IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 October 2023, accepted 16 October 2023, date of publication 19 October 2023, date of current version 25 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3325885

RESEARCH ARTICLE

Precise Detection for Dense PCB Components Based on Modified YOLOv8

QIN LING, NOR ASHIDI MAT ISA, AND MOHD SHAHRIMIE MOHD ASAARI
School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, Nibong Tebal, Pulau Pinang 14300, Malaysia

Corresponding author: Nor Ashidi Mat Isa (ashidi@usm.my)

This work was supported by Geran Penyelidikan Pemadanan Universiti Sains Malaysia (USM)-Industri titled "Defect Assessment of Solder Joint Based on X-Ray Images Using Integration of Adaptive Image Processing and Fuzzy-Based Look-Up-Table (LUT) Approaches" under Grant 1001.PELECT.8070024.

ABSTRACT Effective detection of dense printed circuit board (PCB) components contributes to the optimization of automatic flow of production. In addition, PCB component recognition is also the essential prerequisite for early defect detection. Current PCB component detection approaches are not adept in both rapid and precise detection. YOLOv8 models have exhibited effective performances for detecting common objects, such as person, car, chair, dog etc. However, it is still tricky for YOLOv8 models to inspect dense and disparate PCB components precisely. Thus, a novel convolution neural network (CNN) model is proposed for dense PCB component detection by introducing several modifications onto YOLOv8. First, creative C2Focal module is designed as the core element of the backbone, combining both fine-grained local and coarse-grained global features concurrently. Then, the lightweight Ghost convolutions are inserted to effectively reduce the computation cost, meanwhile maintaining the detection performance. Finally, a new bounding box regression loss that is Sig-IoU loss, is proposed to facilitate the prediction regression and promote the positioning accuracy. The experiments on our PCB component dataset demonstrate that our proposed model performs the highest mean average precisions of 87.7% (mAP@0.5) and 75.3% (mAP@0.5:0.95) respectively, exceeding other state-of-the-arts. Besides, the detection speed hits 110 frames per second using RTX A4000, which is potential to realize the real-time inspection.

INDEX TERMS PCB component detection, high-precision, lightweight, Sig-IoU loss, ghost convolution.

I. INTRODUCTION

Printed circuit boards (PCBs) are the backbone in modern electronic information industry, carrying integrated circuits, resistors, capacitors, and other electronic components. The rapid development of electronic techniques has led to the miniaturization, integration, and diversification of PCBs [1]. This trend is expected to continue in the future, as PCBs become increasingly smaller, more complex, and more specialized. The surface mount technique realizes automatic assembly of components at a high density and speed. During the manufacturing and assembly process, the rapid and precise recognition and positioning of PCB components is increasingly important [2]. The effective detection of dense components contributes to the optimization of automatic flow of production. In addition, component recognition is an essential prerequisite for early PCB defect detection, which has a profound impact on the cost of quality control in the PCB industry.

PCB components are greatly diverse in types, appearances and dimensions. This makes traditional inspection methods based on human vision ineffective and prone to errors. As a result, these methods cannot meet the practical requirements for automated production [3]. In order to mitigate the dependence on manual inspection, the automatic optical inspection (AOI) technique integrated with machine vision has been widely implemented, especially in modern factories [4].

Researchers have proposed a variety of detection methods for PCB components based on machine vision [5], [6], [7], [8], [9], [10]. They modified and then applied traditional machine learning algorithms such as genetic algorithm [5],

The associate editor coordinating the review of this manuscript and approving it for publication was Porfirio Tramontana.

hybrid genetic algorithm [6], random forest pixel classifier [7], template matching [8], particle swarm optimization algorithm [9], [10] to classify and recognize PCB components. However, these methods have their intrinsic drawbacks.

