_ε are separated from the computational graph to mitigate the impact of the penalty term on the convergence of gradient descent.

Based on WIoUv1, additional enhancement is achieved by introducing a monotonic focus mechanism using cross-entropy. This requires defining monotonic focus coefficients to emphasize challenging examples and enhance them, resulting in WIoUv2, defined as in Equation (18):

$$L_{w_{loUN2}} = (\frac{L_{lou}^*}{L_{loU}})^\gamma * L_{w_{loUN1}}, \gamma > 0 \tag{18}$$

In WIoUv3, an outlier factor is introduced and defined as in equation (19):

$$\beta = \frac{L_{loU}^*}{L_{loU}} \in [0, +\infty) \tag{19}$$

β indicates the high quality of the predicted bounding box. When the outlier factor is small, a smaller gradient gain is assigned. Conversely, when handling predicted bounding boxes with large outlier factors, a smaller gradient gain is assigned. This strategy mitigates the risk of low-quality samples producing significantly harmful gradients, thereby enabling bounding box regression to prioritize high-quality predictions. WIoUv3 is defined as in Equation (20):

$$L_{WIoUV3} = rR_{WIoU} L_{IoU}, r = \frac{\beta}{\delta \alpha^{\beta-\delta}} \tag{20}$$

In the definition of WIoUv3, α and δ are used as hyperparameters. When β equals δ, r equals 1. If the outlier factor β of the predicted bounding box matches a specific constant value C, the bounding box is awarded the highest gain. This mechanism ensures that WIoUv3 dynamically adjusts its gradient gain allocation strategy based on the current situation.

Therefore, this paper uses Wise-IoUv3 (WIoUv3) instead of the CIoU loss function in YOLOv8 to better meet the complex requirements of PCB defect detection.

3.5. Experiments and Analysis of Results

3.5.1. Data Set Selection

This paper selected the DeepPCB dataset to validate the effectiveness of the proposed YOLOv8-MPSW algorithm in PCB surface defect detection tasks. The DeepPCB dataset is a publicly available PCB defect dataset released by the Institute of Image Processing and Pattern Recognition at Shanghai Jiao Tong University. The dataset includes six categories of PCB surface defect images: holes, mouse bites, open circuits, short circuits, burrs, and stray copper. Each defect image contains multiple defects of different types, with a total of 1,068 images.

3.5.2. Analysis of Experimental Results of Improved Loss Function

This section compares mainstream loss functions such as DIoU, Inner-IoU, and MPDIoU with the WIoUv3 loss function used in this paper. During the experiments, a unified experimental configuration and standardized dataset were used. In the experiments, each loss function was applied to the training of the YOLOv8 model, and performance was recorded at a fixed number of iterations, as shown in Figure 3. The loss function curves reflect the trends observed during the training process. The experimental results indicate that WIOUv3 demonstrates outstanding performance in terms of model convergence. Compared to other loss functions, the YOLOv8 model using WIOUv3 exhibits less overfitting and higher recognition accuracy. Therefore, this study selects the WIOUv3 loss function to replace the original YOLOv8 model loss function, aiming to achieve better performance in PCB surface defect detection.

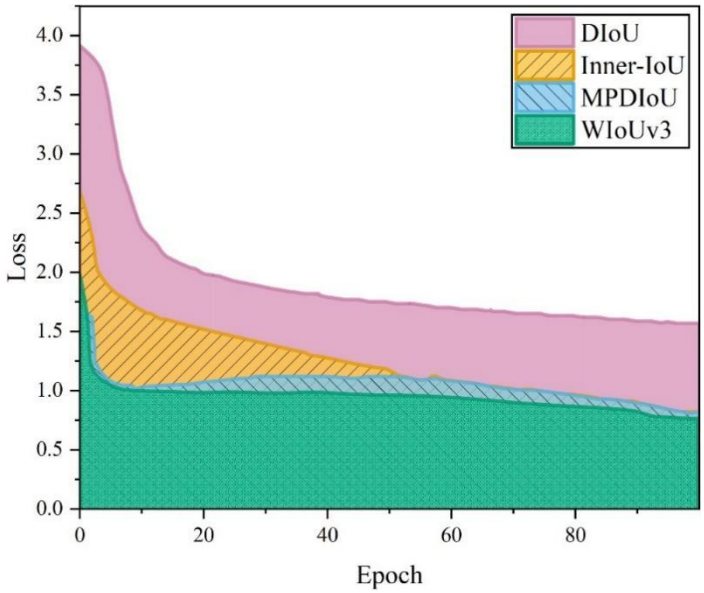


Figure 3. Comparison diagram of bounding box loss.

