International Journal of Computer Applications Technology and Research
Volume 14–Issue 09, 01 – 05, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1409.1001

Research on Lightweight PCB Defect Detection Algorithm based on YoloV11

Jiankang Yu*
Chengdu University of Information Technology
College of Communication Engineering
ChengDu, China

JiaCui Tang
Chengdu University of Information Technology
College of Communication Engineering
ChengDu, China

Ziming Tang
Chengdu University of Information Technology
College of Communication Engineering
ChengDu, China

Abstract: With the rapid development of the electronics manufacturing industry, defect detection for printed circuit boards (PCBs) is crucial for product quality control. Traditional inspection methods rely on manual visual inspection or traditional image processing techniques, which suffer from low efficiency and high missed detection rates. Although deep learning-based object detection algorithms have significantly improved detection accuracy, existing models suffer from large parameter counts and high computational complexity, making them difficult to meet the real-time and lightweight deployment requirements of industrial scenarios. To address this, this paper proposes a lightweight PCB defect detection algorithm based on YOLOv11, the YoloV11-CGL model. First, the Context-Guided module is introduced to replace the existing C3k2 module. This module utilizes a multi-stage context fusion mechanism and efficient channel separation computation to significantly reduce parameter size while maintaining pixel-level classification accuracy. Second, the LAE module performs down-sampling, replacing the existing down-sampling. This module utilizes adaptive weight fusion and channel information reorganization to dynamically retain high-entropy pixels during the four-fold down-sampling process, avoiding the edge feature loss of traditional convolution and reducing the number of parameters to 1/N of the traditional method (N is the number of group convolutions). Furthermore, to address the accuracy degradation caused by lightweight models, knowledge distillation is employed to effectively improve model accuracy while minimizing the number of parameters. Experimental results demonstrate that the YoloV11-CGL model achieves excellent results on the public DsPCBSD+ datasets. While significantly reducing the number of parameters by 26%, accuracy remains largely unchanged, making it suitable for the real-time and lightweight deployment requirements of industrial scenarios.
Keywords: YoloV11 model; deep learning; PCB defect detection; Light-weight

1. Introduction

In the electronics manufacturing industry, printed circuit boards (PCBs) are core components of electronic devices, and their quality directly affects the reliability and performance of products. As electronic products develop towards miniaturization and high density, the difficulty of detecting PCB surface defects has increased significantly. Traditional detection methods rely on manual visual inspection or automatic optical inspection (AOI) systems, but the former has problems such as low efficiency and high missed detection rate, while the latter is sensitive to lighting conditions and image alignment accuracy, making it difficult to identify tiny defects in complex backgrounds [1]. In recent years, target detection algorithms based on deep learning have become the mainstream method for PCB defect detection due to their high efficiency. However, existing algorithms still face two major challenges in industrial applications: insufficient model light-weighting and limited small target detection accuracy. For example, the original model of YOLOv5 has limited inference speed on edge devices [2,3], and although YOLOv7-tiny [4,5] has fewer parameters, its feature fusion module has redundant calculations, which affects real-time performance. To this end, this paper proposes a lightweight improvement strategy: by replacing the original C3k2 module and down-sampling module in YoloV11, the number of model parameters can be significantly reduced, achieving the goal of light-weighting. At the same time, knowledge distillation is introduced into the model to effectively restore the model accuracy problem caused by the reduction of parameters.

2. Network Structure

2.1 YoloV11-CGL network structure

In view of the insufficient adaptability of existing target detection algorithms in miniaturized and high-density PCB defect detection scenarios, this study proposes a triple lightweight improvement architecture based on the YoloV11 model, named YoloV11-CGL: 1. Introducing the Context-Guided module [6] to replace the original C3k2 module. This module uses a multi-stage context fusion mechanism and efficient channel separation calculation to significantly compress the parameter scale while taking into account pixel-level classification accuracy. 2. Down-sampling is performed through the LAE module [7] to replace the original down-sampling. This module uses adaptive weight fusion and channel information reorganization to dynamically retain high information entropy pixels during the four-fold down-sampling process, avoiding the edge feature loss problem of traditional convolution, while reducing the parameters to 1/N of the traditional method (N is the number of groups of group convolution). 3. In response to the accuracy drop caused by the lightweight model, knowledge distillation [8,9] is used to effectively improve the accuracy of the model under the constraint of reduced parameters. The network structure diagram of the YoloV11-CGL model is shown below:

