



Article

Advancements in Electric Vehicle PCB Inspection: Application of Multi-Scale CBAM, Partial Convolution, and NWD Loss in YOLOv5

Hanlin Xu, Li Wang ^{*} and Feng Chen

College of Electrical Engineering, Nantong University, Nantong 226019, China;
xhl99125@stmail.ntu.edu.cn (H.X.); hardwk@ntu.edu.cn (F.C.)

* Correspondence: lwee@ntu.edu.cn

Abstract: In the rapidly evolving electric vehicle industry, the reliability of electronic systems is critical to ensuring vehicle safety and performance. Printed circuit boards (PCBs), serving as a cornerstone in these systems, necessitate efficient and accurate surface defect detection. Traditional PCB surface defect detection methods, like basic image processing and manual inspection, are inefficient and error-prone, especially for complex, minute, or irregular defects. Addressing this issue, this study introduces a technology based on the YOLOv5 network structure. By integrating the Convolutional Block Attention Module (CBAM), the model's capability in recognizing intricate and small defects is enhanced. Further, partial convolution (PConv) replaces traditional convolution for more effective spatial feature extraction and reduced redundant computation. In the network's final stage, multi-scale defect detection is implemented. Additionally, the normalized Wasserstein distance (NWD) loss function is introduced, considering relationships between different categories, thereby effectively solving class imbalance and multi-scale defect detection issues. Training and validation on a public PCB dataset showed the model's superior detection accuracy and reduced false detection rate compared to traditional methods. Real-time monitoring results confirm the model's ability to accurately detect various types and sizes of PCB surface defects, satisfying the real-time detection needs of electric vehicle production lines and providing crucial technical support for electric vehicle reliability.



Citation: Xu, H.; Wang, L.; Chen, F. Advancements in Electric Vehicle PCB Inspection: Application of Multi-Scale CBAM, Partial Convolution, and NWD Loss in YOLOv5. *World Electr. Veh. J.* **2024**, *15*, 15. <https://doi.org/10.3390/wevj15010015>

Academic Editor: Ghanim A. Putrus

Received: 24 November 2023

Revised: 14 December 2023

Accepted: 20 December 2023

Published: 3 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: electric vehicle; PCB; surface defect detection; YOLOv5; CBAM; PConv; multi-scale; NWD loss function

1. Introduction

Quality and safety have always been the most critical requirements in automotive design, as they are directly linked to the safety of drivers and passengers. In recent years, the rapid development of new energy vehicles, exemplified by electric vehicles, has led to the widespread use of PCBs [1]. As illustrated in Figure 1, PCBs are abundantly present in EVs, serving as the medium for communication and electric power transmission. Core components of electric vehicles, such as batteries, electric drives, and electronic controls, heavily rely on PCBs. If the PCBs used in vehicles are flawed, the consequences can range from the accelerated deterioration of components to the failure of key functions, posing a threat to driving safety. It can be said that PCBs are not only the foundation of modern EV electronic systems but also directly influence the performance and safety of the vehicles. Therefore, conducting defect detection on PCBs used in vehicles is essential to reduce the risk of vehicle failure [2].

Firstly, as PCBs are the core components of numerous electronic systems in electric vehicles, any defects in them can directly affect the performance and reliability of these systems [3]. For example, critical components such as the battery management system, motor control unit, charging system, and onboard infotainment systems all rely on the

proper functioning of PCBs. Defects in PCBs might lead to interruptions or errors in electronic signal transmission, potentially causing a decline in system performance. This could manifest in various ways, including reduced battery charging efficiency, sluggish response of the electric motor, or the failure of navigation and entertainment systems. In the worst-case scenario, these defects might lead to failures in safety systems, such as brake assist or emergency braking systems, which in extreme cases could result in accidents. Moreover, surface defects on PCBs could lead to electrical short circuits, causing the battery to overheat, potentially leading to fires or explosions [4,5]. Such faults pose not only a risk to passenger safety but can also cause significant property damage. Due to the dependence of electric vehicles on complex electronic systems, the reliability of PCBs becomes a crucial factor in ensuring overall vehicle performance and passenger safety. The presence of defects could necessitate frequent vehicle maintenance, increasing maintenance costs, and also affecting consumer trust and satisfaction with the brand [6].

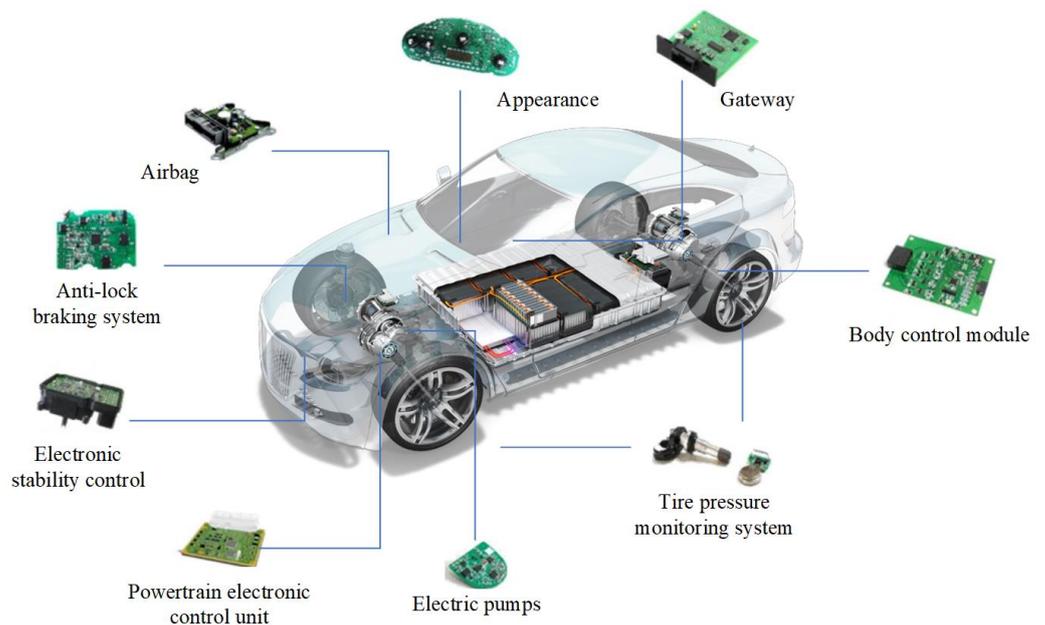


Figure 1. Electric vehicle structure.

In summary, defects on the surface of PCBs can have a widespread negative impact on the performance, safety, reliability, and overall consumer experience of electric vehicles. Thus, conducting efficient and precise PCB defect detection becomes especially critical during the manufacturing process of electric vehicles [7,8]. Traditional methods of detection, such as manual inspection or simple image processing techniques, face challenges when dealing with complex, minute, or irregular defects and struggle to meet the strict standards for detection speed and accuracy required in the EV industry. Therefore, developing new detection technologies to enhance the reliability of electronic systems in electric vehicles is an important task in the current landscape.

In recent years, deep learning and convolutional neural networks (CNNs) have made remarkable progress in the field of image processing and recognition. Particularly, real-time object detection algorithms, such as the YOLO (You Only Look Once) series, have provided powerful tools for real-time image detection [9]. YOLOv5, known for its efficient detection speed and high accuracy, has garnered widespread attention. However, directly applying the original YOLOv5 model may not fully meet the specific requirements of PCB defect detection, such as issues with class imbalance, multi-scale defects, and complex backgrounds [10].