3.5.3. Analysis of comparative experimental results

On the DeepPCB dataset, the YOLOv8-MPSW algorithm proposed in this paper was compared with Fast-RCNN, SSD, YOLOX, YOLOv3, YOLOv7, PCB-YOLO, YOLO-MBBi, YOLOv5s, and YOLOv8s under the same experimental conditions. The Average Precision (AP) value was used as the performance evaluation metric, with higher AP values indicating better detection performance. The experimental results are shown in Figure 4. As shown in the figure, for the two types of defects—hole defects and copper contamination—that are morphologically similar and easily confused, the YOLOv8-MPSW algorithm achieved high-precision detection rates of 99.41% and 98.31%, respectively, demonstrating its excellent feature discrimination capability. For other types of defects, such as rodent bites (98.54%), open circuits (98.95%), short circuits (98.32%), and burrs (97.79%), the algorithm also achieved high detection accuracy, demonstrating its excellent versatility.

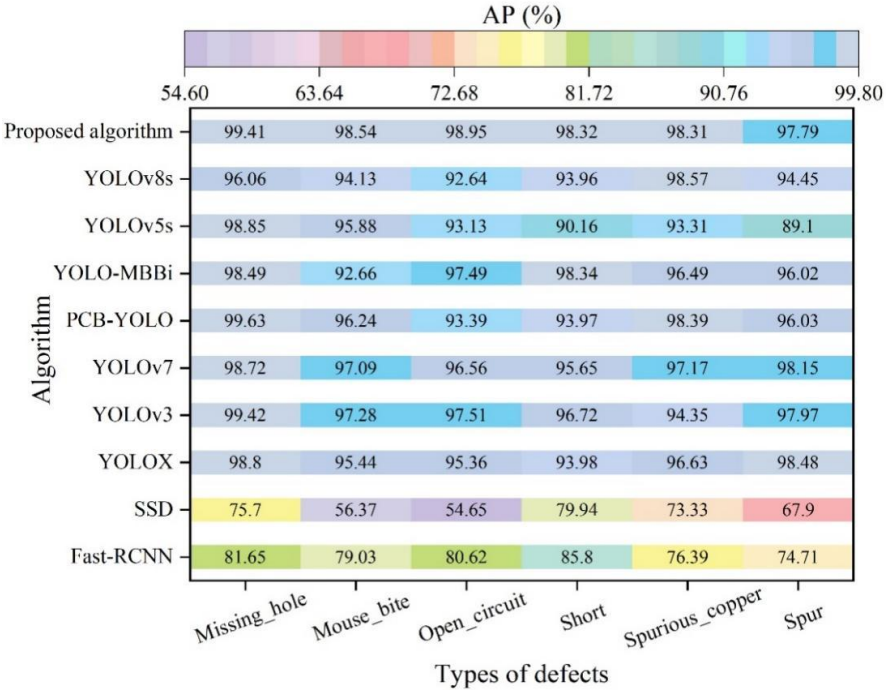
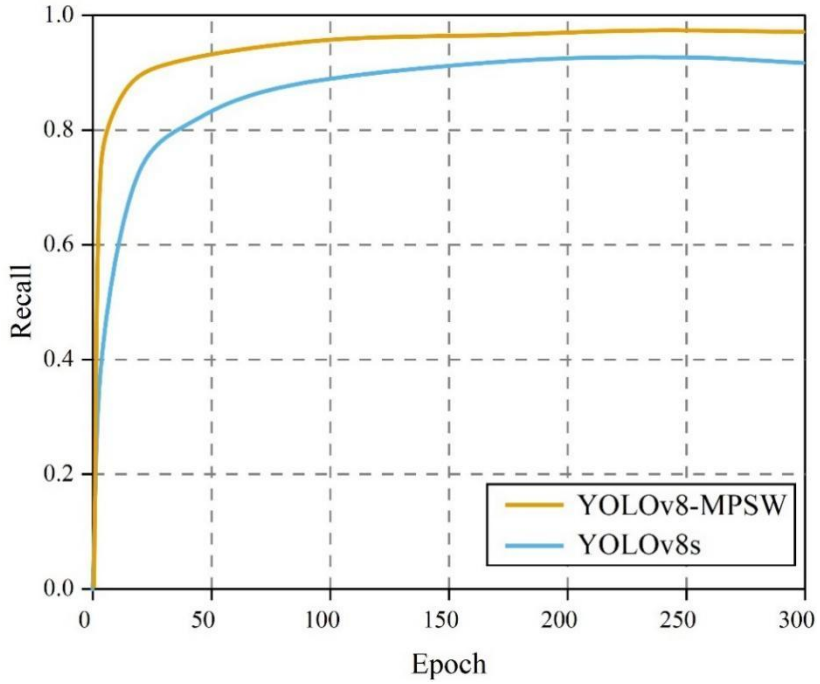
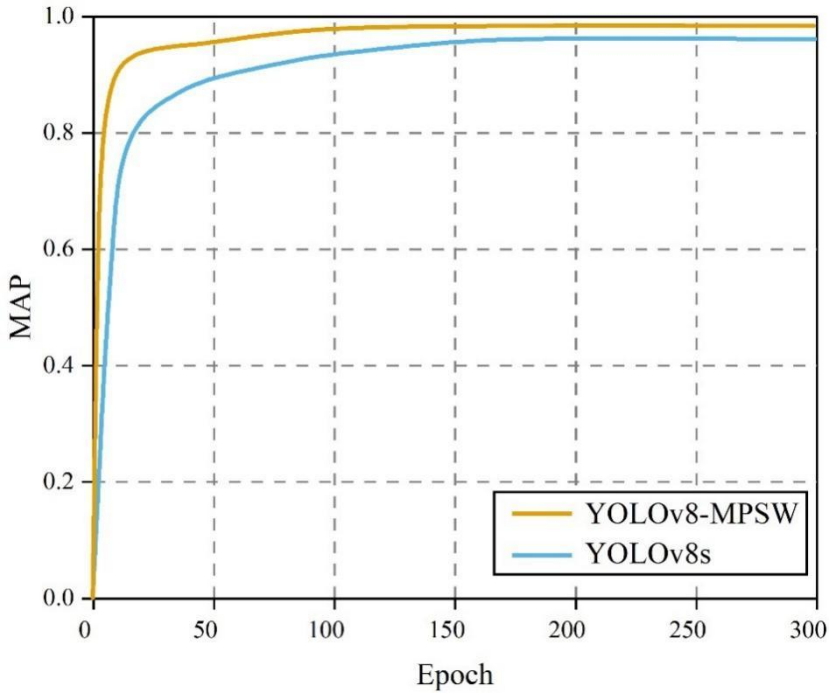