The detection results by aforesaid works are closely related to manual designed features, which are normally extracted using traditional image processing. During the process of image pre-processing, templates are usually required to execute image matching with inspected images. The acquisition of perfect templates is cumbersome and costly. Additionally, pixel-level alignment between the templates and tested images must be always held to avoid extra noise. To keep the good performance, complicated and stable illumination environment is indispensable.

Moreover, most traditional machine learning approaches for component detection are two-stage or multi-stage. The combination of several image processing techniques has to be done to position the targets, then different machine learning algorithms are accordingly performed to make the classification. These image processing operations and machine learning algorithms usually calculate intricate matrix with high sapcetime complexity, resulting in huge computation cost. The parameter preference and optimization for those machine learning algorithms can be a burdensome task as well. As a consequence, they definitely struggle to handle high-resolution PCB images at a high speed.

In recent years, there was an explosive growth in deep learning methods, especially convolution neural networks (CNNs) in the computer vision studies. They have stronger feature extraction capability and outstanding anti-interference ability. Deep CNNs are proficient at extracting multi-level features even though objects are placed in nosy surroundings. Many classical CNN models have achieved significant performances in object detection and segmentation [11]. Therefore, researchers have applied various deep learning models to work out target recognition or inspection in the industry [12], [13].

In addtion, some researchers have also achieved significant results in the segmentation of concrete crack or damage [14], [15] and medical examination [16], by proposing and introducing specific attention module into deep learning networks. In [14], Faster RCNN [17] was used to detect cracks, then a modified tubularity flow field (TuFF) algorithm was applied to these bounding boxes from Faster RCNN to segment cracks on pixel-level. Finally, a modified distance transform method was implemented to calculate crack thickness and length from the segmented cracks. In [15], to overcome the difficulty of preparing ground truth for internal damage, an attention-based generative adversarial network (AGAN) was proposed to produce synthetic internal damage images for training the proposed segmentation network. The attention modules implemented in the generator and discriminator of the AGAN can improve the performance of the generative adversarial network. The authors in [16] proposed a dual encoder–decoder named Polyp Segmentation Network (PSNet) to segment colorectal polyps. Several novel modules for the polyp segmentation were proposed, including PS encoder, PS decoder, the merge module, and the relevant components, such as the local feature extraction module and the dual complex convolutional module. The newly proposed dual encoder combines both a CNN-based encoder and a transformer encoder. The dual decoder contains two synchronous decoders, PS decoder and the transformer decoder, and a synchronous set of merge modules.

Several studies [18], [19], [20], [21], [22] have implemented various deep learning models to detect PCB components. For instance, Li et al. [18] modified YOLOv3 network [23] to get the detection mean average precision (mAP) of 93.07% for 29 component categories. Inspired by Faster R-CNN [17] and PeleeNet [24], Shen et al. [19] built a novel lightweight CNN model for PCB component inspection. This method boosted the detection precision to 85.8% for 12 different electronic components.

Although these CNN based works have made great progress in PCB component detection than traditional image processing and machine learning methods, there is still room for improvement. All the CNN models they used have a large number of parameters, which leads to slow detection speed, especially for high-resolution inputs. Moreover, vast parameter amount produces heavyweight trained models, leading to the difficulty of embedding in real industrial inspection devices. In addition, the detection precisions of past published works still need to be promoted to fulfill the industrial requirement. Therefore, proposing a novel end-to-end lightweight CNN model for dense PCB component inspection is urgent. Recently, YOLOv8-nano [25] has been renowned in the object detection community due to its real-time speed and excellent precision. Nevertheless, because of the large scale variance of size, diverse appearance and dense distribution of PCB components, YOLOv8-nano needs to be further improved to accomplish fleet and precise inspection for PCB components.