International Journal of Computer Applications Technology and Research
Volume 14–Issue 09, 01 – 05, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1409.1001

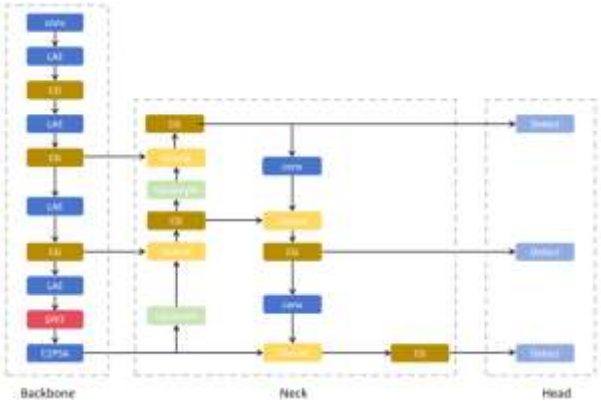

Figure 1 Schematic diagram of YoloV11-CGL network structure

2.2 Context-Guided Module

The Context Guided Block (CGBlock) is the core module of CGNet. It efficiently improves the accuracy and lightweight performance of semantic segmentation by fusing local features, surrounding context, and global context. Its design is inspired by the human visual system's multi-layered understanding of scenes. The module consists of four key components, as shown in Figure 2: $f_{loc}(*)$--a local feature extractor, uses standard 3×3 convolution to extract local features. By sliding the convolution kernel over the entire input data, it extracts local features at different locations, thereby achieving local perception of the input data; $f_{sur}(*)$--a surrounding context extractor, uses 3×3 dilated convolution to expand the receptive field and efficiently obtain contextual information such as co-occurrence relationships in the target's surrounding area; $f_{joi}(*)$--a joint feature extractor, concatenates and fuses the two aforementioned features, forming a joint feature containing multi-layer information through batch normalization (BN) and parameterized ReLU (PReLU); and $f_{glo}(*)$--a global context extractor, uses global average pooling and a multi-layer perceptron (MLP) to generate channel-wise attention weights. It adaptively reweights the joint features, highlighting important information and suppressing redundancy.

The CG block also introduces residual connections to optimize information flow and gradient propagation. Experiments have shown that global residual learning (GRL), which connects the input and the global context weighted output, is more effective than local residual learning (LRL). Its advantage lies in achieving end-to-end fusion of multi-scale context, overcoming the limitation of traditional methods that rely only on a single level of context; at the same time, lightweight designs such as channel-separated convolutions significantly reduce the number of parameters, making it very suitable for mobile deployment. The CG block is embedded in all stages of the network to ensure that CGNet can continuously capture rich spatial and semantic context information from shallow to deep layers, thereby achieving high-precision target detection with extremely small parameters. .

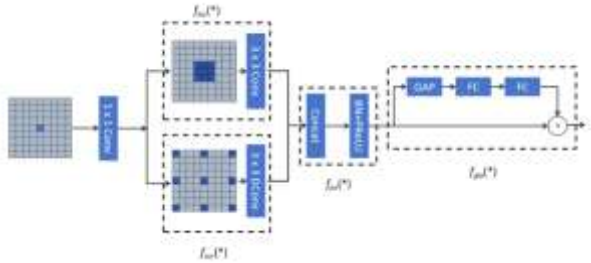

Figure 2 Context-Guided module structure diagram

2.3 LAE Module

To address the need for refined identification and light-weighting of small targets in PCB defect inspection, this paper introduces the Lightweight Adaptive Extraction (LAE) module. Compared to traditional convolutional methods, LAE significantly reduces the number of parameters and computational cost in multi-scale feature extraction, while also extracting features with richer semantic information. The LAE module architecture is shown in Figure 3.