Figure 4. Comparison results on DeepPCB dataset.

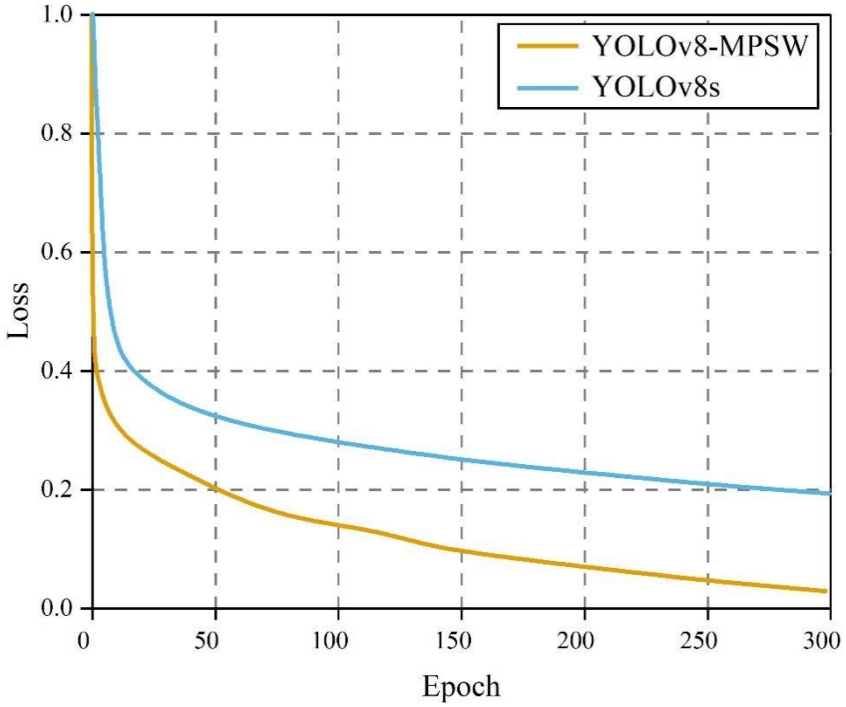
We compared the improved YOLOv8-MPSW algorithm with the baseline YOLOv8s algorithm to evaluate their performance in PCB surface defect detection tasks. The experimental results are shown in Figure 5. Figures (a) to (c) correspond to recall, mAP, and the loss function, respectively. Specifically, the YOLOv8-MPSW algorithm achieves an mAP of 97.79%, which is 3.34% higher than the detection accuracy of YOLOv8s. Additionally, YOLOv8-MPSW also shows improvements in F1 score, average recall rate for different defect types, and average precision rate for different defect types, which validates the effectiveness of the YOLOv8-MPSW optimization strategy.



(a) Recall



(b) mAP@0.5



(c) Loss function

Figure 5. Comparison of performance indicators before and after improvement.

To further evaluate the applicability of the YOLOv8-MPSW algorithm in PCB surface defect detection tasks, the DeepPCB dataset was selected for experimentation. Since the initial dataset contained only 1,068 images, it was expanded to 4,000 images. YOLOv3, YOLO53, YOLOv7, and YOLOv8 were selected as comparison algorithms. The experimental comparison results are detailed in Table 4. As can be seen, the YOLOv8-MPSW algorithm achieves an accuracy of 97.67% on the DeepPCB dataset, representing a significant improvement over some mainstream algorithms.

Table 4. Comparison results on DeepPCB dataset.