In light of the aforementioned concerns, we design a novel lightweight deep learning model used to detect PCB components on bounding box level, by introducing a couple of improvements into YOLOv8-nano. A novel C2Focal module integrating FocalNeXt [26] block is proposed to constitute the backbone of the model. C2Focal module can mitigate the adverse impact from the scale variance to achieve preferable detection results, only introducing little additional computation. Then, the common convolution is replaced by a lighter Ghost convolution [27], that produces richer ghost features from cheaper computation cost. Moreover, a novel bounding box regression loss named Sig-IoU is proposed to accelerate the convergence and promote the regression accuracy. As a result, our method attains substantially rapid and precise detection for dense PCB components. The main contributions of this paper can be summarised as follows:

- Instead of C2f module in YOLOv8, creative C2Focal module integrating FocalNeXt block is proposed.

FocalNeXt block involving dilated depthwise convolution [28] and two jump joins, can expand the receptive field of neurons and combine both fine-grained local and coarse-grained global features concurrently. Thus, C2Focal module can handle the scale variance issue to perform favorable detection performances.
- The lightweight Ghost convolution is introduced to replace the common convolution so that the requirement for computing resources is effectively reduced without affecting the performance of the model.
- A new bounding box regression loss function that is Sig-IoU loss is proposed, to replace the original Complete IOU (CIoU) loss [29], facilitating the prediction regression and promoting the positioning accuracy.

Benefiting by the stated contributions, our method can attain outstandingly precise detection for 32 classes of densely packed PCB components, while meet the need of real-time inspection in industry. According to the experiment results, our model outperforms classical state of the arts (SOTAs) with the best detection mAP@0.5 of 87.7%, and mAP@0.5:0.95 of 75.3%, respectively.

The remaining part of this paper is organized as follow: the review of relevant works is presented in Section II. The details and improvements of the proposed method are illustrated in Section III, followed by the experiments and discussions in Section IV. Finally, Section V summarises the paper.

II. LITERATURE REVIEW

A. MACHINE LEARNING BASED METHODS FOR PCB INSPECTION

Machine learning algorithms for PCB component detection are closely integrated with image processing, which usually requires qualified templates. Crispin et al. [5] applied genetic algorithm combined with normalised cross correlation (NCC) template matching to locate and identify resistors in PCBs. The mean inference time reached 39.5s per one small size image. However, this method had large computation cost, leading to a slow running. Moreover, both building the qualified template and optimal parameter choice for genetic algorithm are intricate and experiential, contributing to a low detection efficiency. Mashohor et al. [6] proposed a hybrid genetic algorithm to detect missing components and segment solder joints. With the help of image processing techniques, this method identified missing components or solder joints, but failed to classify them.

Li et al. [7] used random forest pixel classifier to segment components based on depth images of PCBs. The component identification rate of 83.64% was performed for real PCB images. Yin [8] designed a multi-level template matching algorithm to detect PCB components. A fast coarse matching was implemented to locate the similar region, then a precise matching was carried out to estimate whether the similar regions were targeted components. This method detected resistors, inductors and capacitors at the precisions of 95.3%, 94.1% and 96.5% respectively.

Wang et al. [9] combined multi-template matching and species based particle swarm optimization algorithm to inspect PCB components. The addition of swarm optimization reduced the computation cost. The average detection rate for six resistors was 100% and the best mean running time was about 25.96s for a 762×612 pixel image. Superior to [9], Dong et al. [10] proposed an improved particle swarm optimization algorithm to detect PCB components with the assistance of template matching. This particle swarm optimizer used the notion of chaos and species to determine neighborhood best values. As the result, the best average running time for the same images in [9] was reduced to 15.71s, maintaining the successful rate of 100%.

According to previous reviews, traditional machine learning based works for detecting PCBs are summarised in Table 1. In spite of good detection performances for PCB inspection, traditional machine learning based approaches still have certain limitations. Many image processing operations must be executed to get good accuracy and localize the targets. Normally, image processing such as template matching or image subtraction needs a perfect template, that is extremely sensitive to variable surroundings. Thus, these methods have poor generalization ability and weak robustness. Furthermore, all of them are multi-stage approaches. In consequence, none of them may meet current demands of real-time and automatic detection in the real applications at all.