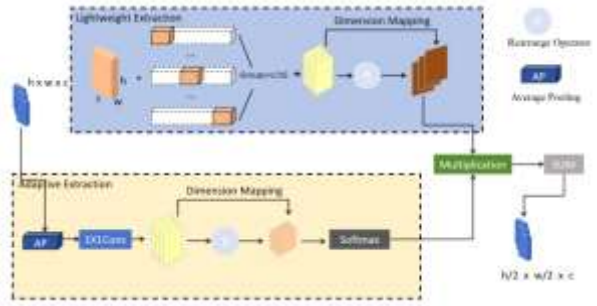

Figure 3 LAE module structure diagram

The LAE module, based on the concept of group convolution, efficiently maps input features to the target output dimension with a low number of parameters. By employing N groups of convolutions, this module significantly reduces the number of parameters to 1/N that of traditional convolution. Each LAE unit performs 4x down-sampling, halving both the height and width of the feature map. To mitigate the loss of edge information during down-sampling, LAE first reorders the spatial information of the feature map to the channel dimension. This operation transforms the feature map from four dimensions (batch, channel, height, weight) to five dimensions (batch, channel, height, weight, N), where N represents the down-sampling factor. The LAE module also includes an adaptive feature extraction path that facilitates information exchange through average pooling and convolution operations. It also performs a reordering operation on the input feature map, equivalent to clustering adjacent 2x2 pixel regions. This path then applies a Softmax function to generate adaptive weights that reflect the importance of each clustered pixel, also outputting a five-dimensional feature map. In the fifth dimension (N), these adaptive weights are combined with the output of another branch (the spatial-to-channel information reordering branch).

2.4 Knowledge Distillation

When training deep neural networks, complex and large models are usually able to extract richer multi-level feature information from image data. Knowledge distillation is an effective model compression technology. Its core idea is to use a teacher model with stronger representation ability to guide the training process of a lightweight student model [10]. In the specific implementation, the category probability distribution output by the teacher model is used as a soft target, and the continuous probability value of each defect category is generated through the Softmax function. Compared with the one-hot vector generated by hard targets - only the highest probability category is marked as 1 and other category information is ignored - soft targets contain richer inter-class relationship knowledge [11]. As shown in the formula:

Soft target:

$$p_i = \frac{exp(z_j/T)}{\sum_j exp(z_j/T)}$$

International Journal of Computer Applications Technology and Research
Volume 14–Issue 09, 01 – 05, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1409.1001

Hard Target:

$$y_i = f(x) = \begin{cases} 1, & arg\,max(z) \\ 0, & otherwise \end{cases}$$

Among them, p_i is the function output after Softmax; z_i is the output probability of category i; z_j is the output probability of category j. T is the temperature variable. When T=1, the function is the original Softmax function; when T tends to 0, the Softmax output is a hard target; when T tends to infinity, the Softmax output becomes a soft target. The larger the T value, the greater the entropy of the Softmax output probability distribution, and the more information the negative label obtains. In the PCB defect detection task, this method enables the student model to maintain low computational consumption while transferring the high-precision feature extraction capability of the teacher model, significantly improving the recognition accuracy of minor defects (such as missing solder joints and broken lines) [12]. The knowledge distillation process is shown in Figure 4:

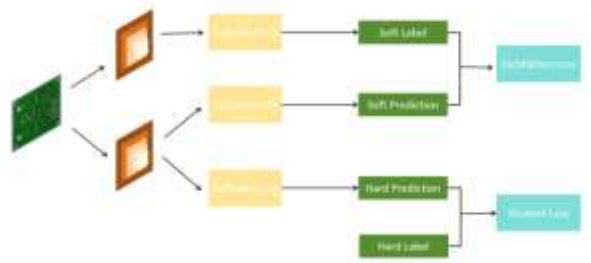

Figure 4 Knowledge distillation process

3. Results and Analysis

3.1 Data sources and characteristics

This paper uses the publicly available PCB surface defect (DsPCBSD+) datasets. This datasets divides PCB surface defects into nine different categories based on factors such as the cause, location, and morphology of the PCB defects, namely, Figure 6(a) short circuit; Figure 6(b) copper spur; Figure 6(c) residual copper; Figure 6(d) open circuit; Figure 6(e) rat bite; Figure 6(f) hole deviation; Figure 6(g) scratch; Figure 6(h) conductive foreign matter; Figure 6(i) substrate foreign matter [13]. Based on this, this experiment divided the datasets into 80% training set, 10% validation set, and 10% test set.