Algorithm	Missing_hole	Mouse_bite	Open_circuit	Short	Spurious_copper	Spur	MAP (%)
YOLOv3	97.76	92.17	93.68	91.43	94.5	88.75	93.05
YOLOv7	95.85	96.25	95.27	95.67	86.17	92.3	93.59
YOLOv5s	94.72	94.96	93.65	95.33	93.69	96.39	94.79
YOLOv8s	97.47	94.9	96.11	94.08	94.04	94.14	95.12
Proposed algorithm	98.67	97.87	96.32	96.48	98.85	97.84	97.67

The YOLOv8-MPSW algorithm proposed in this paper not only demonstrates excellent detection performance in PCB surface defect detection tasks but also maintains precise localization and high accuracy. This indicates that the proposed algorithm can ensure excellent detection results when applied to PCB industrial defect detection tasks.

4. PCB Surface Defect Inspection System

In practical applications, to meet the inspection requirements of the PCB industry's production process, this chapter will build a simple and efficient PCB surface defect detection system based on the PDE-YOLO and YOLOv8-MPSW algorithms proposed in this paper.

4.1. Overall System Architecture

The overall framework of the system is shown in Figure 6. It mainly consists of three parts: tools, functions, and users. The tools used include Python, PyCharm IDE, Qt Designer, and PyUIC. In terms of system functions, it is divided into target detection and information detection modules, model loading and initialization modules, and image, video, and camera loading modules.

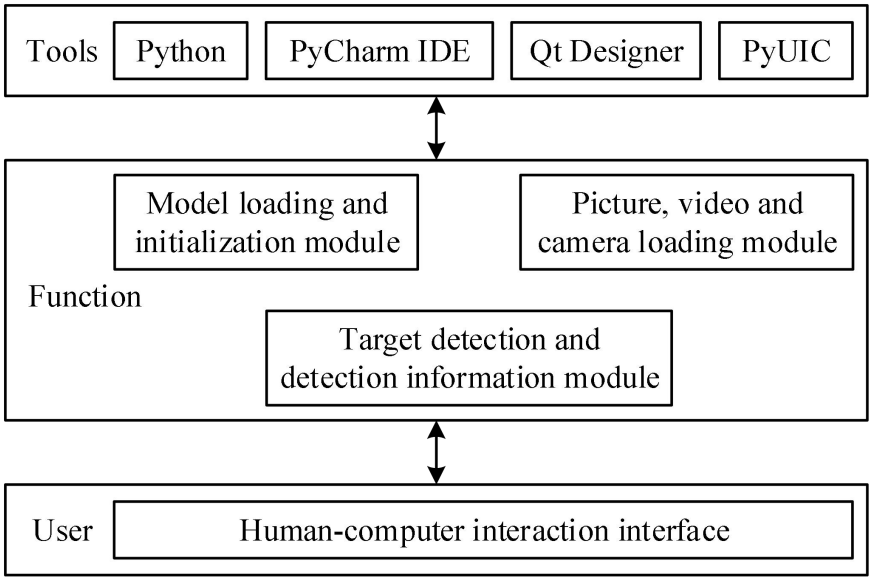


Figure 6. Overall system structure.

4.2. Interface Design and Function Implementation

4.2.1. Human-computer interaction interface design based on Qt Designer

PyQt5 is a Python implementation based on the Qt framework, making it one of the most powerful GUI libraries available. When designing human-computer interaction interfaces, developers utilize the visual graphical interface editor Qt Designer, which is part of PyQt5. Using Qt Designer, you can quickly create graphical interface files (*.ui), which can then be converted into *.py files using PyUIC, a tool configured in the PyCharm environment. These *.py files can be integrated into the source code, fulfilling the “what you see is what you get” (WYSIWYG) requirement, thereby making interface layout design extremely straightforward.

4.2.2. System Function Implementation

The Python program is the core component of the PCB defect detection system, with its functional architecture illustrated in Figure 7. The functional modules of the entire system can be summarized as follows: model loading and initialization module, image/video and camera loading module, and target detection and detection information module. The model loading and initialization module is primarily used to select different weight files and initialize them, primarily utilizing the initialization method from the original YOLOv5, with key parameters configurable.