To address the aforementioned challenges, Ali Sezer and Aytaeli Altan proposed an optimized deep learning model for detecting post-soldering defects in PCBs, utilizing 2D signal processing methods [11]. Despite the availability of advanced sensing technologies,

setting pass/fail criteria based on a limited number of failure samples has always been a challenge. To overcome these issues, Jungsuk Kim and colleagues introduced an advanced PCB inspection system based on a convolutional autoencoder with skip connections [12]. For comprehensive automation of the detection process, Yehonatan Fridman and others proposed an automated, integrated change detection system named ChangeChip. This system, based on computer vision and unsupervised learning, can detect a range of issues, from soldering defects to missing or misplaced electronic components [13]. Bing Hu developed a new network based on Faster R-CNN, utilizing ResNet50 with a feature pyramid network as the backbone for feature extraction, enhancing the detection of small defects on PCBs. Additionally, GARPNet was used for more accurate anchor prediction, and residual units from ShuffleNetV2 were integrated [14]. To address the challenge of achieving high detection accuracy, fast detection speed, and low memory consumption simultaneously, Xinting Liao and colleagues improved the activation functions in the backbone and neck prediction networks of YOLOv4, yielding results superior to other state-of-the-art detectors compared [15]. To enhance sensitivity to small defects, Wang Xuan and team proposed a new lightweight deep learning-based defect detection network named YOLOX-MC-CA. This network, developed on the basis of YOLOX, adopted a coordinate attention mechanism to improve the recognition of small PCB surface defects and modified the backbone network of YOLOX to a new CSPDarknet structure with some inverted residual blocks [16].

This study proposes a customized network structure based on YOLOv5, incorporating the NWD loss in its loss function to enhance the accuracy and robustness of the model in detecting PCB surface defects. The YOLOv5 model has been appropriately adjusted in terms of parameters and optimized in its network structure to meet the specific demands of PCB defect detection. Particularly, this is achieved by introducing the CBAM along with additional connections and convolution layers, facilitating multi-scale feature fusion and enhancing the model's attention mechanism. Furthermore, custom anchor sizes have been defined to accommodate defects of varying sizes and shapes.

This research aims to provide an effective technological approach in the electric vehicle industry for enhancing the quality of electronic components, thereby improving overall vehicle performance and safety. It is dedicated to exploring the application of deep learning technology in PCB defect detection and how the optimization of network structure and design of loss functions can improve the model's detection performance. The model was trained and validated on a public PCB image dataset. Preliminary results show that compared to the YOLOv5 model based on traditional loss functions, the model proposed in this study demonstrates a significant advantage in detection accuracy and false detection rate, particularly in detecting complex and minute defects. These technological advancements are expected to contribute to significant improvements in production efficiency and product reliability in the electric vehicle industry.

The contribution of this study lies in the innovative enhancement of the YOLOv5 framework, integrating multi-scale CBAM, partial convolution, and NWD loss to enhance the accuracy of defect detection in electric vehicle PCBs. Experiments on a public PCB dataset demonstrate significant advantages in detection precision and reduced false positive rates, compared to existing technologies. These advancements offer new perspectives in intelligent manufacturing and automated inspection.

2. Method Theory

In response to issues such as low accuracy, imprecise recognition, and inefficiency in PCB surface defect detection within the electric vehicle manufacturing industry, this paper proposes a version of YOLOv5 that incorporates a multi-scale fusion attention mechanism. The network structure is shown in Figure 2 below. Because PCB defects range in size from tiny scratches to larger damage. Compared with the original YOLOv5 model, the detection head of this network integrates features from multiple scales. Multi-scale feature fusion allows the model to capture defects of these different sizes at the same time, thus improving

the accuracy and robustness of detection. With specific improvements categorized into the following three points:

- Multi-scale fusion with CBAM attention mechanism: The integration of multi-scale fusion allows the model to incorporate feature maps from various resolutions, thereby enhancing its ability to recognize defects of different sizes. At the same time, the attention mechanism aids the model in capturing subtle features of these variations, improving the model's generalization capabilities across diverse PCB samples;
- Using partial convolution in the place of traditional convolution: Partial convolution is particularly effective in scenarios with irregular shapes or missing data, which is advantageous for detecting defects with unclear edges. Additionally, partial convolution reduces redundant computation and efficiently accesses memory, thereby enhancing the detection efficiency;
- Introduction of the NWD (normalized Wasserstein distance) loss function: The NWD loss provides smoother gradients, which helps avoid issues like gradient vanishing or exploding during the training process. Moreover, by more accurately measuring the differences between distributions, the NWD loss function aids in improving the model's generalization ability for unseen data.

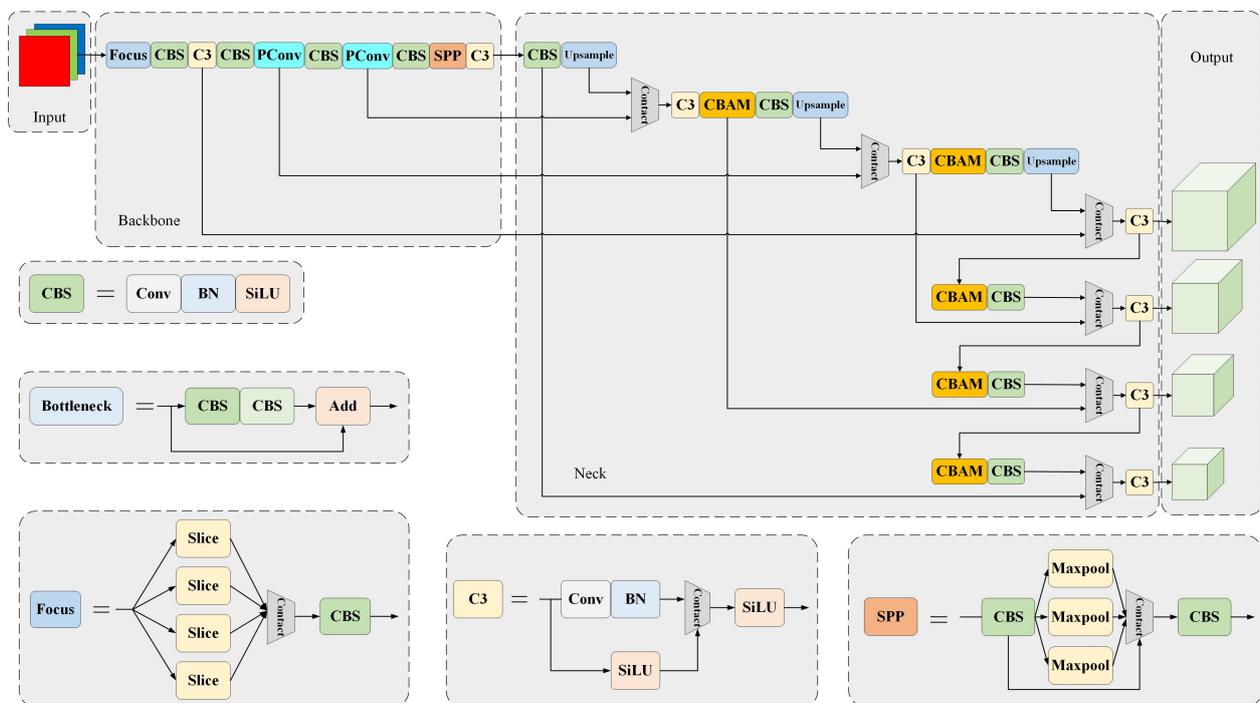


Figure 2. The network structure of the model.

2.1. CBAM

CBAM is an attention mechanism module designed to enhance the performance of convolutional neural networks. It is a combination of spatial and channel attention [17]. The introduction of this structure is because minor changes in certain areas may be key indicators of defects in PCB defect detection. CBAM helps the model to focus on these crucial areas by enhancing important features and suppressing less significant ones, thereby improving the expressiveness of features. This enhancement enables easier differentiation between normal areas and those with defects, especially in cases where the defects are not obvious or there is a significant amount of background noise, ultimately improving the accuracy of detection. Its structure is shown in Figure 3.

Channel attention module (CAM): This module focuses on the importance of each channel in the input feature map. It computes the global average value for each channel through global average pooling and then processes these global averages using two fully

connected layers to produce a vector representing the channel weights [18]. Finally, this vector is normalized using a Sigmoid activation function, yielding the weight coefficients for each channel. This allows the model to automatically learn and select channels that are more useful for target classification and feature representation.

Spatial attention module (SAM): This module focuses on the importance of each spatial location in the input feature map. It calculates the maximum value for each channel using channel max pooling and processes these maxima through two convolution layers to produce a feature map representing spatial weights [19]. This feature map is then element-wise multiplied with the input feature map to enhance the information at important spatial locations and suppress less important ones.

