



Incorporating long-tail data in complex backgrounds for visual surface defect detection in PCBs

Liyang Zhu¹ · Sen Wang¹ · Mingfang Chen¹ · Aiping Shen¹ · Xuangang Li¹

Received: 19 March 2024 / Accepted: 6 July 2024
© The Author(s) 2024

Abstract

High-quality printed circuit boards (PCBs) are essential components in modern electronic circuits. Nevertheless, most of the existing methods for PCB surface defect detection neglect the fact that PCB surface defects in complex backgrounds are prone to long-tailed data distributions, which in turn affects the effectiveness of defect detection. Additionally, most of the existing methods ignore the intra-scale features of defects and do not utilize auxiliary supervision strategies to improve the detection performance of the network. To tackle these issues, we propose a lightweight long-tailed data mining network (LLM-Net) for identifying PCB surface defects. Firstly, the proposed Efficient Feature Fusion Network (EFFNet) is applied to embed intra-scale feature associations and multi-scale features of defects into LLM-Net. Next, an auxiliary supervision method with a soft label assignment strategy is designed to help LLM-Net learn more accurate defect features. Finally, the issue of inadequate tail data detection is addressed by employing the devised Binary Cross-Entropy Loss Rank Mining method (BCE-LRM) to identify challenging samples. The performance of LLM-Net was evaluated on a homemade dataset of PCB surface soldering defects, and the results show that LLM-Net achieves the best accuracy of mAP@0.5 for the evaluation metric of the COCO dataset, and it has a real-time inference speed of 188 frames per second (FPS).

Keywords PCB defect detection · Object detection · EFFNet · Auxiliary supervision · BCE-LRM

Introduction

PCBs are widely applied in electronics, communications, computers, security, medicine, industrial control, aerospace, etc. [1] As the carrier of modern large-scale integrated circuits, the quality of PCBs is closely related to the operational efficiency and safety of modern circuits. Unfortunately, PCB in the manufacturing process by the environment and the

influence of personal factors of technicians, resulting in welding spikes as shown in Fig. 1 aiguille, interconnect pad, dissymmetry, holes, and solder residue of these five types of common welding defects, the ensuing short circuit and electrical fire and other hazards to PCB manufacturers and users will bring huge economic losses [2] and even casualties. Detection of PCB defects is essential.

Traditional methods for PCB defect detection include visual inspection, electrical testing [3], and Automatic Optical Inspection (AOI) [4]. Visual inspection by the operator is subject to leakage and mistakes related to the subjective state of the operator [5]. The electrical test method is incapable of detecting short-circuit defects and extensive defects, and the AOI technology is affected by the working environment, technical requirements, hardware equipment, and other factors, which makes its adaptability insufficient. Template matching technology [6] has been applied to PCB surface defect detection with the development of image processing technology. Unfortunately, there are still a large number of targets that are incorrectly detected and overlooked. Machine learning has been applied to PCB surface defect detection [2, 7], but the characterization of defects

✉ Sen Wang
wangsen0401@126.com

Liyang Zhu
zly18313755987@126.com

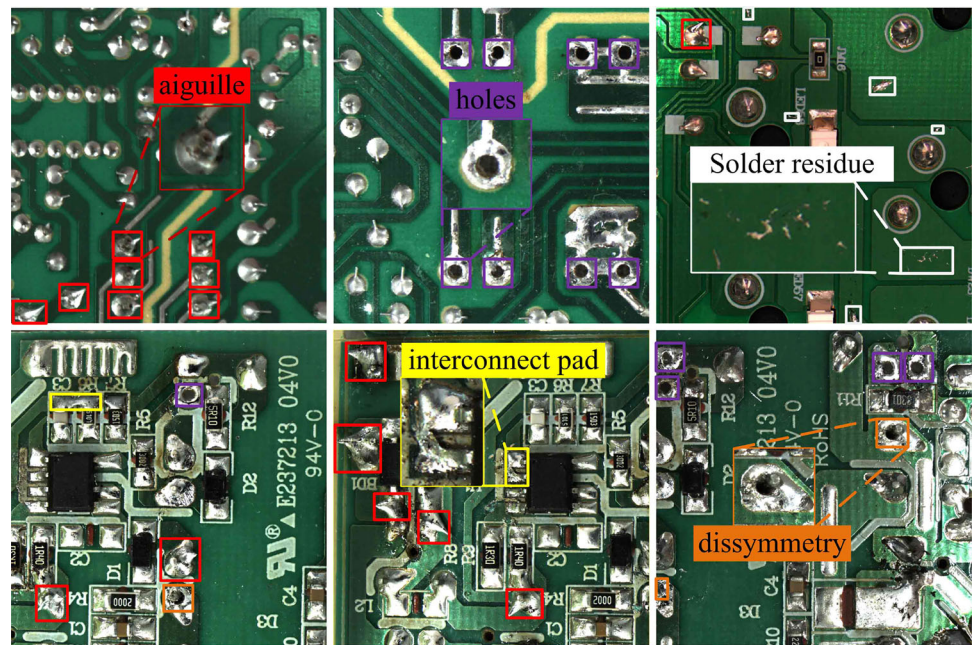
Mingfang Chen
mfchen111@sina.com

Aiping Shen
18107183239@163.com

Xuangang Li
18313513662@163.com

¹ Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming 650500, Yunnan Province, China

Fig. 1 Common soldering defects on the surface of PCBs



is cumbersome. The emergence of Convolutional Neural Networks (CNNs) [8, 9] has provided new options for detecting PCB surface defects. Related target detection algorithms have also attempted to recognize PCB surface defects. For instance, RAR-SSD [1] can acquire all the characteristics of defects by applying multi-scale feature fusion, relying on lightweight Receiving Field Block Modules (RFB-s) and an attention mechanism to highlight the importance of different features, which brings the problem of limited detection accuracy while achieving lightweight defect detection. Feature fusion approaches, including TD-Net [10], TDD-Net [11], improved Faster-RCNN [12], LDD-Net [13], and Edge Multiscale Reverse Attention Network (EMRA-Net) [14], emphasize the employment of multiscale feature fusion to improve the detection accuracy of the algorithms at the cost of huge memory consumption. Specifically, the five feature inputs utilized for feature fusion in LDD-Net and the two-stage approaches such as TDD-Net inevitably reduce the inference speed of the algorithm. These methods especially suffer from a lack of effective interaction with the in-scale features of defects, which is important for defect detection in complex contexts. Although the Focal Loss introduced by the extended feature pyramid model [15] addresses the imbalance between foreground and background, it ignores the imbalance between defect classes.

In conclusion, many deep learning algorithms are not suitable for application in real industrial sites due to the large number of parameters and slow inference speed. Furthermore, the majority of feature fusion approaches fail to acknowledge the significance of intra-scale feature interactions in the context of defects. However, the global information of the in-scale features, in conjunction with long-

range dependencies, is more effective in assisting the network in the learning of PCB surface defect features in a complex context. Existing methods often overlook the issue of severe class imbalance in PCB surface defect detection, particularly in industrial long-tailed data. Long-tailed data reduces the detection accuracy of some defect categories that are at the tail of the data, or worse, a large number of tailed categories is missed. Networks rarely employ the method of auxiliary head supervision in the process of learning PCB surface defects. Nevertheless, the method of auxiliary head supervision allows more accurate defect feature information to be provided to the algorithm, which helps the algorithm to better identify PCB surface defects.

The main contributions to the problems in PCB table defect detection are as follows:

(1) The proposed EFF-Net enables intra-scale feature interaction and cross-scale feature fusion of defects, enabling the network to efficiently fuse long-range dependencies of PCB surface defect features in combination with multi-scale feature information.

(2) The design of an auxiliary head supervision strategy for the supervision of the middle layer network, which consequently assists the network in achieving accurate learning of PCB defect information.

(3) The designed BCE-LRM loss is utilized for mining hard samples to achieve improved detection accuracy of tail data in the defect data.

The paper is structured as follows: Section “[Related work](#)” reviews the state-of-the-art methods for target defect detection in recent years, Section “[Proposed approaches](#)” describes our overall approach in detail, Section “[Experiment](#)” experimentally validates the feasibility of the proposed

method, and Section “[Conclusion](#)” summarizes the work in this paper, highlighting our improvements and future research directions.