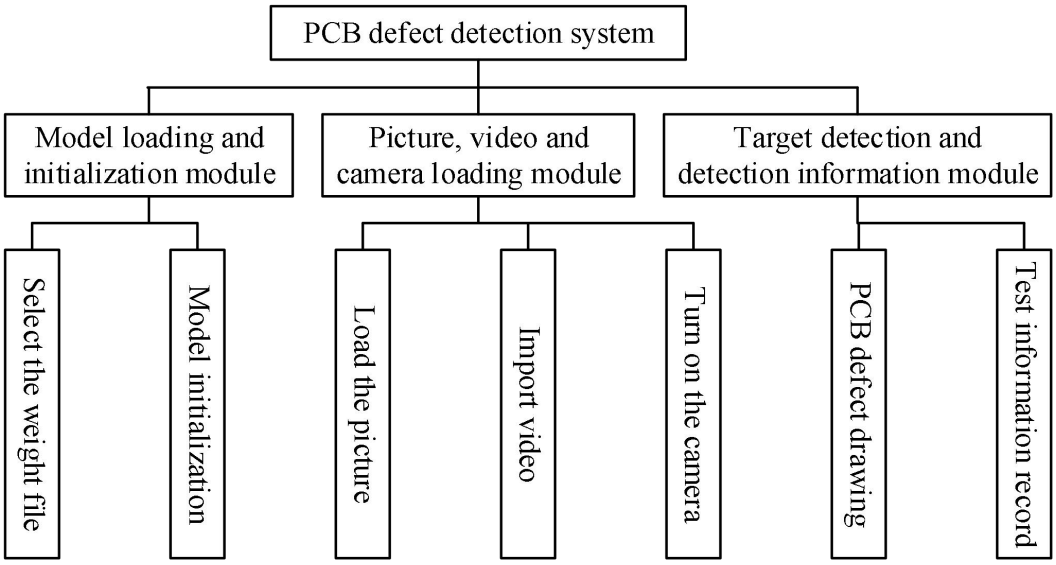


Figure 7. Functional module architecture.

The operation area is divided into three main parts: the function selection area, the detection result display area, and the detection information display area. The function selection area includes selecting weights, initializing the model, image detection, video detection, camera detection, pause/resume detection, and end detection. The detection results for images are exported and saved to the corresponding folder, while video and camera detection are real-time detections and only display detection information.

The functional buttons and their corresponding event links utilize the relevant interfaces provided when the model was established, so it is only necessary to reuse the detection function, model reading function, and separate the initialization function. The display of detection information in the interface is achieved by reading the detection information into a text box using the interface provided by the original YOLO model developer, and then further encapsulating it. The method for processing the camera video stream utilizes OpenCV's cv2.VideoCapture(), with its parameters set to 0, to achieve the functionality of reading the local camera.

1) Function Selection Area: Primarily responsible for selecting and initializing weight files, loading images, importing videos, opening the camera, pausing or directly ending the detection process during detection, etc.

2) Detection Result Display Area: Loaded images, imported videos, camera-detected scenes, and defects identified within them are displayed synchronously in this area in real time. When users perform operations, corresponding operation responses are also displayed in this area.

3) Detection Information Display Area: The position coordinates, defect categories, and confidence levels of detected defects are displayed in this area.

4.3. System Function Testing

4.3.1. Defect detection performance testing

Three random sets of PCB images containing defects were selected from the DeepPCB dataset, named G1 to G3, each containing 100 images with a resolution of 2240×2016 pixels, for performance testing of defect detection. The detection performance was evaluated based on the number of defects in the images, testing for missed detections and false positives, while also measuring the detection time. The relevant test results are shown in Table 5. Based on the defect detection test results, the false negative rate is low, meeting the application requirements. Additionally, the false positive rate remains below 5%, meeting the standards. In terms of detection time, the process takes approximately 21 seconds, demonstrating fast processing speed.

Table 5. Defect detection performance test.

Groups	Number of defects	Number of missed detections	Missing detection ratio	Number of false detections	Misdetection ratio	Detection time (s)
G1	188	6	3.19%	4	2.13%	21.62
G2	214	4	1.87%	2	0.93%	21.55
G3	126	5	3.97%	3	2.38%	20.47

4.3.2. Defect Classification Performance Testing

Three PCB boards of different sizes and component types were prepared for testing. The specific testing results are shown in Table 6. As the size of the PCB board increases, the corresponding accuracy decreases, and the average processing time increases. This may be related to the quality of the images captured. However, as shown in the table, in the surface defect detection work of the three different types of PCB boards (X, Y, and Z), the detection accuracy of the system in this paper is all greater than 98%, and the average processing time is controlled within 10 seconds, which can meet the requirements for PCB surface defect detection in everyday applications.

Table 6. Statistics of image Mosaic detection results

Model	Total number of circuit boards	Template pixel size (pixel)	Pixel size (pixel)	Accuracy (%)	Average time (ms)
X	20	450×720	410×685	98.43	6.55
Y	15	550×860	515×780	98.37	7.29
W	10	800×1400	842×1175	98.75	8.64

Further testing was conducted on the system's defect classification performance. Six hundred images were randomly selected from the DeepPCB dataset, and predictions were made for six types of PCB defects. The relevant test results are shown in Table 7. It can be seen that the average test accuracy of the system for the six common types of PCB defects, such as holes, notches, and open circuits, reached 98.33%, which is a good defect classification result. This can effectively help predict defect types and assist in defect analysis.

Table 7. Defect classification performance test

Defect type	Number of images	Accurate recognition number	Number of misidentifications	Recognition accuracy (%)
Hole	100	97	3	97.00%
Gap	100	100	0	100.00%
Open road	100	100	0	100.00%
Short circuit	100	99	1	99.00%
Burr	100	98	2	98.00%
Residual copper	100	96	4	96.00%
Total	600	590	10	98.33%

Overall, the PCB surface defect detection system proposed in this paper has excellent defect detection performance and can help to better complete the task of detecting defects on the surface of PCBs during the production process.

5. Conclusion

To address the issues of lightweight deployment and detection of minute defects in PCB surface defect detection, this paper proposes two improved algorithms based on YOLOv8: the lightweight PCB surface defect detection method PDE-YOLO based on partial convolution, and the PCB surface defect detection method YOLOv8-MPSW based on multi-scale feature enhancement. Based on the PDE-YOLO and YOLOv8-MPSW algorithms, a PCB surface defect detection system is constructed.