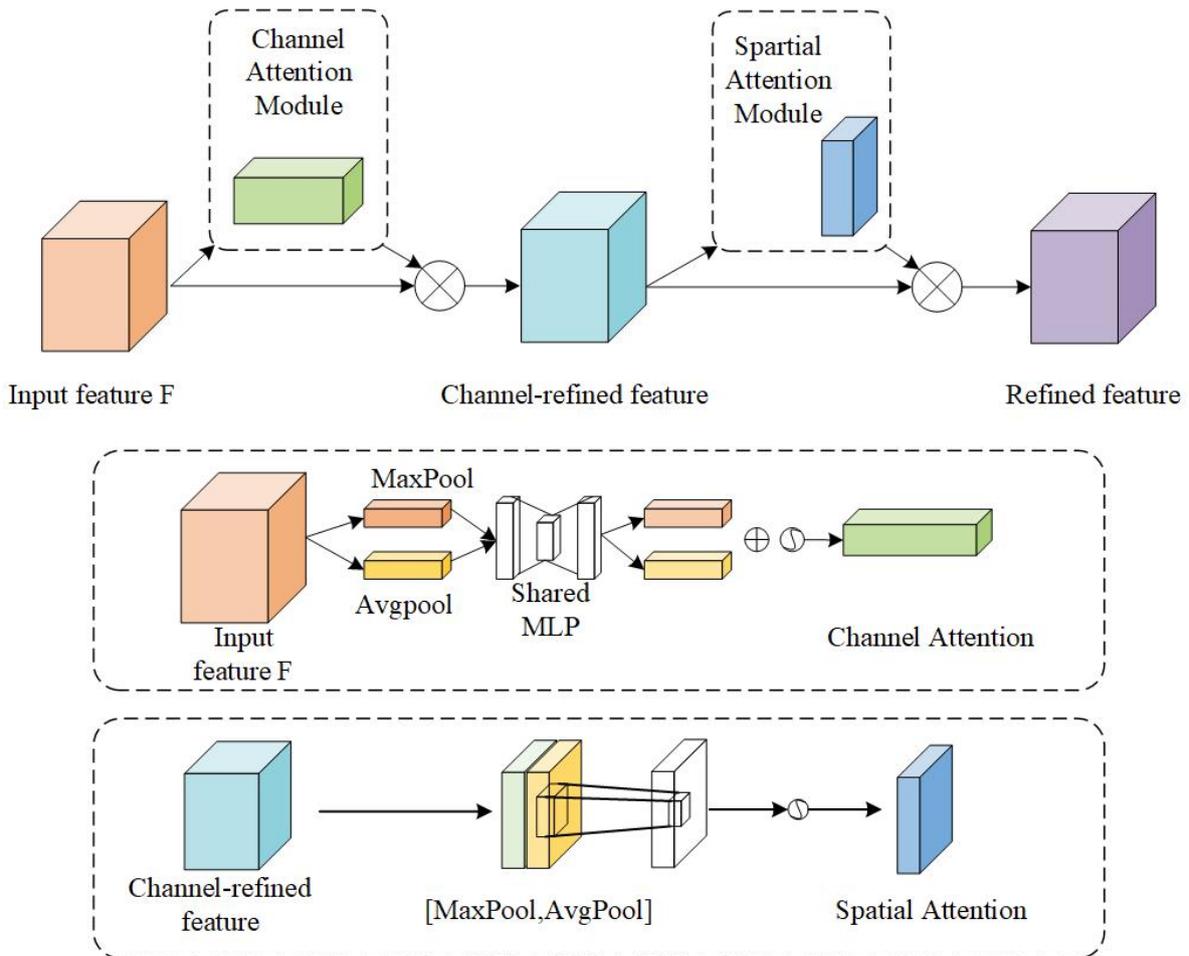


Figure 3. Schematic diagram of CBAM structure.

Channel attention is learning the weights of different channels and multiple different channels with weights, and enhancing attention to key channel domains [20]. For a feature map of $F \in R^{(C \times H \times W)}$ layers, where C denotes the number of channels and H and W denote the length and width of the feature map in pixels, the channel attention module first calculates the weights for each channel $M_C \in R^{(C \times 1 \times 1)}$ according to the following Equation (1).

$$M_C(F) = \sigma(W_1(W_2(F_{avg}^C)) + W_1(W_2(F_{max}^C))). \tag{1}$$

In the above equation, F_{avg}^C and F_{max}^C represent the averaged and maximally pooled feature maps, W_1 and W_2 represent the weights of the two layers of multilayer perceptron, and σ is the sigmoid activation function. The channel attention feature map is then obtained by multiplying with the original feature map.

The feature map is sent to the spatial attention module. Spatial attention focuses on the positional information of objects and selectively aggregates the spatial features of each space through the weighted sum of spatial features. Taking the channel-focused feature map as input, maximal pooling and average pooling are performed in sequence, as shown in Equation (2). Then, the spatial attention weight map is obtained through convolution with a 7×7 kernel, as shown in Equation (3).

$$F_S = \frac{1}{c} \sum_{i \in c} F_C(i) + \max_{i \in c} F_C(i) \quad (2)$$

$$M_S = \sigma(f^{(7 \times 7)}(F_S)). \quad (3)$$

2.2. PConv

In practical vehicle assembly line detection scenarios, models often need to operate on resource-constrained embedded or edge devices, making model light-weighting key to achieving efficient, real-time detection. To reduce the number of parameters in the network model without sacrificing performance, PConv is employed in place of traditional convolution operations. As shown in Figure 4, traditional convolution requires multi-channel convolutional kernels to traverse the entire input data. In contrast, the fundamental idea of PConv is to perform regular convolution operations only on a portion of the channels in the input feature map for spatial feature extraction, while keeping the information in other channels unchanged.

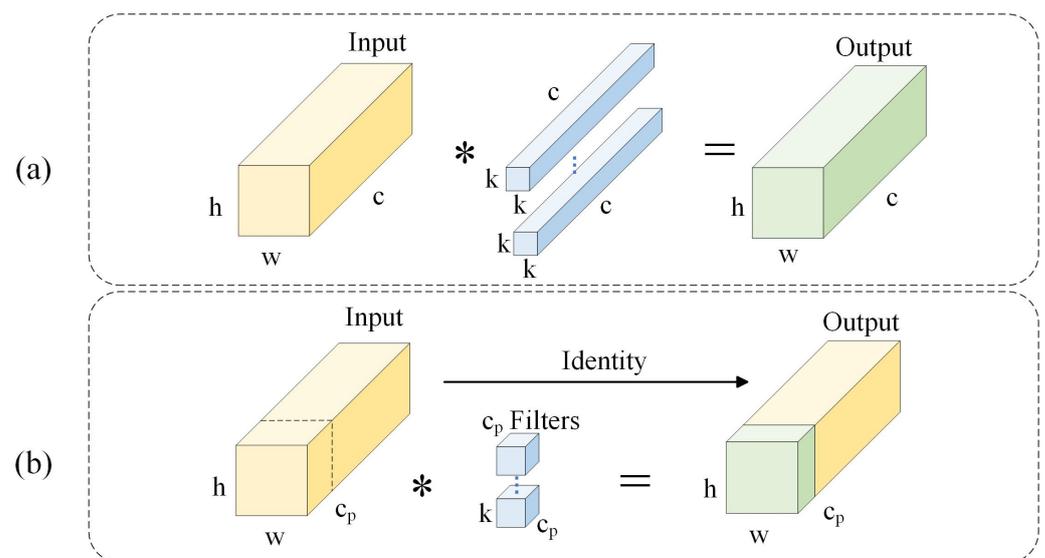


Figure 4. (a) Convolution with $k \times k \times c$ convolution kernels; (b) Partial convolution.