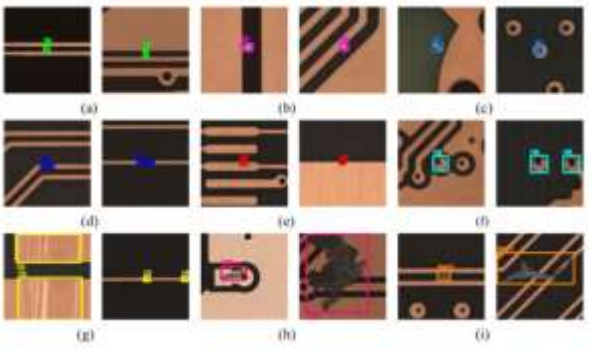

Figure 6 9 defect categories

3.2 Experimental environment and evaluation criteria

The hardware and software configuration used in this experiment is as follows: an Nvidia GeForce RTX 4060 graphics card with 8GB of video memory, an Intel i5 12600kF CPU with 32GB of memory. The software environment configuration is as follows: Windows 10 operating system, PyTorch 2.5.0.

The main network parameters for this training are shown in Table 1:

Table 1. YoloV11-CGL model main training parameters

Parameter Name	Parameter Value
epochs	300
batch	8
imgsize	640
optimizer	SGD
mosaic	4
amp	True
learning_rate	0.001

This experiment uses the number of parameters (Params), the amount of computation (FLOPs), the precision P, the mean average precision (mAP), and the mAP@50-95 index to evaluate the model performance.

(1) The actual correct proportion of predicted positive results (TP / Total Predictions), the percentage of correct predictions, the value range is [0, 1]. Its calculation formula is as follows [14]:

$$P = \frac{TP}{TP + FP}$$

Where TP and FP represent the number of correctly detected true positives and false positives, respectively.

(2) mAP (Mean Average Precision): mAP refers to the average accuracy rate, which is an evaluation indicator of the quality of the model in machine learning [15]. It is usually calculated under different recall rates and the average AP of all categories is taken. Its calculation formula is:

$$AP = \int_0^1 PdR$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N}$$

Where P(R) is the precision value under the precision-recall curve.

(3) mAP50-95 is a more rigorous evaluation metric that calculates the average precision over the recall range of 50% to 95% and takes the average of these values. This allows for a more comprehensive assessment of the model's performance at different recall thresholds.

3.3 Experimental results and analysis

Through systematic ablation experiments, this study quantitatively validated the detection performance of each improved model module. The experimental setup employed a controlled approach, using the baseline YOLOv11 model as a reference. By sequentially comparing the YOLOv11n weights, Context-Guided, LAE, and knowledge distillation modules, the

YOLOv11-CGL model was ultimately developed. The results of each model on the PCB defect detection task are detailed in Table 2.

Table 2 Ablation experiment results on the DsPCBSD+ dataset

Model	Params(M)	FLOPs(G)	P(%)	mAP50(%)	mAP(50-95)(%)
YOLOV11	2.58	6.3	83.8%	84.8%	51.9%
YOLOV11n	2.58	6.3	83.4%	85.4%	52.4%
YoloV11-CG	2.28	5.6	81.5%	84.8%	51.6%
YoloV11-LAE	2.21	6.2	80.2%	85.5%	52.6%
ContextGuided-LAE	1.90	5.4	81.5%	81.5%	51%
YoloV11-CGL	1.90	5.4	82.8%	84.5%	51.6%

Experimental data analysis shows that after the introduction of Context-Guided, LAE, and knowledge distillation modules, the three indicators of precision, mAP50, and mAP50-95 have not changed significantly compared with before the improvement, while the number of parameters and computational complexity have decreased significantly, indicating that the YOLOV11-CGL model has brought significant improvement in the light-weighting of PCB defects.