Experimental testing was conducted to evaluate the performance of the PDE-YOLO algorithm for PCB surface defect detection. Ablation experiment results showed that when C2f_PConv_Block, DySample, and EIOU were all incorporated into the YOLOv8n model, the recall rate and mAP reached 90.4% and 96.8%, respectively, with GFLOPs remaining stable at 7.1. When comparing several mainstream loss functions such as IoU, GIoU, DIoU, and CIoU, the EIOU loss function used by PDE-YOLO has the fastest convergence speed and stable regression loss. In terms of overall performance, PDE-YOLO Algorithm mAP@0.5, recall rate reached 95.8% and 90.4%, achieving an effective improvement in detection performance.

The performance of YOLOv8-MPSW in PCB surface defect detection tasks was verified and analyzed. Compared to mainstream loss functions such as DIoU, Inner-IoU, and MPDIoU, the WIoUv3 loss function used by the YOLOv8-MPSW algorithm demonstrated outstanding model convergence, exhibiting less overfitting and higher recognition accuracy. Across various PCB surface defect types such as holes, mouse bites, open circuits, short circuits, burrs, and stray copper, YOLOv8-MPSW consistently achieves detection accuracy exceeding 97%. Compared to the baseline YOLOv8s algorithm, the mAP of the YOLOv8-MPSW algorithm in this paper reached 97.79%, which is 3.34% higher than the baseline, and there were also improvements in F1 scores and average recall and precision rates for different defect types.

Finally, the functionality of the PCB surface defect detection system constructed in this paper was evaluated. In terms of defect detection performance, the system has a low false negative rate, while the false positive rate remains consistently below 5%, meeting the standards. The detection time is stable at around 21 seconds, demonstrating fast processing speed. In terms of defect classification performance, when conducting surface defect detection on three completely different types of PCB boards (X, Y, and Z), the detection accuracy rate of the system exceeds 98% for all cases, with an average processing time controlled within 10 seconds, showcasing excellent performance. For six common PCB defect types such as holes, notches, and open circuits, the average test accuracy rate reaches 98.33%. Clearly, the PCB surface defect detection system constructed in this paper can play an effective role in PCB surface defect detection work.

About the Author

Xiaoyan Xu. She received the bachelor's degree in electrical engineering from Nanjing Agricultural University in Jiangsu, China in 2007, and her master's degree in engineering in 2010. She is currently studying for a doctorate in electronic engineering at Mapúa University. She worked as a laboratory master of Lishui University in Zhejiang Province, China in 2016. She has presided over and participated in 5 provincial and municipal projects and project projects, published more than ten journal papers, and actively participated in international academic exchanges. Her research interests are in the areas of sensor identification and detection, intelligent control, and her current doctoral research direction is the application of deep learning in the field of computer vision. Jennifer C. Dela Cruz. She holds a Doctor of Philosophy in Electronics and Communications Engineering from De La Salle University, Manila, Philippines. She is a Senior IEEE member and currently the Region 10 Section and Chapter Committee Chair, IEEE MGA SAC Awards Subcommittee Chair, and Chairman of the IEEE Education Society of Philippines Section. She is also a local University program evaluator for Electronics Engineering. She has published more than 200++ conference papers and journals. She has been a consistent project leader and member of various local and internationally funded research projects. This accomplishment has led to awarding her the highest number of Scopus-indexed papers from 2015-2024 by Mapua University and a Thought Leader in 2024. Her research interests are in the areas of Biomedical Engineering, Machine Learning, Artificial Intelligence, Wireless Technology, Radio Frequency Techniques, and Antenna Systems.

References

1. Lu, T., Markvicka, E. J., Jin, Y., & Majidi, C. (2017). Soft-matter printed circuit board with UV laser micropatterning. ACS applied materials & interfaces, 9(26), 22055-22062.
2. Kaya, M. (2019). Printed circuit boards (PCBs). In Electronic Waste and Printed Circuit Board Recycling Technologies (pp. 33-57). Cham: Springer International Publishing.
3. Chen, P. S., Chen, J. C. M., Huang, W. T., & Kuo, L. Y. (2020). The development of a modified design chain operations reference model in new product development of the printed circuit board: a case study. Applied Sciences, 10(11), 3703.