B. DEEP LEARNING BASED METHODS FOR PCB INSPECTION

Different from traditional machine learning algorithm based methods, CNNs can achieve PCB component detection automatically without the intervene of reference images. Advanced CNN models have performed great results in the object detection for industrial products. Similarly, many works utilized CNNs to detect PCB defects and components, pursuing progress in performances.

Mamidi et al. [30] proposed a PCB inspection system based on a YOLOv4-tiny [31] network to identify PCB surface defects. The overall mAP achieved 79.72% for 6 types of defects, such as spurious copper, short, missing hole, spur, open circuit and mouse bite. Lan et al. [32] designed an improved YOLOv3 network to detect PCB surface defects. The batch normalization layer was combined to the convolutional layer to improve the forward inference speed. GIoU loss function was used to improve the detection effect of the model on small targets. Moreover, K-means++ clustering was applied to get more appropriate anchors. This method performed the mAP of 92.13%, with the detection speed of 63 FPS, which was superior to the original YOLOv3 model.

Based on extended FPN, a PCB defect detection algorithm was introduced in [33]. The backbone was built upon ResNet-101. To precisely classify the PCB defects, focal loss function was introduced. The mAP of this method achieved

TABLE 1. Summary of machine learning based works.

Work	Proposed approach	Results	Merits	Limitations
[5]	Combined genetic algorithm with template matching	Detected resistors successfully	Improving the image matching accuracy	- Required template matching and reference images - Time consuming, 39.5s per small image - Selection of parameters
[6]	Hybrid genetic algorithm with various image processing operations	Detected missing components and segmented solder joints	Produced high accuracy for image registration	- Required template matching and reference images - Failed to achieve classification - Time consuming - Selection of parameters
[7]	Random forest pixel classifier	Recognised PCB components	High recognition rate for synthetic images	- Poor performance for real images - Only classified defective components without the exact defect types - Hard to choose the optimal parameters and random seed - Time consuming
[8]	Multi-level template matching algorithm	Detected resistors, inductors and capacitors	High precision	- Required template matching and reference images - Low precision when handling small components - Focused on sparsely distributed components
[9]	Combined multi-template matching with particle swarm optimization	Detected 6 resistors	High detection rate of 100%	- Required template matching and reference images - Time consuming, 25.96s per image - Only focused on resistors
[10]	Improved particle swarm optimization algorithm	Detected 6 resistors	- High detection rate of 100% - Less running time than [9]	- Required template matching and reference images - Still time consuming, 15.71s per image - Only focused on resistors

96.2% for 6 types of PCB surface defects in the public dataset.

Evangelidis et al. [34] proposed a methodology to replace expensive laboratory sensors with low-cost ones by introducing a data-driven soft sensor model. With the assistance of the methodology, a PCB inspection system for glue deposits, which relied on an industrial camera and Jetson AGX Xavier unit, was designed. An instance segmentation network based on Mask R-CNN [35] and Faster R-CNN [17] was proposed to predict coordinates of each glue deposit and classify their types. Then, an improved residual architecture (R^2esNet) was proposed for volume estimation of glue deposits. The method achieved greatly satisfactory results for glue defects in PCB manufacturing, which can facilitate the quality inspection process efficiently, and optimize the production process.

Li et al. [18] proposed an improved YOLOv3 to detect PCB components, where an extra output layer was added. The detection mAP for PCB components reached 93.07%. However, the introduction of extra prediction layer brought in additional computation, delaying the inference. Contrast to the heavyweight model in [18], Shen et al. [19] proposed a lightweight deep learning model, integrating Faster R-CNN and PeleeNet, which performed rapid and precise detection for various PCB components. The detection precision of this method attained 85.8% and the detection speed reached 27 FPS.