As shown in Figure 5, to ensure no channel information is lost during PConv, pointwise convolution (PWConv) usually follows. PWConv approximates regular convolution in feature transformation, facilitating the more efficient capture and preservation of spatial features [21]. This combination also brings the model's performance on the receptive field closer to that of T-shaped convolution. As shown in Figure 5b, T-shaped convolution, a special convolution operation, focuses on the central area of the feature map for efficient central feature processing without involving selective channel processing. In contrast, PConv and PWConv focus on a portion of the input channels, enhancing efficiency by reducing redundant computations and memory access. This combination not only inherits the central concentration characteristic of T-shaped convolution but also further reduces floating-point operations by decomposing its computational process and leveraging filter redundancy [22].

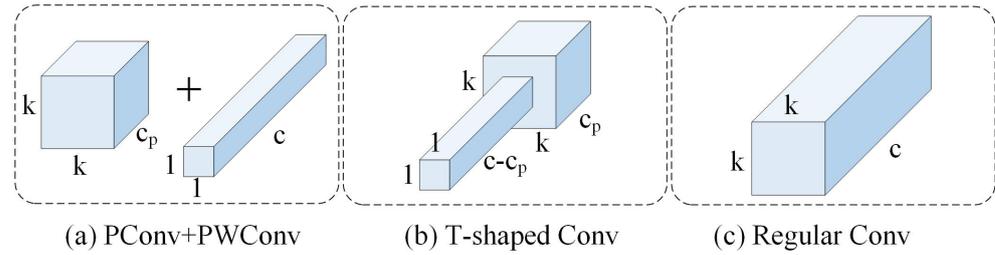


Figure 5. Schematic of three different convolutions: (a) Schematic diagram of PConv and PWConv structure; (b) Schematic diagram of T-shaped Conv structure; (c) Schematic diagram of regular Conv structure.

To ensure representativeness in the analysis, it is assumed that the number of channels in the input and output feature maps remains consistent. In traditional convolution operations, as illustrated in Figure 4a, assume that the dimensions of the input feature map are height h , width w , and number of channels c . *FLOPs* can be calculated using the following formula to quantify the computational complexity of the convolution operation [23]. This approach provides a basis for further analysis and optimization of the convolutional neural network structure, thereby reducing the consumption of computational resources while ensuring network performance.

$$FLOPs = h \times w \times k^2 \times c^2. \quad (4)$$

However, PConv *FLOPs* calculation is only

$$FLOPs = h \times w \times k^2 \times c_p^2. \quad (5)$$

In this formula, c_p represents the number of consecutive channels at the beginning or end of the input feature map, indicating the channels involved in the convolution operation. As depicted in Figure 4b, the dotted part in front of the input feature is chosen to represent the entire feature map for calculations.

It can be seen from Equations (4) and (5) that, when the ratio of c_p to c is $\frac{1}{4}$, the calculation amount of PConv is only $\frac{1}{16}$ times that of the existing traditional convolution, which greatly improves the operating efficiency.

In discussing the efficiency and efficacy of convolution operations, T-shaped convolution and PConv are two common convolution structures. T-shaped convolution, while capable of efficient computation during feature filtering, tends to have higher consumption in terms of computational load and memory access compared to PConv. This leads to computational redundancy and lower time efficiency [24]. To quantify the performance differences between these convolution structures, T-shaped convolution calculates *FLOPs* as shown in Equation (6). PConv and PWConv are calculated according to Equation (7).

$$FLOPs = h \times w \times \left(k^2 \times c_p \times c + c \times (c - c_p) \right) \quad (6)$$

$$FLOPs = h \times w \times \left(k^2 \times c_p^2 + c^2 \right). \quad (7)$$

By comparing Equations (6) and (7), T-shaped convolution needs $(1 - k^2 c_p)(c - c_p)$ more operations. This difference highlights the relatively lower computational efficiency of T-shaped convolution and provides a direction for optimizing convolution structures to reduce computational complexity.

2.3. NWD Loss Function

In the YOLO loss function, the localization loss typically relies on Intersection over Union (IoU) to measure the match degree between predicted and actual bounding boxes. However, IoU is particularly sensitive to minor deviations in bounding boxes, especially

for smaller objects, potentially leading to instability in the loss function. To resolve these problems, this paper incorporates the NWD loss function. It models bounding boxes as Gaussian distributions and uses the Wasserstein distance to more accurately measure the differences between the predicted and actual bounding boxes [25]. This improves the model’s detection capability for small objects, enhances spatial accuracy, better handles overlapping objects, and offers a more stable and balanced training process. The loss is then calculated through normalization, with the entire process depicted in Figure 6.

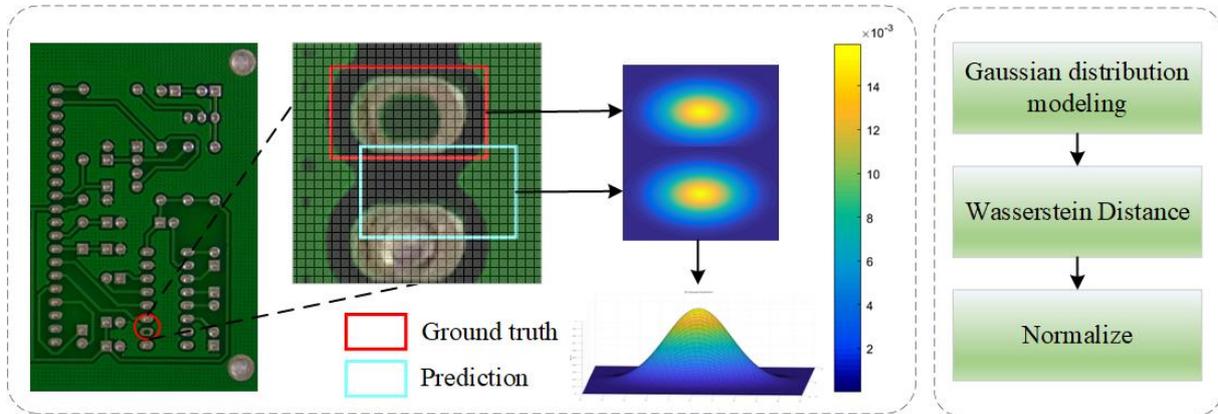


Figure 6. Loss function improvement schematic.

For smaller-scale objects, which in practice are mostly not strictly rectangular and tend to occupy only a few pixels in the middle of the bounding box, usually concentrated around the center, while irrelevant elements like the background are distributed near the edge areas. To more accurately represent the importance of different pixels within the bounding box, a two-dimensional (2D) Gaussian distribution can be used to model the bounding box [26,27]. In this model, the central pixels of the bounding box receive the highest weight, and the importance of pixels gradually decreases from the center towards the edges.

Specifically, for the horizontal bounding box $R = (c_x, c_y, w, h)$, where (c_x, c_y) , w , and h represent the center coordinates, width, and height, respectively. The equation of its interior ellipse can be expressed accordingly:

$$\frac{(x - \mu_x)^2}{\sigma_x^2} + \frac{(y - \mu_y)^2}{\sigma_y^2} = 1, \tag{8}$$

where (μ_x, μ_y) are the center coordinates of the ellipse. σ_x, σ_y are the lengths of the semi-axes along the x and y axes. $\mu_x = c_x, \mu_y = c_y, \sigma_x = \frac{w}{2}, \sigma_y = \frac{h}{2}$.