Related work

As a practical application of the computer vision field in engineering, the main task of defect detection is to localize and classify defects in industrial products. For example, pixel segmentation [16] is used to identify rail surface defects, YOLO-attention [17] detects defects in wire arc additive manufacturing, improved YOLOV4 [18] identifies surface defects in aluminum strips, improved YOLOV5s [19, 20] identifies small defects on the surface of ceramic tiles, and SR-ResNetYOLO [21] detects defects on the surface of gears. All these methods use multi-scale feature fusion to improve the detection accuracy of the algorithms. The resulting large number of parameters as well as the memory footprint pose difficulties for practical deployment. More to the point, these methods only focus on the inter-layer relationship of defect features, ignoring the intra-scale feature information capture of defects. For instance, SOD-YOLO [22] did not try in a complex context when detecting small target defects in wind turbine blades, and generative adversarial networks [23] have deployment difficulties in identifying surface defects in wood. PEI-YOLOv5 [24] did not address the issues of intra-scale feature interaction and inter-class sample imbalance when detecting fabric defects. The application of a cumbersome multi-scale feature fusion module for the detection of steel surfaces [25] increased training time.

In addition to traditional CNN networks, Transformers are beginning to be widely used in the field of defect detection. For example, DAT-Net [26] detects tool wear defects without considering the inter-class imbalance of defects, and DefectTR [27] detects defects in sewage pipe networks with poor detection accuracy. For the detection of roller surface defects, the multi-layer Transformer encoder used by the Cas-VSwin transformer [28] results in huge computational as well as parametric quantities. DefT [29] suffered from slow inference when applied to the detection of industrial surface defects. LSwIn Transformer [30] did not implement intra-scale feature interaction and cross-scale feature fusion for the detection of steel surface defects. Swin-MFInet [31] did not make full use of multiscale feature information when performing the detection of surface defects on manufactured materials. RDTor [32] performed the detection of PCB surface defects where the intra-scale interaction of low-level features is unnecessary because of the risk of duplication and confusion with high-level feature interactions. At the same time, all of the above approaches suffer from difficulties in practical deployment, and the Transformer framework is usually accompanied by a huge amount of computation, which

makes it difficult to adapt to the high real-time as well as high embeddedness requirements of the industry. Nevertheless, the Transformer framework is able to effectively encode global information and efficiently learn the contextual information of PCB surface defect features. Consequently, it helps the algorithm to recognize PCB surface defects with variable shapes and complex backgrounds.

PCB surface defect detection, as an important branch in industrial defect detection, provides many inspection methods with exploratory significance and practice. Such as improving the CIoU and feature pyramid based on YOLOV5 [33], a combination of lightweight YOLOX and positional attention mechanism [34], adding a trunk feature layer in YOLOV3 [35], YOLOV4-Tiny [36], YOLOV5 combined with Transformer [37] and so on. These methods do not consider the issue of inter-category imbalance of PCB surface defects. As a result, it can be confusing to identify the tail data categories in long-tailed data. The method of adding a backbone feature layer in YOLOV3 increases the memory consumption and computational cost to a large extent, and ignores the intra-layer representation of features, while the long-range dependency of intra-layer representation and global information is extremely important for PCB surface defects detection in complex backgrounds. While few existing defect detection methods utilize auxiliary supervision strategies, auxiliary supervision is extremely beneficial in the training of lightweight networks. Auxiliary supervision is typically effective in providing more comprehensive and reliable abstracted semantic feature information for the detection of PCB surface defects. It also helps the algorithm to more accurately exclude large redundant features, thus achieving the purpose of feature purification.

Proposed approaches

The LLM-Net network structure we constructed in Fig. 2 mainly consists of a backbone feature extraction network, an efficient feature fusion network, and a detector. Inspired by YOLOV5 and DETR [38]. We propose an Efficient Feature Fusion Network (EFFNet) by combining the encoder simplification of DETR with the PAFPN network. Multi-scale features are converted into image features utilizing intra-scale feature interaction and cross-scale feature fusion. Enables LLM-Net to efficiently fuse long-range dependencies of PCB surface defect features with multi-scale feature information. The proposed BCE-LRM results in better learning of hard samples by LLM-Net. The errors of LLM-Net during the training process are reduced by our designed assisted supervision strategy, which allows the network training time to be reduced. At the same time, it gives LLM-Net a stronger ability to characterize PCB surface defects.

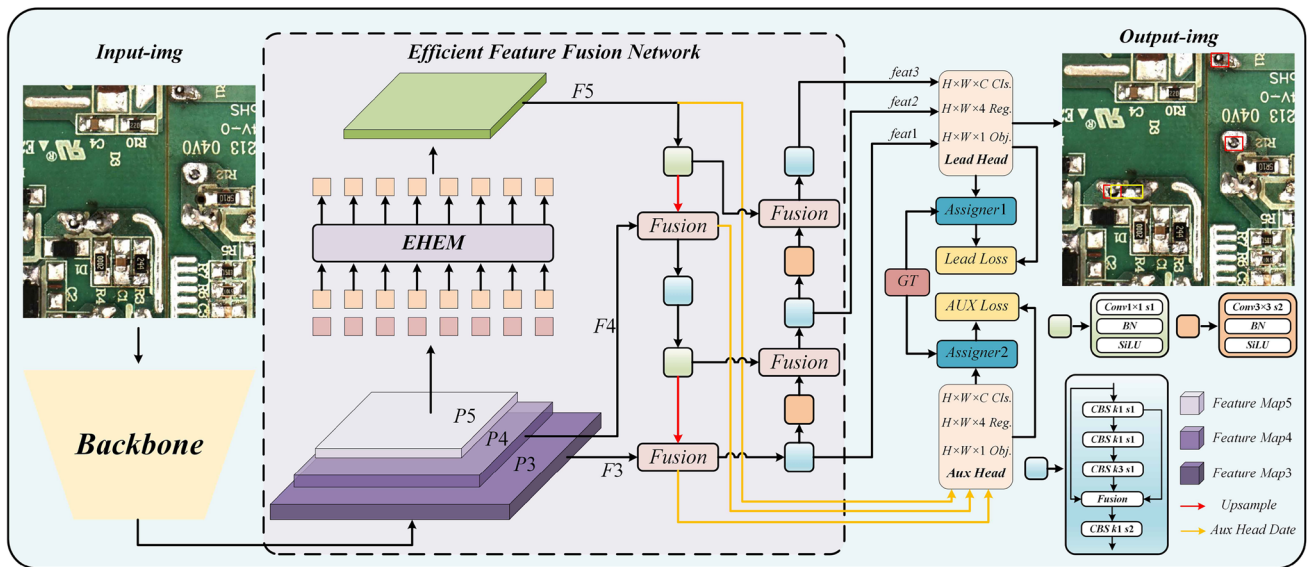


Fig. 2 Architecture of LLM-Net, where, Efficient Hybrid Encoding Module (EHEM) is utilized to interact intra-scale features of PCB surface defects in an attentional manner. The Efficient Feature Fusion Network ultimately achieves intra-scale feature interaction and cross-scale feature fusion

An efficient feature fusion network

The feature fusion network proposed consists mainly of an efficient hybrid encoding module (EHEM) and a multi-scale feature fusion network. In particular, the EHEM module is mainly responsible for interacting with the in-scale features of PCB surface defects in an attentional manner to achieve efficient embedding of global information about defects. Multi-scale feature fusion network transforms the defect feature maps into feature layers for predicting defects of different sizes, enabling the network to adapt to PCB surface defects of different shapes and sizes.

Efficient hybrid encoding module

The deeper semantic features of PCB surface defects often contain richer abstracted semantic information, and we encode the deeper semantic features provided by the backbone feature network through an efficient hybrid encoding module, which effectively exploits the semantic features of defects. The abstracted feature information in such semantic information is more beneficial for defect classification for regression. To achieve efficient and accurate identification of PCB surface defects, we simplified the encoder of DETR by adopting a simple single-layer transform coding module. The hybrid coding module constructed is shown in Fig. 5.