Zhang et al. [20] just used YOLOv3 to recognize PCB components without any modification. The overall detection rate for 11 types of PCB components was about 92%, and the running time was 0.55s for each input image. Based on depthwise convolution and separable convolution, Chen et al. [21] designed a simple CNN model to classify normal components and defective components. The detection precisions for qualified products and defective products reached 96.96% and 95.70% respectively. The average inference time reached 27ms per component, which is almost real-time. However, it only detected a small region of 42×42 pixel instead of a whole PCB image.

Liu et al. [22] proposed Gaussian IoU (GsIoU) loss function for bounding box regression to replace the original one in YOLOv4. The introduction of GsIoU produced the mAP increase of 3.3% compared with the baseline. However, the dataset only contained 6 types of PCB components, which are large or medium-size, and dual in-line packaged in the board sparsely. Therefore, this method may struggle to detect tiny and dense PCB components.

Deep learning based works for PCB inspection are summarised in Table 2. Compared with machine learning based methods, CNN based methods can extract and fuse feature maps from different depths so that they are immune to complex noises. Nevertheless, there still exist several limitations. Concerning the PCB component inspection, components used in past works are neither abundant in categories nor densely distributed, mitigating the difficulty of recognition. In addition, most of them could not attain rapid and precise component detection simultaneously. Therefore, there is an urgent need for a novel lightweight and high-precision CNN model detecting densely packed PCB components.

III. METHODOLOGY

A. DATA PREPARATION

1) DATA ACQUISITION

We use 47 physical PCB assembly images from the public dataset in [36]. There are 18108 objects in total, meaning that each image contains 385 components on average. What's more, sometimes 50% of components such as tiny resistors and capacitors are concentrated in a small area, only taking up 10% of the entire image. In a word, the components are densely placed on the board or a small local area of the board.

The original data classified the PCB components according to their types and silkscreen number, such as 'resistor R10',

TABLE 2. Summary of deep learning based works.

Work	Proposed approach	Results	Merits	Limitations
[30]	YOLOv4-tiny without modification	Identified 6 types of PCB surface defects	NA	- Low precision - Not real-time yet
[32]	Improved YOLOv3	Identified 6 types of PCB surface defects	High inference speed	- Precision needs to be promoted further - Large model size
[33]	Extended FPN	Detected 6 types of PCB surface defects	High precision	- Not real-time yet - Heavyweight model
[34]	An instance segmentation network and a R^2esNet	Detected glue defects in PCBs	Satisfactory results in real PCB manufacturing case	The generalization ability in other industrial cases needs to be evaluated further
[18]	Improved YOLOv3	Detected various PCB components	High precision	- Not real-time yet - Heavyweight model
[19]	Integrated Faster R-CNN and PeleeNet	Detected PCB components	High precision	- Template marching or image comparison was required - Three-stage method - Not real-time yet
[20]	YOLOv3 without modification	Recognised 11 types of PCB components	NA	- Precision needs to be promoted further - Large model size - Not real-time yet
[21]	CNN model based on depthwise separable convolution	Classified normal components and defective components	High classification rate	- Not real-time yet - Only detected a small region instead of a whole PCB image
[22]	Proposed GsIoU loss into YOLOv4	Detected 6 types of PCB components	Better performance than original YOLOv4	- Not real-time yet - Components are large and distributed sparsely, easy to be detected

* Note: NA indicates that information is not available.

'connector CN2' etc. Thus, the labels in this data are very huge and detailed. Here, for the task of component identification, we remove the silkscreen numbers of electronic components and group the same type of components together. All the silkscreen numbers printed on the board are labelled as one class, that is text. Finally, there are totally 32 classes of components (including text) in our dataset.