The probability density function of the 2D Gaussian distribution is given by the following equation:

$$f(x|\mu, \Sigma) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{(x - \mu)^T}{2\Sigma(x - \mu)}\right), \tag{9}$$

where x, μ , and Σ represent the coordinates (x, y) , the mean vector, and the covariance matrix, respectively. The Wasserstein distance is next used to measure the difference between two probability distributions. For two two-dimensional Gaussian distributions $\mu_1 \sim N(m_1, \Sigma_1), \mu_2 \sim N(m_2, \Sigma_2)$, the second-order Wasserstein distance between μ_1 and μ_2 is defined as follows:

$$W_2^2(\mu_1, \mu_2) = \|m_1 - m_2\|_2^2 + \text{Tr}[\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{\frac{1}{2}}\Sigma_2^{\frac{1}{2}})^{\frac{1}{2}}]. \tag{10}$$

The above equation can be simplified to:

$$W_2^2(\mu_1, \mu_2) = \|m_1 - m_2\|_2^2 + \left\| \Sigma_1^{\frac{1}{2}} - \Sigma_2^{\frac{1}{2}} \right\|_F^2. \quad (11)$$

Thus, for two Gaussian distributions modelling frames N_a, N_b , the distance metric can be expressed as the following equation:

$$W_2^2(N_a, N_b) = \left\| \left(\begin{bmatrix} c_{x_a}, c_{y_a}, \frac{w_a}{2}, \frac{h_a}{2} \end{bmatrix}^T, \begin{bmatrix} c_{x_b}, c_{y_b}, \frac{w_b}{2}, \frac{h_b}{2} \end{bmatrix}^T \right) \right\|_2^2. \quad (12)$$

However, distance metrics cannot be used directly as similarity measures. Therefore, the paper employs the Softmax function for normalization. The Softmax function can convert a set of arbitrary real numbers into a probability distribution, which is particularly suitable for extracting similarity information from distance metrics. Taking the negative of the ratio of the square root of the Wasserstein distance to a constant C confines the output results within the 0 to 1 range, thus creating a new metric named *NWD*:

$$NWD(N_a, N_b) = \exp\left(-\frac{\sqrt{W_2^2(N_a, N_b)}}{C}\right), \quad (13)$$

where C is a constant closely related to the data set. In order for the frame loss to reflect the information of both *NWD* and *IoU* similarity measures, this paper controls the relative contributions of *NWD* and *IoU* in the total frame loss, and the values of the weights of both are set to 0.5, which ensures that the contributions of *NWD* and *IoU* to the total frame loss are equal. The final loss function expression is:

$$Loss = \frac{1}{2N} \sum_{i=1}^N (1 - NWD_i) + \frac{1}{2N} \sum_{i=1}^N (1 - IoU_i), \quad (14)$$

where N is the number of detection frames, and the mean value is taken in order to aggregate the loss of multiple targets to a single value that can be used for the further computation and optimization of the loss function.

3. Experimental Section

In this subsection, the paper will evaluate the performance of the present model in PCB defect detection through an exhaustive series of experiments. The model will be compared with several leading methods, reflecting the ability of this paper's model to identify PCB defects more accurately.

3.1. Data Set Processing and Training

The data set used in this article is a public synthetic PCB data set from the Intelligent Robot Development Laboratory of Peking University. According to literature [28], although the performance of synthetic data is slightly inferior to that of real data, synthetic PCB defects still have a positive effect in increasing the diversity of training data in the absence of a large amount of real defect data, and in literature [29] show that, even if the training data differ from the target data in some features, the model can be effectively transferred from synthetic data to real data through appropriate domain adaptation techniques. The data set comprises 1386 images and six types of defects: missing hole, mouse bite, open circuit, short, spur, and spurious copper, as shown in Figure 7. The synthetic defects are highly similar to the real defects, which to a certain extent makes up for the problem of insufficient data and helps to improve the generalization ability and accuracy of the model in real scenarios. The average size of each defect pixel is 130×110 , and each image has a resolution of 2777×2138 pixels, with each defect occupying approximately 0.24% of the entire image. This represents a significant challenge for the model's performance. For this

study, 535 images were used as the training set and 158 images as the validation set. The final results showed a marked improvement.

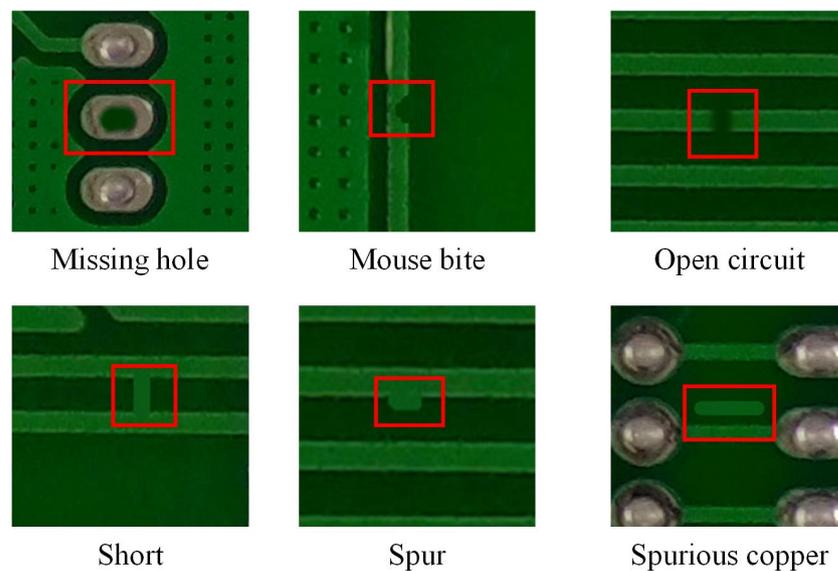


Figure 7. PCB data set composition.

3.2. Evaluation Metrics

To quantify the effectiveness of the model's detection capabilities, this paper employs four evaluation metrics. Precision refers to the ratio of correctly detected positive samples to all detected positive samples, including false positives. Recall is the ratio of correctly detected positive samples to all actual positive samples, including those not detected. Mean average precision (mAP) is the average of precision values across multiple categories and recall levels. Frames per second (FPS) is used to measure the timeliness of the model's detection. Each metric evaluates the model's performance from a different perspective. The formulas for these calculations are as follows:

$$P = \frac{TP}{TP + FP} \quad (15)$$

$$R = \frac{TP}{TP + FN} \quad (16)$$

$$AP = \int_0^1 P(R) dR \quad (17)$$

$$mAP = \frac{\sum_{n=1}^M AP_n}{M} \quad (18)$$

Here, *TPs* (True Positives) refer to the number of positive sample targets correctly identified, i.e., the targets correctly recognized. *FNs* (False Negatives) refer to the number of positive sample targets not correctly identified, i.e., the targets that are missed. *FPS* (False Positives) refer to the number of non-positive sample targets that are incorrectly identified as positive, i.e., the targets that are falsely detected.