4. Holgersson, S., Steenari, B. M., Björkman, M., & Cullbrand, K. (2018). Analysis of the metal content of small-size Waste Electric and Electronic Equipment (WEEE) printed circuit boards—part 1: Internet routers, mobile phones and smartphones. Resources, conservation and recycling, 133, 300-308.
5. Perdigones, F., & Quero, J. M. (2022). Printed circuit boards: the layers' functions for electronic and biomedical engineering. Micromachines, 13(3), 460.
6. Shi, Y., Xin, Z., Loh, P. C., & Blaabjerg, F. (2020). A review of traditional helical to recent miniaturized printed circuit board Rogowski coils for power-electronic applications. IEEE Transactions on Power Electronics, 35(11), 12207-12222.
7. Swettlen, T., Boggs, D., Landeros, J., Amir, D., & Mokler, S. (2020). Device Miniaturization–the Impact of a High Density Soc Direct Chip Attach on Surface Mount and Pcb Technologies. Intel Corp.: Hillsboro, OR, USA.
8. Mirvakili, S. M., Broderick, K., & Langer, R. S. (2019). A new approach for microfabrication of printed circuit boards with ultrafine traces. ACS Applied Materials & Interfaces, 11(38), 35376-35381.
9. Dong, Y., Bao, C., & Kim, W. S. (2018). Sustainable additive manufacturing of printed circuit boards. Joule, 2(4), 579-582.
10. Yevsieiev, V., Maksymova, S., & Alkhalaileh, A. (2025). A METHOD DEVELOPMENT FOR MODELING THE TECHNOLOGICAL PROCESS OF PRINTED CIRCUIT BOARD PRODUCTION BASED ON THE Q-SCHEME. Multidisciplinary Journal of Science and Technology, 5(4), 9-21.
11. Sankar, V. U., Lakshmi, G., & Sankar, Y. S. (2022). A review of various defects in PCB. Journal of Electronic Testing, 38(5), 481-491.
12. Kuttiyil Thomas, O. T., & Gopalan, P. P. (2022). PCB defects. In Electronics Production Defects and Analysis (pp. 39-73). Singapore: Springer Singapore.
13. Koblah, D. S., Dizon-Paradis, O. P., Schubeck, J., Botero, U. J., Woodard, D. L., & Forte, D. (2023). A Comprehensive Taxonomy of Visual Printed Circuit Board Defects. Journal of Hardware and Systems Security, 7(2), 25-43.
14. Singh, K., Kharche, S., Chauhan, A., & Salvi, P. (2024). PCB Defect Detection Methods: A Review of Existing Methods and Potential Enhancements. Journal of Engineering Science & Technology Review, 17(1).
15. Chen, I. C., Hwang, R. C., & Huang, H. C. (2023). PCB defect detection based on deep learning algorithm. Processes, 11(3), 775.
16. Abd Al Rahman, M., & Mousavi, A. (2020). A review and analysis of automatic optical inspection and quality monitoring methods in electronics industry. Ieee Access, 8, 183192-183271.
17. Mavaie, P., Holder, L., & Skinner, M. K. (2023). Hybrid deep learning approach to improve classification of low-volume high-dimensional data. BMC bioinformatics, 24(1), 419.
18. Jiang, F., Tao, W., Liu, S., Ren, J., Guo, X., & Zhao, D. (2017). An end-to-end compression framework based on convolutional neural networks. IEEE Transactions on Circuits and Systems for Video Technology, 28(10), 3007-3018.
19. Zhou, Z., Lu, Y., & Lv, L. (2025). Real-time surface defect detection algorithm for printed circuit boards based on improved YOLOv11n. The Journal of Supercomputing, 81(10), 1148.
20. Lei, L., Li, H. X., & Yang, H. D. (2024). Reliable and lightweight adaptive convolution network for PCB surface defect detection. IEEE Transactions on Instrumentation and Measurement, 73, 1-8.
21. Shen, J., Liu, N., & Sun, H. (2020). Defect detection of printed circuit board based on lightweight deep convolution network. IET Image Processing, 14(15), 3932-3940.
22. Aiswarya, R. S. (2024). Surface Defect Detection in Printed Circuit Boards Using Deep Convolutional Neural Networks. Int. J. of Multidisciplinary and Current research, 12.
23. Zhao, X., Zhang, H., Song, C., Li, H., & Guo, H. (2025). Lightweight intelligent detection algorithm for surface defects in printed circuit board. Computers & Industrial Engineering, 203, 111030.
24. Rangel, G., Cuevas-Tello, J. C., Nunez-Varela, J., Puente, C., & Silva-Trujillo, A. G. (2024). A survey on convolutional neural networks and their performance limitations in image recognition tasks. Journal of sensors, 2024(1), 2797320.
25. Tulbure, A. A., Tulbure, A. A., & Dulf, E. H. (2022). A review on modern defect detection models using DCNNs–Deep convolutional neural networks. Journal of Advanced Research, 35, 33-48.
26. Ge, H., Shi, Y., Zhang, M., Wei, Y., Zhang, H., & Cao, X. (2024, October). YOLO: An Improved High-Accuracy Method for PCB Defect Detection. In 2024 IEEE 12th International Conference on Computer Science and Network Technology (ICCSNT) (pp. 159-165). IEEE.
27. Zhang, M., & Xi, H. (2023). Detection Algorithm of Surface Defect Word on Printed Circuit Board. Computer Systems Science & Engineering, 46(3).
28. Hu, X., Kong, D., Liu, X., Zhang, J., & Zhang, D. (2024). Printed Circuit Board (PCB) Surface Micro Defect Detection Model Based on Residual Network with Novel Attention Mechanism. Computers, Materials & Continua, 78(1).
29. Zhao, Q., Ji, T., Liang, S., & Yu, W. (2024). PCB surface defect fast detection method based on attention and multi-source fusion. Multimedia Tools and Applications, 83(2), 5451-5472.
30. Wang, J., Xie, X., Liu, G., & Wu, L. (2025). A Lightweight PCB Defect Detection Algorithm Based on Improved YOLOv8-PCB. Symmetry, 17(2), 309.
31. Zhou, D., Zeng, L., Liang, J., & Zhang, K. (2017). Improved method for SAR image registration based on scale invariant feature transform. IET Radar, Sonar & Navigation, 11(4), 579-585.