The efficient hybrid encoding module, with the scaled dot product attention in Fig. 3 as its core, can improve the encoding of the global information and learn the contextual information of the PCB surface defect features to a limited extent, which in turn assists the algorithm in detect-

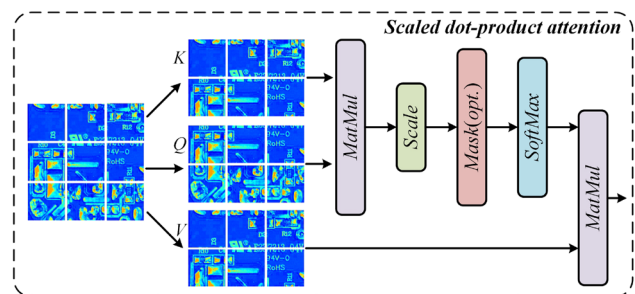


Fig. 3 Scaled dot-product attention

ing PCB surface defects with variable shapes and complex backgrounds. The attention input consists of the query, key (of dimension K), and value (of dimension V), which are described in Eq. (1).

$$Att(K, Q, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

where the three matrices K (key), Q (query), and V (value) are obtained by linear transformation, and d_k gives the length of K and Q . The Q matrix of the element is then dot-multiplied by the K matrix of the other elements in the sequence, establishing the dependence relationship between one element and the others in the sequence. Following the encoding of Eq. (1), a relationship is established between the defective features on the PCB surface in terms of order and spatial location. As a consequence, the ability of the algorithm to extract the positional features of the defects is

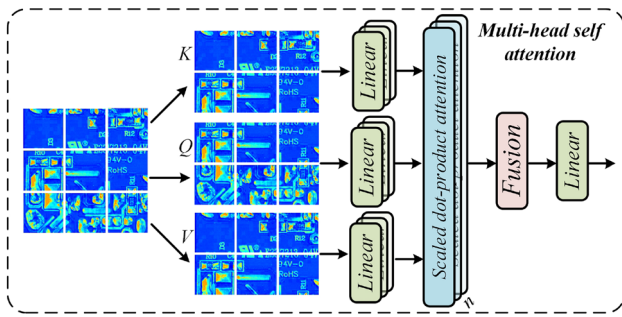


Fig. 4 Multi-head self attention

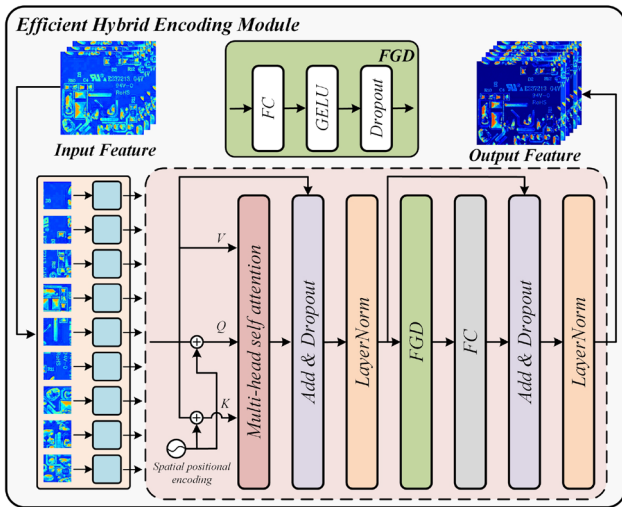


Fig. 5 Efficient hybrid encoding module

improved, and the ability of the algorithm to locate the defects more accurately is enhanced.

The multi-head attention mechanism is obtained by splicing the output of multiple parallel scaled dot product attention, as shown in Fig. 4. The multi-head attention module achieves compensation between Q , K , and V through parallel processing, resulting in n sets of results with the same number of heads after n (number of heads) linear transformations. The results provide sequence information on defects and dependency information between elements. Not only are they extremely helpful for the network to obtain contextual information on defects on the PCB surface, but they are also indispensable for the subsequent classification and prediction of defects.

The efficient hybrid encoding module in Fig. 5 further reduces computational redundancy based on the transform encoder, which only performs in-scale interactions of features on $P5$. We argue that applying the self-attention operation to high-level features with richer semantic concepts captures the connections between conceptual entities in PCB defective images, which helps the subsequent modules detect and recognize defects in images. Simultaneously, due to the

absence of semantic concepts, interactions between low-level features within the scale are unnecessary. Otherwise, there is a risk of duplication and confusion with interactions between high-level features.

Efficient feature fusion network

The backbone network outputs several feature layers. The $P3$ layer contains fine-grained features and location information for recognizing subtle defects. The $P4$ layer contains more abstract semantic features suitable for recognizing medium-sized defects. The richest semantic features are found in the $P5$ layer, which plays a crucial role in classifying and recognizing defects. An efficient feature fusion network is constructed by combining the proposed EHEM with a multi-scale feature fusion network. The EFF-Net processes the deep semantic feature $P5$ with multi-head attention and obtains the PCB surface defect feature $F5$, which contains the long-range dependencies and contextual information. Subsequently, the $F5$ feature is utilized for cross-scalar feature fusion, and at the same time, the information contained in $F5$ is transported to the other two scales of features. Ultimately, EFF-Net accomplishes the embedding of multi-scale defect context information and long-range dependencies, which in turn effectively supports the detector in detecting and identifying defects in the image. Equation (2) characterizes our efficient feature fusion network.

$$\begin{aligned}
 Q = K = V &= Flatten(P5) \\
 F5 &= Reshape(Att(K, Q, V)) \\
 \{feat1, feat2, feat3\} &= EFF(\{F3, F4, F5\}) \quad (2)
 \end{aligned}$$

where Att represents the multi-head attention module, $\{feat1, feat2, feat3\}$ are the outputs of the efficient feature fusion network, $Reshape$ and $Flatten$ are inverse processes to each other, and $P5$ is the high-level feature of the backbone outputs. $F3$ and $F4$ are the shallow and middle outputs of the backbone network (equal to $P3$ and $P4$, respectively, in Fig. 2). The efficient feature fusion network is characterized by EFF .

Assisted supervision strategy

Deep supervision [39] is frequently employed as a means in deep network training, and auxiliary heads are usually added to the middle layer of the network to obtain the auxiliary supervision loss. To enhance the ability of the algorithm to obtain more accurate feature information on PCB surface defects, we designed the auxiliary supervision strategy. The auxiliary head label assignment strategy is independent, preventing the auxiliary head loss from affecting the main detection head loss. Meanwhile, in the label assignment strat-

Algorithm 1: Pseudo-code of Auxhead label assignment strategy

```

Input: Prediction results  $P$ ; Ground truth  $Targets$ .
Output: Index information  $indices$ ; Anchor box information  $anch$ .
1 Initialization:  $na \leftarrow 3, nt \leftarrow$  Number of targets,  $indices \leftarrow$ 
  list[],  $anch \leftarrow$  list[],  $anchor\_t \leftarrow 4.0, g \leftarrow 1.0$ 
   $gain \leftarrow Tensor ([7,1]), a_i \leftarrow matrix [na,nt]$ 
   $Targets \leftarrow Concat [(a_i, Targets), dim = 2]$ 
   $off \leftarrow Tensor ([[0, 0], [1, 0], [0, 1], [-1, 0], [0, -1],]) * g$ 
2 for  $i : num\_anchors$  do
3    $anchors \leftarrow anchors [i]$ 
   Update the elements of  $gain$ 
    $t \leftarrow Targets * 1.0$ 
   if  $nt > 0$  then
4      $r \leftarrow t (h, w) / anchors (h, w)$ 
      $j \leftarrow max (r, 1/r) < anchor\_t$ 
      $t \leftarrow t [j]$ 
      $g_{xy} \leftarrow t_{center} (x, y)$ 
      $g_{xi} \leftarrow gain_{center} (x, y) - g_{xy}$ 
      $j, k \leftarrow ((g_{xy} // 1.0 < g) \cap (g_{xy} > 1.0)). T$ 
      $l, m \leftarrow ((g_{xi} // 1.0 < g) \cap (g_{xi} > 1.0)). T$ 
      $offsets \leftarrow g_{xy} + off [j]$ 
5   else
6      $t \leftarrow Targets [0]$ 
      $offsets \leftarrow 0$ 
7    $b, c \leftarrow t (x_{min}, y_{min}), t (x_{max}, y_{max}), g_{xy} \leftarrow t_{center} (x, y)$ 
      $ghw \leftarrow t (h, w), g_{ij} \leftarrow (g_{xy} - offsets), a \leftarrow t_{classes}$ 
     Index information is added to the  $indices$  list.
     Anchor frame information added to  $anch$  list.
   end
8 return  $anch, indices$ 

```

egy of the auxiliary head, we adopt the state-of-the-art soft label assignment strategy. Efficiently avoids increasing the error in the network learning process when using the original hard labels, which in turn effectively helps the algorithm to learn the defective features. The loss function employed in the auxiliary head is the same as that of the main detection head, which prevents the auxiliary head from ignoring hard samples and also promotes the algorithm to converge faster during the training process. Algorithm 1 presents the pseudo-code for the core code of the label assignment strategy in the auxiliary head.

The application of auxiliary heads greatly enhances lightweight algorithms for learning from PCB surface defects with variable morphology on long-tail datasets. Following the fully supervised training of the mid-layer network, we incorporate the supervised loss obtained from the main detection head and perform backpropagation and gradient updates. The designed auxiliary supervision strategy combines hard labeling with soft labeling to achieve a more effective defect target allocation strategy. Additionally, the auxiliary supervision can quickly correct learning errors during the training process for the lightweight defect detection network, result-

ing in more effective identification of PCB surface defects by the lightweight algorithm. The auxiliary supervision strategy is solely utilized during the training process and is not involved in the predictive inference process. With this strategy, the efficiency of the algorithm is ensured.