3.3. Model Training

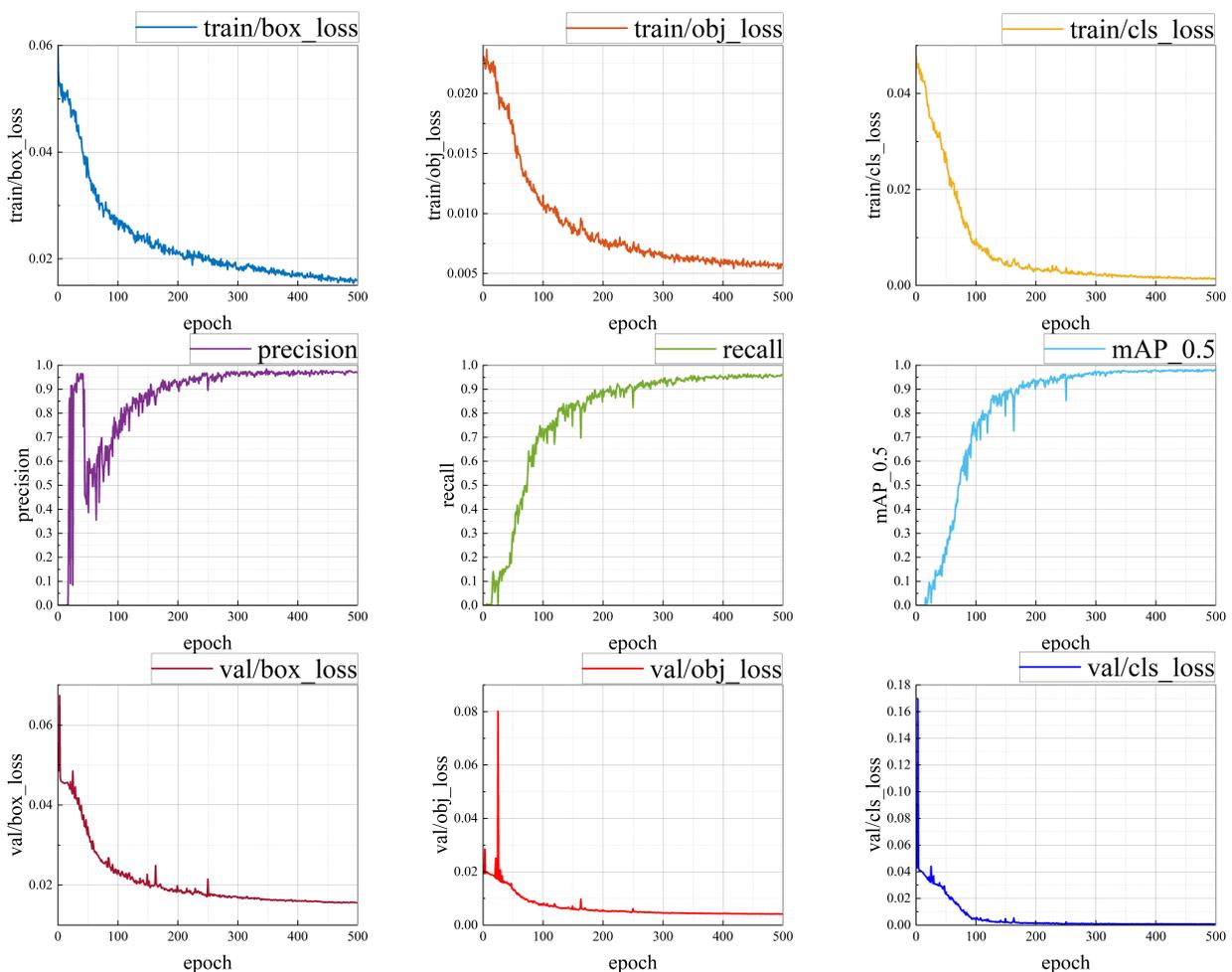
3.3.1. Model Results

In this experiment, the PyCharm-integrated development environment was used and Python 3.8 was chosen as the programming language. The experiment was run on an efficiently configured hardware platform to ensure smooth execution of computational tasks. Table 1 provides detailed hardware and software configuration information:

Table 1. Configuration table of experimental platform.

Experimental Platform	Specific Model
CPU	Intel(R) Core(TM) i7-12700H
GPU	Nvidia GeForce RTX 3060
Operating system	Windows 11 64 bit
Memory	16 GB
Training framework	Pytorch

With the support of the above hardware configurations, the YOLOv5 model proposed in this paper achieves a significantly superior performance on the PCB defect detection task. Taking 500 epochs of training as an example, the training curve is shown in Figure 8.

**Figure 8.** Training results graph.

An analysis of the YOLOv5 model's performance data over 500 training epochs shows a continuous decline in training loss, reflecting improvements in the model's ability to predict defect bounding boxes, objects, and categories. The introduction of the CBAM module, which enhances the model's focus on important features, contributes to an improved detection performance. Consequently, in the later stages of training, the model exhibits very high precision and recall rates. This indicates that the model not only accurately locates and recognizes most defects but also has a very low rate of missing detections. Additionally, the model's mAP performance is robust, particularly approaching 98% at an IoU threshold of 0.5, and maintains a high level even under more stringent IoU thresholds (from 0.5 to 0.95). Towards the end of training, the model's loss stabilizes and remains

low, indicating that the model has converged well. The consistency of validation loss with training loss suggests good generalization ability of the model and the absence of overfitting. The learning rate gradually decreases in line with the training process, aligning with the expected learning rate decay strategy. Overall, the model demonstrates excellent training efficacy and potential for generalization, suggesting its efficiency in detecting PCB surface defects in practical applications.

3.3.2. Ablation Experiments

To thoroughly investigate the impact of different components on model performance, a series of ablation studies were conducted. These experiments individually examined the effects of multi-scale fusion attention CBAM, PConv, and NWD on precision, recall, and mAP, as presented in Table 2.

Table 2. The impact of different components on model performance.

Multi-Scale CBAM	PConv	NWD	P/%	R/%	mAP/%
×	×	×	96.07	93.14	95.68
✓	×	×	96.63	92.50	95.96
×	✓	×	97.63	95.97	97.64
×	×	✓	97.90	95.86	97.52
✓	✓	✓	96.77	96.33	98.13

A value in bold indicates that it is the best value for that indicator.

The data reveal that the individual application of PConv and NWD significantly enhances all performance metrics, while CBAM primarily impacts precision and mAP, contributing less to the improvement in recall. When these three technologies are combined, the model exhibits an optimal performance across all metrics, particularly achieving the highest values in mAP. This suggests that, although each technological component positively contributes to the model's performance, their integrated application provides a more comprehensive enhancement of the model's overall capabilities.

3.3.3. Model Comparison

In order to measure the superiority of the proposed model in this paper, not only is its evaluated performance quantified, but the model is analyzed in comparison with the current popular target detection architectures such as SSD512, YOLOv3, YOLOv5, YOLOv7, FAST R-CNN, and DenseNet [30], as presented in Table 3.

Table 3. Comparison of training results of different models.

Model	P/%	R/%	mAP_0.5/%	mAP_0.5:0.95/%
SSD512	84.07	94.85	92.09	48.79
YOLOv3	85.13	95.36	92.75	49.12
YOLOv5s	93.24	92.43	91.43	50.53
YOLOv7	95.33	95.75	95.08	51.25
Faster R-CNN	90.51	86.48	89.23	49.87
DenseNet	87.35	97.46	94.12	51.39
Proposed	96.77	96.33	98.13	51.16

A value in bold indicates that it is the best value for that indicator.

Table 4 shows the precision of different models for each defect. Table 5 shows the recall of different models for each defect. Table 6 shows the mAP_0.5 of different models for each defect.

The proposed model is particularly good at identifying complex defects such as open and short circuits with an accuracy of more than 96%, showing efficient detection capability and strong robustness.

Table 4. Comparison table of the precision of different models for each type of defect.

Model	Missing Hole/%	Mouse Bite/%	Open Circuit/%	Short/%	Spur/%	Spurious Copper/%
Faster R-CNN	90.3	91.6	90.8	89.6	88.5	92.4
TDD-Net	97.4	94.8	95.3	94.3	97.1	94.9
YOLOv4	89.8	88.8	87.4	90.6	89.8	93.3
YOLOv5s	90.7	95.3	91.9	87.4	96.5	94.6
YOLOv7	98.3	93.2	94.9	93.1	97.2	95.4
YOLO-MBBi [31]	98.5	95.3	96.0	91.8	97.6	95.6
Proposed	98.5	96.0	98.3	96.6	96.9	97.3

A value in bold indicates that it is the best value for that indicator.

Table 5. Comparison table of the recall of different models for each type of defect.

Model	Missing Hole/%	Mouse Bite/%	Open Circuit/%	Short/%	Spur/%	Spurious Copper/%
Faster R-CNN	87.0	84.8	86.7	90.1	83.3	86.1
TDD-Net	98.4	92.6	97.8	96.2	92.5	95.2
YOLOv4	91.2	83.1	87.1	90.4	81.1	90.1
YOLOv5s	91.3	90.8	96.7	92.6	90.0	95.6
YOLOv7	98.9	92.1	98.8	95.1	93.3	95.7
YOLO-MBBi [31]	98.9	92.5	97.2	93.4	90.0	95.7
Proposed	100	90.3	96.5	95.3	95.8	94.5

A value in bold indicates that it is the best value for that indicator.

The proposed model has a very high recall rate on PCB defect detection, especially on the “missing hole”, which reaches 100%. This shows its strong ability in accurately identifying various types of defects, which is very suitable for application scenarios with strict requirements for leakage detection rate.

Table 6. Comparison table of the mAP of different models for each type of defect.