32. Wang, Z., Liu, Y., Zhang, J., Fan, C., & Zhang, H. (2022). Interference image registration combined by enhanced scale-invariant feature transform characteristics and correlation coefficient. Journal of Applied Remote Sensing, 16(2), 026508-026508.
33. Bruno, A., Ardizzone, E., Vitabile, S., & Midiri, M. (2020). A novel solution based on scale invariant feature transform descriptors and deep learning for the detection of suspicious regions in mammogram images. Journal of Medical Signals & Sensors, 10(3), 158-173.
34. Huang, L., Chen, C., Yun, J., Sun, Y., Tian, J., Hao, Z., ... & Ma, H. (2022). Multi-scale feature fusion convolutional neural network for indoor small target detection. Frontiers in Neurorobotics, 16, 881021.
35. Liu, X., Yang, L., Chen, J., Yu, S., & Li, K. (2022). Region-to-boundary deep learning model with multi-scale feature fusion for medical image segmentation. Biomedical Signal Processing and Control, 71, 103165.
36. Ju, M., Luo, J., Wang, Z., & Luo, H. (2021). Adaptive feature fusion with attention mechanism for multi-scale target detection. Neural Computing and Applications, 33(7), 2769-2781.
37. Wang, G., Gan, X., Cao, Q., & Zhai, Q. (2023). MFANet: multi-scale feature fusion network with attention mechanism. The Visual Computer, 39(7), 2969-2980.
38. Lyu, Y., Xu, Y., Jiang, X., Liu, J., Zhao, X., & Zhu, X. (2023). AMS-PAN: Breast ultrasound image segmentation model combining attention mechanism and multi-scale features. Biomedical Signal Processing and Control, 81, 104425.
39. Wang, H., Xie, J., Xu, X., & Zheng, Z. (2022). Few-shot PCB surface defect detection based on feature enhancement and multi-scale fusion. IEEE Access, 10, 129911-129924.
40. Hou, Y., & Zhang, X. (2025). A lightweight and high-accuracy framework for Printed Circuit Board defect detection. Engineering Applications of Artificial Intelligence, 148, 110375.
41. Liu, G., & Wen, H. (2021). Printed circuit board defect detection based on MobileNet-Yolo-Fast. Journal of Electronic Imaging, 30(4), 043004-043004.
42. Zhu, L., Wang, S., Chen, M., Zhu, Y., Xing, K., & Shen, A. (2025). Subtle defect detection network: more accurately detect subtle defects on the printed circuit board surface. Engineering Applications of Artificial Intelligence, 152, 110765.
43. Chen, W., Meng, S., & Wang, X. (2024). Local and global context-enhanced lightweight CenterNet for PCB surface defect detection. Sensors, 24(14), 4729.
44. Lang, D., & Lv, Z. (2025). : simple and effective PCB surface defect detection method. Scientific Reports, 15(1), 1-15.
45. Zhang, L., Chen, J., Chen, J., Wen, Z., & Zhou, X. (2024). LDD-Net: Lightweight printed circuit board defect detection network fusing multi-scale features. Engineering Applications of Artificial Intelligence, 129, 107628.
46. Yu, S., Pan, F., Zhang, X., Zhou, L., Zhang, L., & Wang, J. (2025). A lightweight detection algorithm of PCB surface defects based on YOLO. PLoS One, 20(4), e0320344.
47. Rakhimol V. & Maheswari P. Uma. (2022). Restoration of ancient temple murals using cGAN and PConv networks. Computers & Graphics,109,100-110.
48. Banoth Thulasya Naik,Mohammad Farukh Hashmi & Aditya Gupta. (2024). EIoU-distance loss: an automated team-wise player detection and tracking with jersey colour recognition in soccer. Connection Science,36(1).
49. Glajemir E. Bautista,Paula Angelyn D. Gargar & Ramon G. Garcia. (2025). Real-Time Object Detection in Tap Water Utilizing YOLOv8 for Comprehensive Contamination Monitoring. Engineering Proceedings,92(1),93-93.
50. Md. Sakib Bin Islam,Muhammad E. H. Chowdhury,Mazhar Hasan Zia,Saad Bin Abul Kashem,Molla E. Majid,Ali K. Ansaruddin Kunju... & Mohammad Nashbat. (2025). VisioDECT: a novel approach to drone detection using CBAM-integrated YOLO and GELAN-E models. Neural Computing and Applications,(prepublish),1-24.