Boosting for hard samples

In industrial defect detection scenarios, severe imbalances between categories seriously affect detector effectiveness in detecting tail data. To effectively improve the detector performance, we propose a hard sample mining method BCE-LRM, which combines the binary cross-entropy loss. To enhance detector performance effectively, we are inspired by the Literature [40, 41] and propose a difficult sample mining method BCE-LRM by combining the binary cross-entropy loss. Unlike the Literature [40], the original LRM is only for single-scale features, while in the proposed method, we utilize the BCE-LRM for all scales of features. Figure 6 presents the proposed BCE-LRM strategy. Firstly, the loss of different scale features is calculated for each mini-batch. Then, the losses are ranked and stored in a vector. Next, the loss values in the vector are ranked in descending order, and the top β of the ranking is selected for each image. Finally, the

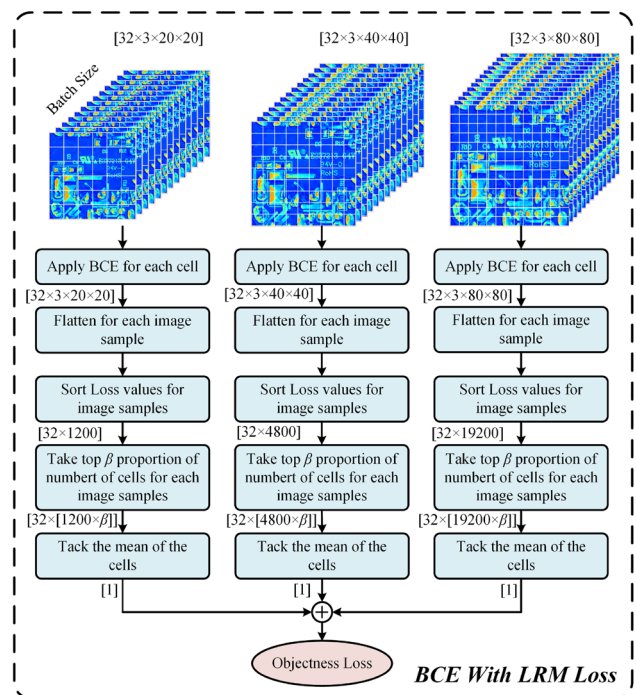


Fig. 6 BCE-LRM. First, the loss of different scale features is calculated for each mini-batch. Then, the losses are ranked and stored in a vector. Next, the loss values in the vector are ranked in descending order, and the top β of the ranking is selected for each image. Finally, the averaged loss is obtained and utilized as the confidence loss for network prediction

averaged loss is obtained and utilized as the confidence loss for network prediction.

The algorithm prioritizes difficult samples with reduced learning effect by the above loss ordering and loss selection, enabling it to effectively learn the difficult samples in the PCB surface defects. The sorting and selection of losses, in comparison to the original binary cross-entropy loss, is effective in learning both bad and difficult samples. In contrast to the Literature [41], we opted to avoid the application of the focal loss in avoiding its neglect of easily detectable faulty categories. Additionally, considering that there are a few quality PCB surface defect samples that have high intersection and concurrency ratios but not high confidence levels, we have avoided utilizing Varifocal Loss.

LRM is utilized for binary cross-entropy losses after feature mapping is complete, and only the losses are ranked and selected in the training phase. Equation (3) characterizes the binary cross-entropy loss.

$$\sigma(P_i) = \frac{1}{1 + e^{P_i}}$$

$$BCEWithLogitsLoss = \frac{1}{N} \sum_{i=1}^N (y_i \cdot \log(\sigma(P_i)) + (1 - y_i) \cdot \log(1 - \sigma(P_i))) \tag{3}$$

where variable p_i represents the probability of a defective sample being classified as a positive example, while y_i represents the true label of the defective sample, and N represents the number of samples.

Equation (4) shows that the core of LRM is a binary mask matrix called *Mask*. The defective feature mapping, *Feat*, obtained from prediction is multiplied with *Mask* to obtain a new feature mapping by multiplying the elements. This new defective feature mapping is mainly utilized to determine whether the sample is a difficult sample or not.

$$Feat_{out} = Feat \cdot Mask \tag{4}$$

The binary parameters in *Mask* are not pre-set and are determined by the final prediction result. When the elements in the feature mapping belong to difficult samples, the elements in the mask *Mask* are set to 1 and the other elements are set to 0. The backpropagation of the network is then carried out, and the parameter updating is shown in Eq. (5).

$$\frac{\partial \hat{Loss}}{\partial f_{i,j}^c} = m_{i,j}^c \cdot \frac{\partial \hat{Loss}}{\partial f_{i,j}^c} = \begin{cases} 0 & m_{i,j}^c = 0 \\ \frac{\partial \hat{Loss}}{\partial f_{i,j}^c} & m_{i,j}^c = 1 \end{cases} \tag{5}$$

where denotes the element in the faulty feature mapping f (corresponding to *Feat*) with position (i, j) when the channel is c , and denotes the element in the mask *Mask* with

position (i, j) when the channel is c . With the mask *Mask*, the algorithm successfully reduces the backpropagation of simple samples and ensures that the algorithm learns difficult faulty samples.

Experiment

Dataset

Figure 7 shows the PCB surface defect acquisition system that was built. The system comprises a strip light source, an industrial camera, a conveyor belt, and a host computer. The acquisition resolution of the industrial camera was set to 3072×2048, and 300 PCB images were obtained, each containing five types of common soldering defects: aiguille, interconnect pad, dissymmetry, holes, and solder residue. To prevent model overfitting during training, we captured screenshots of the dataset using a fixed window size of 640×640 pixels. The step size of the screenshots was set to 320-pixel units, resulting in 940 defect images. Table 1 shows the statistics of the various types of defects in the dataset, and Table 2 displays the camera parameters at the time of acquisition. To ensure a more rigorous verification of the proposed method, we set up the dataset in MS-COCO format.

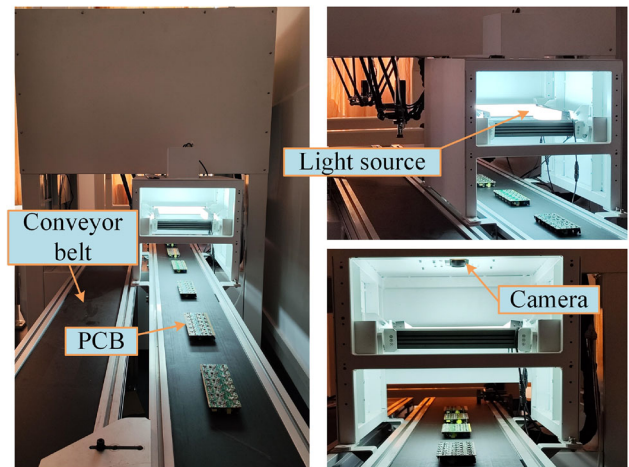


Fig. 7 Diagram of the capture device

Table 1 The number of defects in various categories in the dataset

| Defects classes | Number of defects |
|------------------|-------------------|
| Aiguille | 1697 |
| Dissymmetry | 507 |
| Holes | 1160 |
| Interconnect pad | 1228 |
| Solder residue | 705 |

Table 2 The camera parameters during data collection

| Parameter | Value |
|-------------------|--------------------|
| Camera Model | MV-CS060-10UC-PRO |
| Aperture | F4 |
| Focal Length | 8 mm |
| Light Source Type | Ring Light |
| Exposure Time | 50,000 μ s |
| Camera Resolution | 3072 \times 2048 |
| Image Format | .bmp |
| Pixel Type | RGB 8 |

Table 3 Hyperparameter setting

| Parameter | Value |
|-----------------------|------------------|
| Batch Size | 16 |
| Epoch | 300 |
| Optimizer | SGD |
| Initial learning rate | 0.01 |
| Momentum | 0.937 |
| Data augmentation | Mosaic |
| Input size | 640 \times 640 |

Experimental preparation and parameter settings

The experimental models were executed on an Ubuntu 23.10 device equipped with torch2.0. The device had a 12th Gen Intel(R) Core(TM) i7-12700F 2.10 GHz CPU, 32Gb RAM, and an Nvidia GeForce RTX 4080 16Gb GPU. All models were trained without using pre-training weights or freezing the backbone training, ensuring fairness in the experiments. The hyperparameter settings during training are displayed in Table 3. To ensure data uniformity in the comparison, we maintained the same image size for training, validation, and testing. Additionally, all models were trained, tested, and validated on an equal number of images.