Model	Missing Hole/%	Mouse Bite/%	Open Circuit/%	Short/%	Spur/%	Spurious Copper/%
Faster R-CNN	85.5	88.2	89.5	90.6	90.2	91.4
TDD-Net	97.1	94.5	97.2	91.9	94.0	95.7
YOLOv4	86.0	82.8	85.3	91.4	86.0	94.8
YOLOv5s	85.8	94.2	93.5	89.8	93.8	92.9
YOLOv7	97.7	94.0	97.6	91.8	94.3	96.2
YOLO-MBBi [31]	97.6	94.5	97.3	92.1	94.5	95.7
Proposed	99.3	95.9	99.2	97.0	99.2	97.6

A value in bold indicates that it is the best value for that indicator.

The data from the aforementioned table clearly indicate that the algorithm proposed in this paper performs excellently in this comparative analysis, especially in terms of precision and the mAP_{0.5} metric, where it surpasses all other listed algorithms. Its high accuracy rate of 96.77% means that its predictions are very reliable. The highest mAP_{0.5} score of 98.13% demonstrates the model’s strong detection ability under relatively lenient conditions. While it may not have the highest score in mAP_{0.5:0.95}, its score of 51.16% is still quite high, indicating that it maintains a good performance under various levels of detection difficulty. Compared to other models, the algorithm in this paper has a distinct advantage in terms of overall performance, making it particularly suitable for applications that require extremely high precision.

In comparing the basic YOLOv5 model with the enhanced YOLOv5 model for PCB defect inference detection tasks, a significant performance difference is observed. As shown in Figure 9, the top row in images (a), (b), and (c) shows the inference results of the basic model, while the bottom row shows the results of the model proposed in this paper. Figure 9a illustrates that the basic model misses subtle flaws when faced with the same defects, whereas the model proposed in this paper still correctly classifies the defects.

Figure 9b shows the false detection instances of the basic model, mistaking a short circuit for spurious copper. Figure 9c reflects that, even when detecting the same defects, the basic model usually has lower confidence, increasing the uncertainty in the subsequent analysis. This improvement is evident in two respects: first, the improved model reduces the number of false positives and false negatives, which is crucial for ensuring the completeness of PCB detection. Second, for the defects that are indeed detected, the improved model provides higher confidence scores. This enhanced confidence not only implies more reliable detection results but may also reflect the improved model’s optimization of feature extraction and pattern recognition.

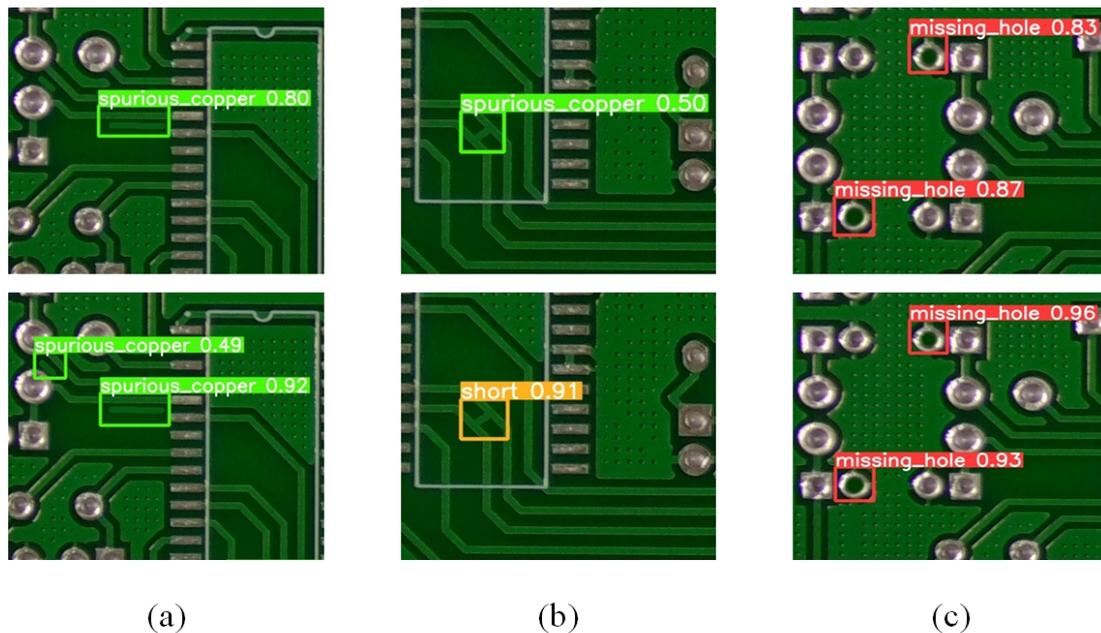


Figure 9. Comparison of inference results.

Detection speed and model size are also key factors for measuring the usefulness of a model. Figure 10 shows a stacked histogram of the percentage of different model sizes and detection FPS [32].

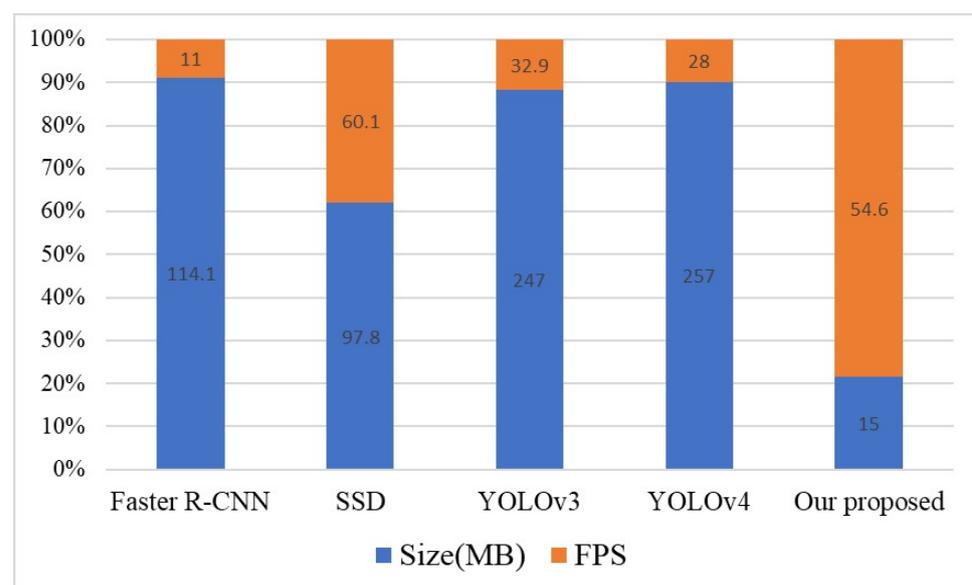


Figure 10. Stacked histogram of size and percentage of detection FPS.

Overall, the model proposed in this paper demonstrates significant advantages in terms of size and speed, reflecting efficient computational performance and lower resource requirements. This makes the model highly suitable for resource-constrained environments and applications that require rapid response, while also reducing the costs associated with storage and deployment.

4. Summary and Outlook

In this research, the proposed enhancements of the YOLOv5 model significantly improve the efficiency and accuracy of PCB surface defect detection, which is of great importance to the electric vehicle industry. By integrating a multi-scale attention mechanism, applying PConv lightweight convolution, and optimizing the loss function, the model achieves an ideal balance between size, speed, and accuracy. These improvements not only enhance the accuracy of defect detection but also contribute to the reliability and safety of the electronic systems in electric vehicles, thereby directly impacting the overall performance of the vehicles. Although the model outperforms traditional methods in several respects, there is still room for improvement in terms of robustness under extreme lighting conditions, and generalization for other datasets. Future research will focus on these challenges, aiming to further enhance the model's generalization ability and adaptability through in-depth network optimization and algorithmic innovations. Additionally, when deploying such models to actual electric vehicle production lines, integration with existing systems, and challenges like data biases and model overfitting encountered in practical operations, must be considered. Through continuous research and collaboration with the industry, it is expected that these theoretical improvements will be translated into reliable solutions in practical applications, bringing higher production efficiency and stronger system reliability to the electric vehicle sector, thereby advancing the industry's technological progress and competitive edge.

Author Contributions: Conceptualization, L.W. and F.C.; manuscript writing, H.X.; image description, F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant 62103205.

Data Availability Statement: The dataset was downloaded from <https://robotics.pkusz.edu.cn/resources/dataset/> (accessed on 11 October 2022).