4. Conclusion

This paper proposes an improved model based on YOLOv11, YOLOv11-CGL, to meet the real-time and lightweight requirements of PCB defect detection in industrial scenarios. This model's complexity is significantly reduced through three innovative designs: 1. Context-Guided Light-weighting: The Context-Guided module replaces the C3k2 module. Through multi-stage context fusion and channel separation calculations, it reduces the number of parameters by 12% while maintaining pixel-level classification accuracy. 2. Adaptive Feature Preservation: The LAE down-sampling module is designed, utilizing spatial-channel reorganization and adaptive weight fusion to reduce the number of down-sampling parameters to 1/N (N is the number of groups) of traditional methods, effectively avoiding edge feature loss. 3. Accuracy Compensation Mechanism: Knowledge distillation technology is introduced to supervise the student model training with soft targets from the teacher model, alleviating the accuracy loss caused by light-weighting. Experiments on the DsPCBSD+ datasets show that YOLOv11-CGL requires only 1.90M parameters and 5.4G FLOPs, which is significantly lighter than the baseline model (2.58M/6.3G), while maintaining 82.8% accuracy and 84.5% mAP50, meeting the needs of industrial real-time detection. However, this study still has the following limitations: First, the model's detection accuracy for small defects (such as micron-level scratches) under complex backgrounds (such as highly reflective substrates and dense component occlusion) still needs to be improved; second, the distillation is highly dependent, and the knowledge distillation effect is constrained by the performance of the teacher model, and the training cycle is extended by 30%; finally, the hardware adaptation has not been verified, and the inference latency has not been measured on embedded devices. The actual deployment efficiency needs further verification. Future work will focus on: (1) quantifying and reducing the computational cost required by the model to enable deployment on lightweight devices; (2) enhancing the detection capability of hidden defects and improving the detection accuracy of the model.

5. References

[1] Chen W, Meng S, Wang X. Local and Global Context-Enhanced Lightweight CenterNet for PCB Surface Defect Detection[J]. Sensors, 2024, 24(14): 4729.

[2] Kim, J. H., Kim, N., Park, Y. W., & Won, C. S. (2022). Object detection and classification based on YOLO-V5 with improved maritime dataset. Journal of Marine Science and Engineering, 10(3), 377.

[3] Yan, H., Huang, J., & Zhou, Z. (2025). Plastic bottle localization and ranging using improved YOLO-PB and binocular stereo vision. Measurement, 118469.

[4] Ma, L., Zhao, L., Wang, Z., Zhang, J., & Chen, G. (2023). Detection and counting of small target apples under complicated environments by using improved YOLOv7-tiny. Agronomy, 13(5), 1419.

[5] Wu T, Tang S, Zhang R, et al. Cgnet: A light-weight context guided network for semantic segmentation[J]. IEEE Transactions on Image Processing, 2020, 30: 1169-1179.

[6] Yan, H., Huang, J., Zheng, M., & Tang, Y. (2025). Zero-shot image segmentation for scene objects based on the L0 gradient minimization and adaptive superpixel method. Neural Computing and Applications, 37(16), 10141-10161.

[7] LSM-YOLO: A Compact and Effective ROI Detector for Medical Detection.

[8] Bai, G., Yan, H., Liu, W., Deng, Y., & Dong, E. (2025). Towards Lightest Low-Light Image Enhancement Architecture for Mobile Devices. Expert Systems with Applications, 129125.

[9] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015.

[10] Yan, H., Huang, J., & Huang, T. (2025). IGDNet: Zero-Shot Robust Underexposed Image Enhancement via Illumination-Guided and Denoising. arXiv preprint arXiv:2507.02445.

[11] Tang, J., et al. (2020). Understanding and Improving Knowledge Distillation.

[12] Hussain, M. (2023). YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital

manufacturing and industrial defect detection. Machines, 11(7), 677.

[13] Lv S, Ouyang B, Deng Z, et al. A dataset for deep learning based detection of printed circuit board surface defect[J]. Scientific Data, 2024, 11(1): 811.

[14] Yan, H., Li, A., Zhang, X., Liu, Z., Shi, Z., Zhu, C., & Zhang, L. (2025). MobileIE: An Extremely Lightweight and Effective ConvNet for Real-Time Image Enhancement on Mobile Devices. arXiv preprint arXiv:2507.01838.

[15] Yan, H., Wu, W., Deng, Z., Huang, J., Li, Z., & Zhang, L. (2022). Image inpainting for 3d reconstruction based on the known region boundaries. Mathematics, 10(15), 2761.