Evaluation metrics

Practical industrial field inspection aims to balance the number of model parameters, inference speed, and detection accuracy. Objective evaluation metrics such as Precision, Recall, mAP@0.5, mAP@0.5:0.95, FPS, GFLOPs, and Parameters are utilized to evaluate the performance of LLM-Net, and the calculation methods of these evaluation metrics are as follows.

Precision is the ratio of correctly predicted positive samples by LLM-Net to the total number of positive samples predicted. The precision of the predictions made by LLM-Net is characterized by the following formula. The calculation formula for precision is as follows.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

TP represents the number of correctly predicted positive samples, while FP represents the number of negative samples that were incorrectly predicted as positive.

Recall is the ratio of correctly predicted positive samples to the total number of actual positive samples in the LLM-Net prediction samples. It characterizes the ability of LLM-Net to identify all positive samples. The calculation formula for the recall is as follows.

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

FN denotes a positive sample that is incorrectly predicted as a negative sample.

The mean average precision (mAP) represents the average AP value for all categories. The AP value is the area under the Precision-Recall curve, and a larger value indicates better detection for a category.

$$AP = \int_0^1 P(R) dR \quad (8)$$

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

Ablation experiment

Validation of the effectiveness of the proposed method on baseline models

The purpose of the ablation experiments is to confirm the effectiveness of the proposed auxiliary supervision strategy, EFFNet, and BCE-LRM for PCB surface defect detection. A series of experiments were conducted on a homemade dataset, with a focus on the metric mAP@0.5 in the experimental metrics. In the field of PCB surface defect detection, accuracy is prioritized over speed when the frame rate meets the field requirements. Therefore, FPS, GFLOPs, and Parameters are considered secondary.

We apply the YOLOV5-n model as a baseline to verify the effects of the auxiliary supervision strategy, EFFNet, and the algorithm after adopting the BCE-LRM on the PCB surface defect detection accuracy (mAP@0.5, mAP@0.5:0.95), FPS, GFLOPs, and Parameter, and the statistical results are shown in Table 4. The auxiliary supervision strategy can effectively reduce the error weights during the training process, so that the algorithm focuses more on learning the positive sample features of the PCB surface defects, which leads to an improvement of 0.3% and 5.5% in the mAP@0.5 and mAP@0.5:0.95 of the baseline model, respectively. With the mixture of the auxiliary supervision strategy and EFFNet, the algorithm effectively learns the refined feature informa-

Table 4 Results of ablation study

| NO. | Method | mAP@0.5 (%) | mAP@0.5:0.95 (%) | Params (Mb) | FPS | GFLOPs |
|-----|------------|-------------|------------------|-------------|-----|--------|
| A | YOLOv5-n | 97.1 | 65.7 | 1.68 | 256 | 4.1 |
| B | A+AUX_head | 97.4 | 71.2 | 2.87 | 169 | 6.8 |
| C | B+EFFNet | 97.4 | 71.8 | 3.96 | 166 | 7.5 |
| D | B+BCE-LRM | 97.9 | 71.1 | 2.87 | 166 | 6.8 |
| E | D+EFFNet | 98.5 | 71.9 | 3.96 | 188 | 7.5 |

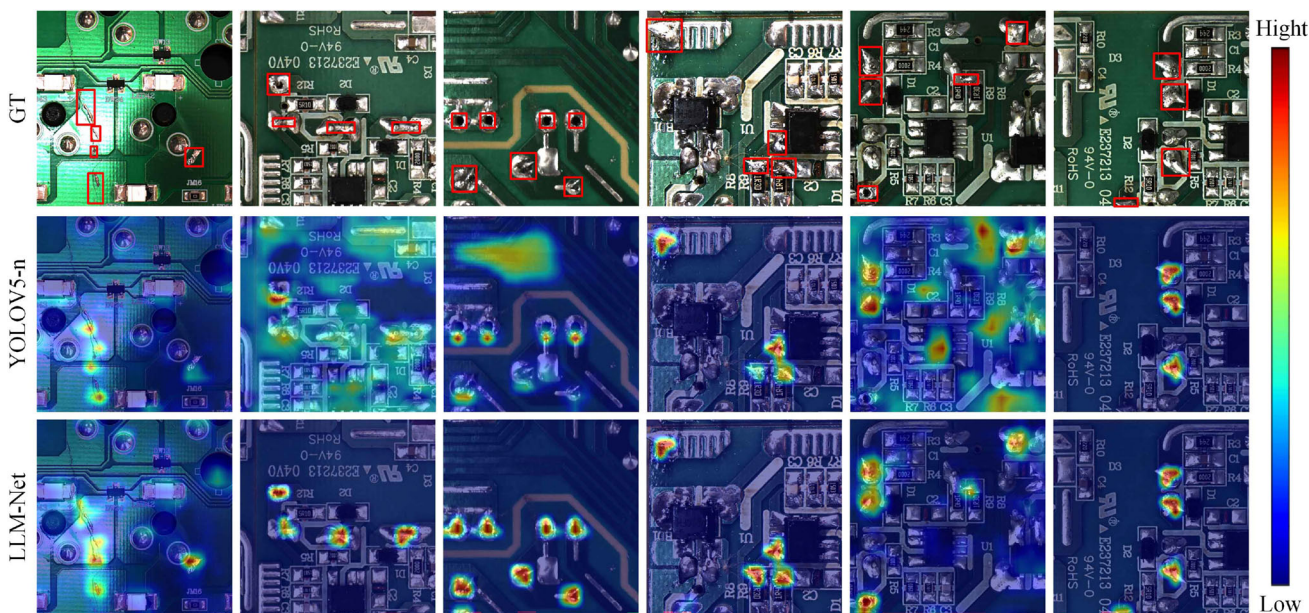


Fig. 8 Visualization results of heatmaps for the baseline model and LLM-Net, where a more reddish color indicates a higher probability of recognition as a target, and conversely a more bluish color indicates a higher probability of recognition as a background

Table 5 The impact of different loss functions on the model accuracy

| Losses | mAP@0.5 (%) | mAP@0.5:0.95 (%) | Params (Mb) | FPS | GFLOPs |
|----------------|-------------|------------------|-------------|-----|--------|
| BCE | 97.3 | 71.8 | 3.96 | 166 | 7.5 |
| Focal Loss | 94.1 | 68.0 | 3.96 | 188 | 7.5 |
| Varifocal Loss | 94.7 | 68.8 | 3.96 | 163 | 7.5 |
| Slide Loss | 97.3 | 71.3 | 3.96 | 181 | 7.5 |
| BCE-LRM (ours) | 98.5 | 71.9 | 3.96 | 188 | 7.5 |

tion of the defects as well as richer contextual information, and the mAP@0.5 and mAP@0.5:0.95 of the algorithm are improved by 0.3% and 6.1%, respectively, compared to the baseline model. Finally, the employment of BCE-LRM improved the ability of the algorithm to learn from hard samples, with mAP@0.5 and mAP@0.5:0.95 improving by 1.4% and 6.2%, respectively, compared to the baseline model. The algorithm that achieved the best accuracy was designated LLM-Net, and the visualized heat map is shown in Fig. 8. The closer the colors in the graph are to blue, the higher the probability that this part is considered to be the background, while the closer the colors are to red, the higher the probability that this part is considered to be the target. From the figure,

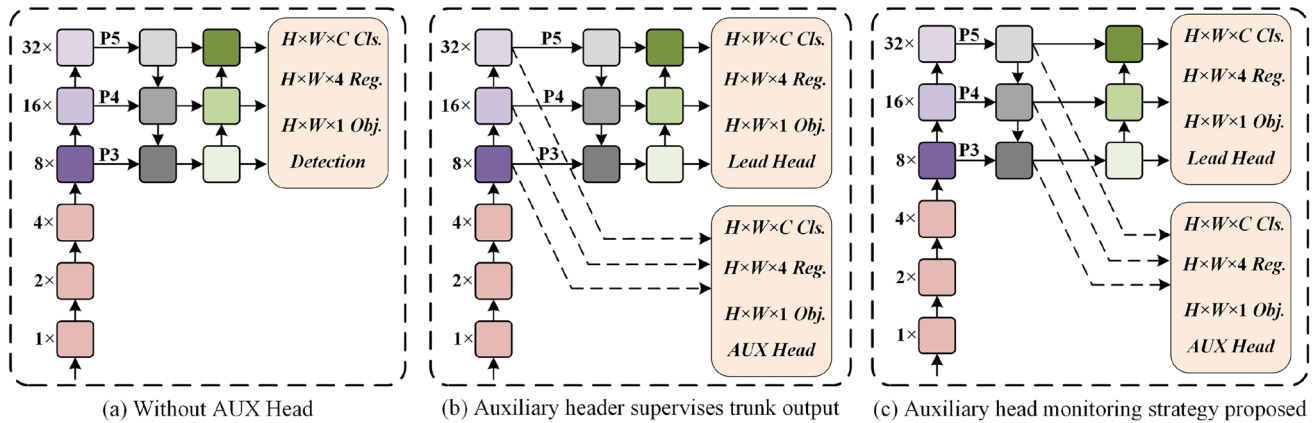
we can see that our LLM-Net has a strong defect perception capability and accurately identifies the location of defects.