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

References

1. Dunn, B.; Kamath, H.; Tarascon, J.M. Electrical Energy Storage for the Grid: A Battery of Choices. *Science* **2011**, *334*, 928–935. [[CrossRef](#)] [[PubMed](#)]
2. Brooker, R.P.; Qin, N. Identification of potential locations of electric vehicle supply equipment. *J. Power Sources* **2015**, *299*, 76–84. [[CrossRef](#)]
3. Lee, W.S.; Chang-Chien, G.P.; Wang, L.C.; Lee, W.J.; Tsai, P.J.; Wu, K.Y.; Lin, C. Source Identification of PCDD/Fs for Various Atmospheric Environments in a Highly Industrialized City. *Environ. Sci. Technol.* **2004**, *38*, 4937–4944. [[CrossRef](#)] [[PubMed](#)]
4. *IEEE Std 2030.1.1-2021 (Revision of IEEE Std 2030.1.1-2015)—Redline*; IEEE Standard for Technical Specifications of a DC Quick and Bidirectional Charger for Use with Electric Vehicles—Redline. IEEE: Piscataway, NJ, USA, 2022; pp. 1–263.
5. Li, J.; Gu, J.; Huang, Z.; Wen, J. Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component Detection. *Appl. Sci.* **2019**, *9*, 3750. [[CrossRef](#)]
6. Gu, X.; Ieromonachou, P.; Zhou, L. Subsidising an electric vehicle supply chain with imperfect information. *Int. J. Prod. Econ.* **2019**, *211*, 82–97. [[CrossRef](#)]
7. Wu, H.; Lei, R.; Peng, Y. PCBNet: A Lightweight Convolutional Neural Network for Defect Inspection in Surface Mount Technology. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3518314. [[CrossRef](#)]
8. Lu, Y.; Yang, B.; Gao, Y.; Xu, Z. An automatic sorting system for electronic components detached from waste printed circuit boards. *Waste Manag.* **2022**, *137*, 1–8. [[CrossRef](#)] [[PubMed](#)]
9. Zeng, N.; Wu, P.; Wang, Z.; Li, H.; Liu, W.; Liu, X. A Small-Sized Object Detection Oriented Multi-Scale Feature Fusion Approach With Application to Defect Detection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3507014. [[CrossRef](#)]

10. Li, G.; Zhao, S.; Zhou, M.; Li, M.; Shao, R.; Zhang, Z.; Han, D. YOLO-RFF: An Industrial Defect Detection Method Based on Expanded Field of Feeling and Feature Fusion. *Electronics* **2022**, *11*, 4211. [[CrossRef](#)]
11. Sezer, A.; Altan, A. Detection of solder paste defects with an optimization-based deep learning model using image processing techniques. *Solder. Surf. Mt. Technol.* **2021**, *33*, 291–298. [[CrossRef](#)]
12. Kim, J.; Ko, J.; Choi, H.; Kim, H. Printed Circuit Board Defect Detection Using Deep Learning via A Skip-Connected Convolutional Autoencoder. *IEEE Sens. Counc.* **2021**, *21*, 4968. [[CrossRef](#)] [[PubMed](#)]
13. Fridman, Y.; Rusanovsky, M.; Oren, G. ChangeChip: A Reference-Based Unsupervised Change Detection for PCB Defect Detection. In Proceedings of the 2021 IEEE Physical Assurance and Inspection of Electronics (PAINE), Washington, DC, USA, 30 November–2 December 2021; pp. 1–8.
14. Hu, B.; Wang, J. Detection of PCB Surface Defects With Improved Faster-RCNN and Feature Pyramid Network. *IEEE Access* **2020**, *8*, 108335–108345. [[CrossRef](#)]
15. Liao, X.; Lv, S.; Li, D.; Luo, Y.; Zhu, Z.; Jiang, C. YOLOv4-MN3 for PCB Surface Defect Detection. *Appl. Sci.* **2021**, *11*, 11701. [[CrossRef](#)]
16. Xuan, W.; Jian-She, G.; Bo-Jie, H.; Zong-Shan, W.; Hong-Wei, D.; Jie, W. A Lightweight Modified YOLOX Network Using Coordinate Attention Mechanism for PCB Surface Defect Detection. *IEEE Sens. J.* **2022**, *22*, 20910–20920. [[CrossRef](#)]
17. Xue, Z.; Lin, H.; Wang, F. A Small Target Forest Fire Detection Model Based on YOLOv5 Improvement. *Forests* **2022**, *13*, 1332. [[CrossRef](#)]
18. Wan, G.; Fang, H.; Wang, D.; Yan, J.; Xie, B. Ceramic tile surface defect detection based on deep learning. *Ceram. Int.* **2022**, *48*, 11085–11093. [[CrossRef](#)]
19. Wang, L.; Cao, Y.; Wang, S.; Song, X.; Zhang, S.; Zhang, J.; Niu, J. Investigation Into Recognition Algorithm of Helmet Violation Based on YOLOv5-CBAM-DCN. *IEEE Access* **2022**, *10*, 60622–60632. [[CrossRef](#)]
20. Sun, Z.; Ibrayim, M.; Hamdulla, A. Detection of Pine Wilt Nematode from Drone Images Using UAV. *Sensors* **2022**, *22*, 4704. [[CrossRef](#)]
21. Liu, G.; Dundar, A.; Shih, K.J.; Wang, T.C.; Reda, F.A.; Sapra, K.; Yu, Z.; Yang, X.; Tao, A.; Catanzaro, B. Partial Convolution for Padding, Inpainting, and Image Synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 6096–6110. [[CrossRef](#)]
22. Chen, J.; hong Kao, S.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
23. Li, R.; Zhang, Y.; Niu, D.; Yang, G.; Zafar, N.; Zhang, C.; Zhao, X. PointVGG: Graph convolutional network with progressive aggregating features on point clouds. *Neurocomputing* **2021**, *429*, 187–198. [[CrossRef](#)]
24. Pan, S.; Chen, K.; Chen, J.; Qin, Z.; Cui, Q.; Li, J. A partial convolution-based deep-learning network for seismic data regularization1. *Comput. Geosci.* **2020**, *145*, 104609. [[CrossRef](#)]
25. Wang, J.; Xu, C.; Yang, W.; Yu, L. A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. *arXiv* **2021**, arXiv:2110.13389.
26. Wang, Q.; Yang, L.; Zhou, B.; Luan, Z.; Zhang, J. YOLO-SS-Large: A Lightweight and High-Performance Model for Defect Detection in Substations. *Sensors* **2023**, *23*, 8080. [[CrossRef](#)] [[PubMed](#)]
27. Zhang, J.; Wei, X.; Zhang, L.; Yu, L.; Chen, Y.; Tu, M. YOLO v7-ECA-PConv-NWD Detects Defective Insulators on Transmission Lines. *Electronics* **2023**, *12*, 3969. [[CrossRef](#)]
28. Wang, Q.; Breckon, T. Unsupervised Domain Adaptation via Structured Prediction Based Selective Pseudo-Labeling. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019.
29. Bhowmik, N.; Wang, Q.; Gaus, Y.F.A.; Szarek, M.; Breckon, T. The Good, the Bad and the Ugly: Evaluating Convolutional Neural Networks for Prohibited Item Detection Using Real and Synthetically Composited X-ray Imagery. *arXiv* **2019**, arXiv:1909.11508.
30. Yang, Y.; Kang, H. An Enhanced Detection Method of PCB Defect Based on Improved YOLOv7. *Electronics* **2023**, *12*, 2120. [[CrossRef](#)]
31. Du, B.; Wan, F.; Lei, G.; Xu, L.; Xu, C.; Xiong, Y. YOLO-MBBi: PCB Surface Defect Detection Method Based on Enhanced YOLOv5. *Electronics* **2023**, *12*, 2821. [[CrossRef](#)]
32. Liu, J.; Cui, G.; Xiao, C. A real-time and efficient surface defect detection method based on YOLOv4. *J. Real-Time Image Process.* **2023**, *20*, 77. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.