Comparison of different loss functions

For the solution of difficult samples and sample imbalance problems, there are also similar methods such as Focal Loss, Varifocal Loss, Slide Loss, etc. as listed in Table 5. For the difficult sample problem in PCB surface defects, Focal Loss suppresses the prediction frames with high positional accuracy and low confidence and ignores the effect of the intersection ratio, which degrades the performance of the model. Thus, this caused the network accuracy metrics

Table 6 Effect of auxiliary head supervision position on model performance

| Method | mAP@0.5 (%) | mAP@0.5:0.95 (%) | Params (Mb) | FPS | GFLOPs |
|--------|-------------|------------------|-------------|-----|--------|
| (a) | 97.3 | 69.7 | 2.78 | 117 | 4.9 |
| (b) | 97.1 | 69.7 | 3.97 | 181 | 7.6 |
| (c) | 98.5 | 71.9 | 3.96 | 188 | 7.5 |

**Fig. 9** Various auxiliary head supervision schemes, where (a–c) correspond to the ‘Method’ column in Table 6

mAP@0.5 and mAP@0.5:0.95 to degrade to 94.1% cent and 68.0%, respectively. Varifocal Loss spends most of its effort on the imbalance between the foreground and background when performing loss compensation, and the imbalance between the samples affects the detection effectiveness of the model. Eventually, the network accuracy metrics mAP@0.5 and mAP@0.5:0.95 are degraded to 94.7% and 68.8%, respectively. Slide Loss takes the average of the intersection and concurrency ratios of the GT and prediction frames to determine whether the target belongs to the difficult samples or not, and emphasizes the boundary samples by weighting them, but the method does not improve the network performance much. Slide Loss merely managed to equal BCE in the mAP@0.5 metric, but caused mAP@0.5:0.95 to degrade to 71.2%. Usually, the hard samples have a large chance of being missed, which will cause difficulties for the subsequent weighting of the slide loss. BCE-LRM ranks the loss of samples and achieves the learning of difficult samples more effectively, which makes the algorithm achieve the best accuracy of 98.5% for the mAP@0.5 metric. At the same time, BCE-LRM is more friendly to most of the target detectors and has a strong embedding ability.

Optimization of auxiliary supervision strategies

It was found that the auxiliary supervision strategy is effective in enhancing the performance of the algorithm. Three experiments were conducted to determine the best approach for this strategy, testing only the auxiliary head on LLM-Net. The results of these experiments are presented in Table 6, with

the ‘Method’ column corresponding to the scheme number in Fig. 9. The performance of the network without the auxiliary supervision strategy shows only a small improvement compared to the baseline model. However, when the auxiliary head supervises the output of the backbone, it learns the error information of the shallow network. Consequently, the auxiliary head affects the prediction of the main detector head, resulting in no effective improvement in the performance of the network. The proposed scheme in this paper implements auxiliary supervision in the middle layer of the network to acquire more defective feature information, resulting in an improved accuracy index mAP@0.5 of 98.5%. Additionally, the combination of soft and hard labels enhances the adaptability of the network to PCB surface defects.

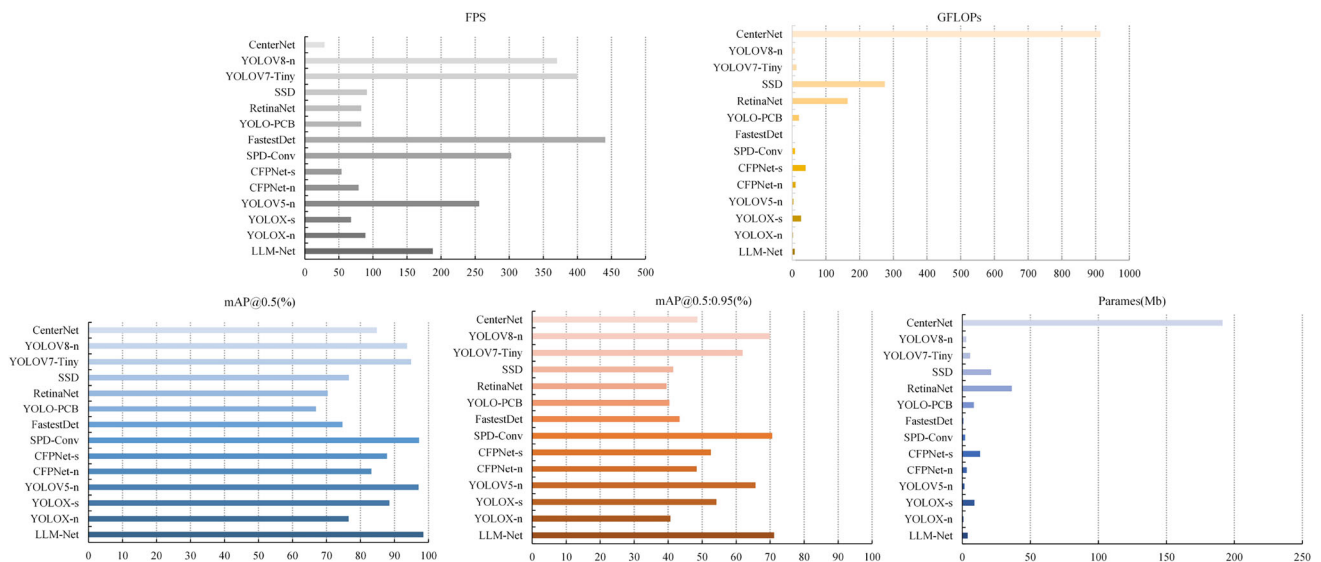
Comparison with other algorithms

Quantitative comparison with previous methods

In real industrial scenarios, PCB surface defect detection not only requires high accuracy and reasonable inference speed but also needs to achieve strong quantization of the model. Therefore, we compare the accuracy metrics (mAP@0.5, mAP@0.5:0.95), FPS, GFLOPs, and Parameter of the network, focusing on the accuracy metrics of the algorithm and treating other metrics as secondary metrics. In Table 7, we comprehensively compare LLM-Net with 13 representative current deep learning algorithms (SSD [42], CenterNet [43], FastestDet [44], RetinaNet [45], YOLO-PCB [33], SPD-Conv [46], YOLOV5-n [47], YOLOV7-Tiny [48],

Table 7 Results compared to other methods

| Module | mAP@0.5 (%) | mAP@0.5:0.95 (%) | Params (Mb) | FPS | GFLOPs |
|------------------|-------------|------------------|-------------|-----|--------|
| SSD [42] | 76.7 | 41.5 | 21.14 | 91 | 274.9 |
| CenterNet [43] | 84.8 | 48.6 | 191.25 | 29 | 914.3 |
| RetinaNet [45] | 70.4 | 39.5 | 36.41 | 83 | 164.7 |
| FastestDet [44] | 74.7 | 43.4 | 0.90 | 441 | 0.8 |
| YOLO-PCB [33] | 66.9 | 40.4 | 8.50 | 83 | 20.1 |
| SPD-Conv [46] | 97.2 | 70.6 | 2.05 | 303 | 8.5 |
| YOLOV5-n [47] | 97.1 | 65.7 | 1.68 | 256 | 4.1 |
| YOLOV7-Tiny [48] | 94.9 | 61.9 | 5.74 | 400 | 13.1 |
| YOLOV8-n [49] | 93.7 | 69.9 | 2.87 | 370 | 8.1 |
| CFPNet-n [50] | 83.2 | 48.4 | 3.30 | 79 | 10.2 |
| CFPNet-s [50] | 87.8 | 52.6 | 13.14 | 54 | 40.0 |
| YOLOX-n [51] | 76.5 | 40.7 | 0.90 | 89 | 2.5 |
| YOLOX-s [51] | 88.5 | 54.2 | 8.94 | 68 | 26.8 |
| LLM-Net | 98.5 | 71.9 | 3.96 | 188 | 7.5 |

**Fig. 10** Comprehensive performance comparison of each algorithm

YOLOV8-n [49], CFPNet-n [50], CFPNet-s [50], YOLOX-n [51], and YOLOX-s [51]) in terms of performance.

Table 7 presents the results of the quantitative comparison between LLM-Net and other methods. The results show that LLM-Net outperforms other methods significantly in terms of the two indicators, mAP@0.5 and mAP@0.5:0.95. Figure 10 demonstrates that LLM-Net is suitable for PCB surface defect detection, with a focus on detection accuracy, despite not achieving the best performance in FPS and Parameter indexes. CenterNet employs the heat map prediction method in PCB surface defect detection, which leads to its inability to detect defects under complex backgrounds well. FastestDet improves detection speed while reducing the number of parameters, at the cost of detection accuracy. YOLO-PCB and YOLOV7-Tiny have the problem of insuf-

ficient learning of tail data in defect identification, which leads to serious leakage. Despite adopting a more reasonable means of downsampling, the feature information of PCB surface defects is still severely lost after downsampling by SPD-Conv. In defect detection, the decoupling head of YOLOV8-n is incapable of accurately identifying the tail data. The explicit visual center pyramid in CFPNet-n and CFPNet-s is not sufficient to extract the feature information of the defects. The huge computational volume makes their actual deployment capability poor. The YOLOX-n and YOLOX-s algorithms fail to consider the intra-scale feature interaction of defective features and the cross-scale feature interaction simultaneously. This results in the loss of global information about defects and long-range dependencies, which in turn leads to suboptimal detection outcomes.

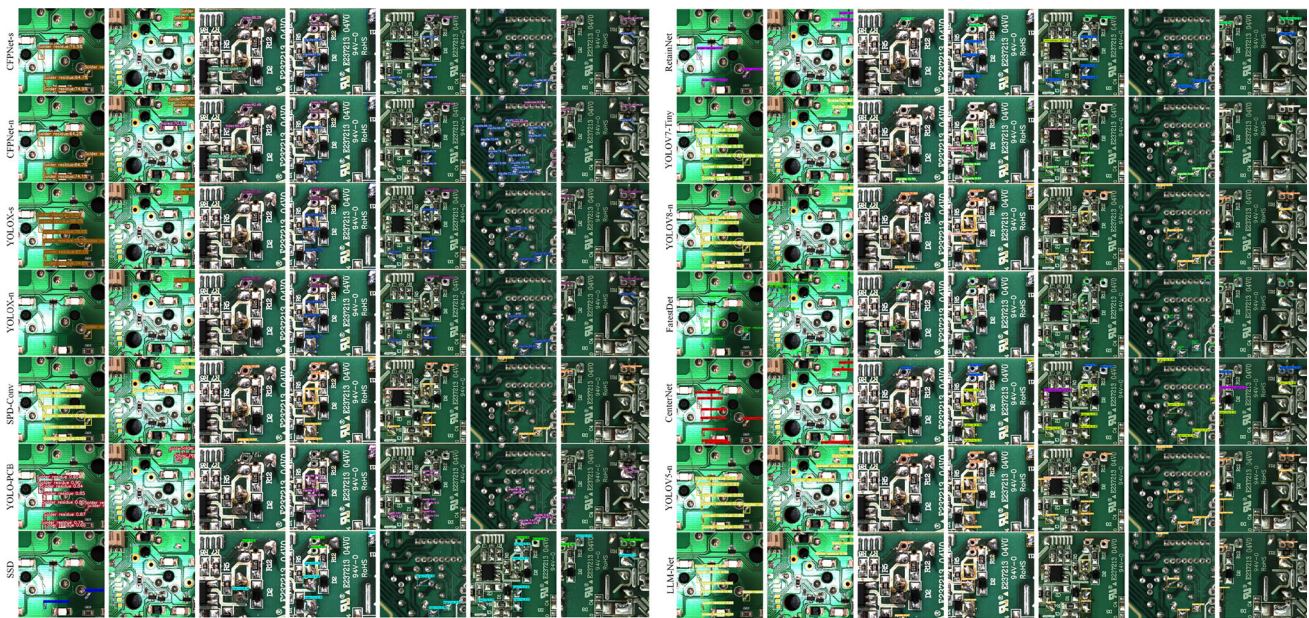


Fig. 11 Visualization of detection results for each algorithm

LLM-Net achieves intra-scale feature interaction and cross-scale feature fusion of PCB surface defects through EFF-Net, which efficiently embeds the long-range dependencies of PCB surface defect features and multi-scale feature information. Applying the designed auxiliary head supervision strategy assists in achieving accurate learning of PCB defect information. Ultimately, LLM-Net achieved the best detection accuracy in several excellent performance algorithms while holding an inference speed of 188FPS. The experimental results show that LLM-Net is more suitable for PCB surface defect identification.

Qualitative comparison with other approaches

Figure 11 shows the visualization of the detection results of various methods. LLM-Net accurately identifies the long-tailed data class (solder residue) and some subtle defects. Intra-scale feature interaction and inter-scale feature fusion provide LLM-Net with richer global information and long-range dependencies. The introduction of multiscale feature linkages can also help LLM-Net achieve more accurate defect recognition in complex contexts. Compared to other comparison algorithms, CenterNet is prone to leakage and misdetection, particularly in identifying small defects such as the solder residue defects in the second column of the visualization result graph. Fastest-Det, despite utilizing a lightweight backbone as well as a feature fusion network, suffers from losing its powerful feature extraction capability, resulting in a large number of missed targets. Excessive attention can decrease the accuracy of the algorithm while introducing false attention features. YOLOV7-Tiny has false

detections when identifying defects. SPD-Conv can identify defects relatively accurately, while still failing to avoid false detections. YOLOV8-n ignores the interaction of features within the scale when extracting features, so it can identify subtle defects with a high degree of accuracy. YOLO-PCB, YOLOV7-Tiny, CFPNet-n, CFPNet-s, YOLOX-n, YOLOX-s, and YOLOX-4 all fail to detect the tail residue, which is closely related to the fact that they do not consider mining the tail data.

Conclusions

The current methods for improving defect recognition accuracy have some limitations. Firstly, they only utilize simple feature fusion to improve defect recognition accuracy, which results in large memory consumption while ignoring the importance of intra-layer feature interaction; Secondly, they neglect the long-tail problem in industrial data; Thirdly, most of the methods ignore the utilization of an auxiliary supervision strategy for PCB surface defect recognition, which can provide accurate defect feature information to the algorithms; Fourthly, they ignore the importance of intra-layer and inter-layer feature interaction to improve defect recognition accuracy. This paper proposes an EFF-Net based on YOLOV5-n to interact with both intra-layer and inter-layer defect features, which achieves the global information of defects as well as the embedding of long-range dependencies. The algorithm is aided by an auxiliary supervision method that utilizes a soft-label assignment strategy to extract

more accurate defect features, and BCE-LRM is designed to improve the detection effect of tail data.

Experiments were conducted to validate LLM-Net using a dataset of PCB surface soldering defects that we collected. The results demonstrate that LLM-Net has the highest detection accuracy and can perform real-time inference at 188 FPS. The visualization results indicate that LLM-Net has the best detection performance and does not present any leakage in randomly selected test images. Currently, we can detect soldering defects on the surface of 5 classes of printed circuit boards in real-time. However, in industrial scenarios, it is crucial to ensure high detection efficiency. To improve the efficiency of the deep learning method in such scenarios, it is essential to enable defect detection based on tracking.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 52065035, and the Yunnan Provincial Department of Science and Technology Basic Research Special Project under Grant 202102AC080002.

Author Contributions Liying Zhu: Conceptualization, Methodology, Software. Sen Wang: Data curation, Writing- Original draft preparation. Mingfang Chen: Software, Validation. Aiping Shen: Visualization, Investigation. Xuangang Li: Supervision, Validation.

Funding This work was supported in part by the National Natural Science Foundation of China under Grant 52065035, and the Yunnan Provincial Department of Science and Technology Basic Research Special Project under Grant 202102AC080002.

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request. Availability of data and materials Available upon request at the corresponding author.

Materials Availability Not applicable.

Code Availability Available upon request at the corresponding author.

Declarations

Conflict of interest The authors have no relevant financial or nonfinancial interests to disclose.

Ethics approval Not applicable.

Consent to participate All authors consent to participate.

Consent for publication All authors consent to participate.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copy-

right holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Jiang W, Li T, Zhang S, Chen W, Yang J (2023) Pcb defects target detection combining multi-scale and attention mechanism. *Eng Appl Artif Intell* 123:106359
- Zhong Z, Ma Z (2021) A novel defect detection algorithm for flexible integrated circuit package substrates. *IEEE Trans Ind Electron* 69(2):2117–2126
- Kim J, Ko J, Choi H, Kim H (2021) Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder. *Sensors* 21(15):4968
- Deng Y-S, Luo A-C, Dai M-J (2018) Building an automatic defect verification system using deep neural network for pcb defect classification. In: 2018 4th International Conference on frontiers of signal processing (ICFSP), pp 145–149. IEEE
- Annaby M, Fouda Y, Rushdi MA (2019) Improved normalized cross-correlation for defect detection in printed-circuit boards. *IEEE Trans Semicond Manuf* 32(2):199–211
- Kuo C-FJ, Tsai C-H, Wang W-R, Wu H-C (2019) Automatic marking point positioning of printed circuit boards based on template matching technique. *J Intell Manuf* 30(2):671–685
- Hua G, Huang W, Liu H (2018) Accurate image registration method for pcb defects detection. *J Eng* 2018(16):1662–1667
- Chen C, Li K, Teo SG, Zou X, Li K, Zeng Z (2020) Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. *ACM Trans Knowl Discov Data (TKDD)* 14(4):1–23
- Yang R, Singh SK, Tavakkoli M, Karami MA, Rai R (2023) Deep learning architecture for computer vision-based structural defect detection. *Appl Intell* 53(19):22850–22862
- Shao R, Zhou M, Li M et al (2024) TD-Net: tiny defect detection network for industrial products. *Complex Intell Syst* 10:3943–3954. <https://doi.org/10.1007/s40747-024-01362-x>
- Ding R, Dai L, Li G, Liu H (2019) Tdd-net: a tiny defect detection network for printed circuit boards. *CAAI Trans Intell Technol* 4(2):110–116
- Hu B, Wang J (2020) Detection of pcb surface defects with improved faster-rcnn and feature pyramid network. *Ieee Access* 8:108335–108345
- Zhang L, Chen J, Chen J, Wen Z, Zhou X (2024) Ldd-net: lightweight printed circuit board defect detection network fusing multi-scale features. *Eng Appl Artif Intell* 129:107628
- Lin Q, Zhou J, Ma Q, Ma Y, Kang L, Wang J (2022) Emra-net: a pixel-wise network fusing local and global features for tiny and low-contrast surface defect detection. *IEEE Trans Instrum Meas* 71:1–14
- Li C-J, Qu Z, Wang S-Y, Bao K-H, Wang S-Y (2021) A method of defect detection for focal hard samples pcb based on extended fpn model. *IEEE Trans Compon Packag Manuf Technol* 12(2):217–227
- Yang L, Fan J, Huo B, Li E, Liu Y (2022) A nondestructive automatic defect detection method with pixelwise segmentation. *Knowl-Based Syst* 242:108338
- Li W, Zhang H, Wang G, Xiong G, Zhao M, Li G, Li R (2023) Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. *Robot Comput-Integr Manuf* 80:102470
- Zhuxi M, Li Y, Huang M, Huang Q, Cheng J, Tang S (2022) A lightweight detector based on attention mechanism for aluminum strip surface defect detection. *Comput Ind* 136:103585

19. Wan G, Fang H, Wang D, Yan J, Xie B (2022) Ceramic tile surface defect detection based on deep learning. *Ceram Int* 48(8):11085–11093
20. Lu Q, Lin J, Luo L, Zhang Y, Zhu W (2022) A supervised approach for automated surface defect detection in ceramic tile quality control. *Adv Eng Inform* 53:101692
21. Su Y, Yan P, Yi R, Chen J, Hu J, Wen C (2022) A cascaded combination method for defect detection of metal gear end-face. *J Manuf Syst* 63:439–453
22. Zhang R, Wen C (2022) Sod-yolo: a small target defect detection algorithm for wind turbine blades based on improved yolov5. *Adv Theory Simul* 5(7):2100631
23. Lian J, Jia W, Zareapoor M, Zheng Y, Luo R, Jain DK, Kumar N (2019) Deep-learning-based small surface defect detection via an exaggerated local variation-based generative adversarial network. *IEEE Trans Ind Inf* 16(2):1343–1351
24. Li X, Zhu Y (2024) A real-time and accurate convolutional neural network for fabric defect detection. *Complex Intell Syst* 10:3371–3387. <https://doi.org/10.1007/s40747-023-01317-8>
25. Li Z, Wei X, Hassaballah M, Li Y, Jiang X (2024) A deep learning model for steel surface defect detection. *Complex Intell Syst* 10(1):885–897
26. Shang H, Sun C, Liu J, Chen X, Yan R (2023) Defect-aware transformer network for intelligent visual surface defect detection. *Adv Eng Inform* 55:101882
27. Dang LM, Wang H, Li Y, Nguyen TN, Moon H (2022) Defecttr: End-to-end defect detection for sewage networks using a transformer. *Constr Build Mater* 325:126584
28. Gao L, Zhang J, Yang C, Zhou Y (2022) Cas-vswin transformer: a variant Swin transformer for surface-defect detection. *Comput Ind* 140:103689
29. Wang J, Xu G, Yan F, Wang J, Wang Z (2023) Defect transformer: an efficient hybrid transformer architecture for surface defect detection. *Measurement* 211:112614
30. Zhu W, Zhang H, Zhang C, Zhu X, Guan Z, Jia J (2023) Surface defect detection and classification of steel using an efficient Swin transformer. *Adv Eng Inform* 57:102061
31. Üzen H, Türkoğlu M, Yanikoglu B, Hanbay D (2022) Swinmfinet: Swin transformer based multi-feature integration network for detection of pixel-level surface defects. *Expert Syst Appl* 209:118269
32. Liu T, Cao G-Z, He Z, Xie S (2024) Refined defect detector with deformable transformer and pyramid feature fusion for PCB detection. *IEEE Trans Instrum Meas* 73:5001111. <https://doi.org/10.1109/TIM.2023.3326460>
33. Lim J, Lim J, Baskaran VM, Wang X (2023) A deep context learning based pcb defect detection model with anomalous trend alarming system. *Results Eng* 17:100968
34. Xuan W, Jian-She G, Bo-Jie H, Zong-Shan W, Hong-Wei D, Jie W (2022) A lightweight modified Yolox network using coordinate attention mechanism for pcb surface defect detection. *IEEE Sens J* 22(21):20910–20920
35. Li J, Gu J, Huang Z, Wen J (2019) Application research of improved yolo v3 algorithm in pcb electronic component detection. *Appl Sci* 9(18):3750
36. Mamidi JSSV, Sameer S, Bayana J (2022) A light weight version of pcb defect detection system using yolo v4 tiny. In: 2022 International Mobile and Embedded Technology Conference (MECON), pp 441–445. IEEE
37. Chen W, Huang Z, Mu Q, Sun Y (2022) Pcb defect detection method based on transformer-yolo. *IEEE Access* 10:129480–129489
38. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: European Conference on computer vision, pp 213–229. Springer
39. Lee C-Y, Xie S, Gallagher P, Zhang Z, Tu Z (2015) Deeply-supervised nets. In: Artificial intelligence and statistics, pp 562–570. Pmlr
40. Yu H, Zhang Z, Qin Z, Wu H, Li D, Zhao J, Lu X (2018) Loss rank mining: a general hard example mining method for real-time detectors. In: 2018 International Joint Conference on neural networks (IJCNN), pp 1–8. IEEE
41. Köksal A, Tuzcuoğlu Ö, İnce KG, Ataseven Y, Alatan AA (2022) Improved hard example mining approach for single shot object detectors. In: 2022 IEEE International Conference on image processing (ICIP), pp 3536–3540. IEEE
42. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: single shot multibox detector. In: Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp 21–37. Springer
43. Zhou X, Wang D, Krähenbühl P (2019) Objects as points. *arXiv preprint arXiv:1904.07850*
44. xuehao ma (2022) FastestDet: Ultra lightweight anchor-free real-time object detection algorithm. <https://github.com/dog-qiuqiu/FastestDet>
45. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on computer vision, pp 2980–2988
46. Sunkara R, Luo T (2022) No more strided convolutions or pooling: A new cnn building block for low-resolution images and small objects. In: Joint European Conference on machine learning and knowledge discovery in databases, pp 443–459. Springer
47. Jocher G (2020) Ultralytics/yolov5: v3.1-Bug fixes and performance improvements, version v3.1. Zenodo. doi: 10.5281/zenodo.4154370. <https://github.com/ultralytics/yolov5>
48. Wang C-Y, Bochkovskiy A, Liao H-YM (2023) Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, pp 7464–7475
49. Jocher G, Chaurasia A, Qiu J (2023) Ultralytics YOLO, version 8.0.0. <https://github.com/ultralytics/ultralytics>
50. Quan Y, Zhang D, Zhang L, Tang J (2023) Centralized feature pyramid for object detection. *IEEE Trans Image Process* 32:4341–4354. <https://doi.org/10.1109/TIP.2023.3297408>
51. Ge Z, Liu S, Wang F, Li Z, Sun J (2021) Yolox: exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.