Random cropping was implemented to obtain sub-images with the size of 640 × 640 pixels. To keep off the over-fitting problem, data augmentation techniques must be applied to produce abundant instances. In this work, we used traditional geometrical transformation and morphological operations, including rotation, flipping, transpose, adding noise and color space conversions including RGB shift and random brightness contrast. Finally, 11330 images was generated, in which training set, validation set and test set were allocated on the ratio of 7: 2: 1. The number and input sizes of training, validation and testing images are tabulated in Table 3. Our data examples are illustrated in Fig. 1. It shows that numerous tiny components and texts are densely and closely packed on the board.

2) DATA ANALYSIS

The distribution and dimensions of labels used for training can be found in Fig. 2. The amounts of each target classes are tabulated in Table 4.

There are a total of 180818 instances in our training set as shown in Table 4. Fig. 2(a) representing the label positions indicates that targets are uniformly distributed in the whole images instead of gathering at a specific region. The x-axis represents the ratio of label center coordinates to the image width, while the y-axis represents the ratio of label center coordinates to the image height. These ratios provide a standardized measure for positioning labels within the image. Fig. 2(b) exhibits the ratios of label size to the image size, indicating the PCB components are extremely diverse in dimensions. The SMD components are extremely tiny and dense, while the PTH (Plated through holes) components are relatively large-sized, resulting in the large scale variance.

TABLE 3. The number and input size of images in different sets.

Dataset	Training	Validation	Testing
Number of images	7930	2278	1122
Input size of images	640×640×3	640×640×3	640×640×3

B. YOLOv8

Based on the success of previous series of YOLO models, YOLOv8 [25] was built by introducing some improvements. The architecture of YOLOv8 is shown in Fig. 3, which can be mainly divided into three ends, including Backbone, Neck and Prediction. More details of each component in the YOLOv8 are introduced in later sections.

1) THE BACKBONE

Before feeding into the network, input images are processed automatically. Concerning training inputs, Mixup augmentation [37] are applied to markedly fertilize the context information of input images and diminish the obstruction from noisy samples. In addition, several image processing techniques such as translation, scaling etc. can be applied randomly to augment input images automatically. Afterwards, augmented images are uniformly scaled and filled to a standard size.

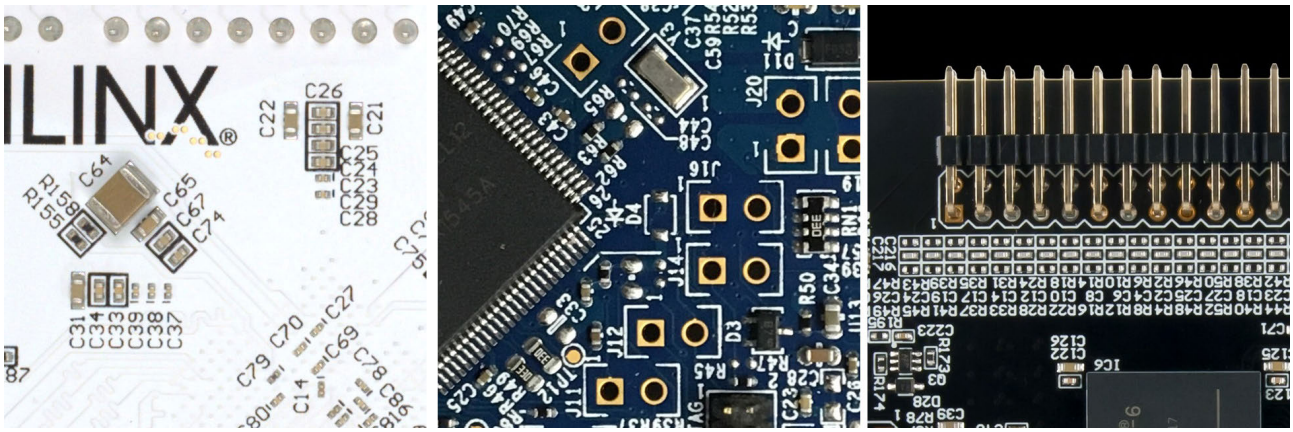

FIGURE 1. Examples in our dataset.

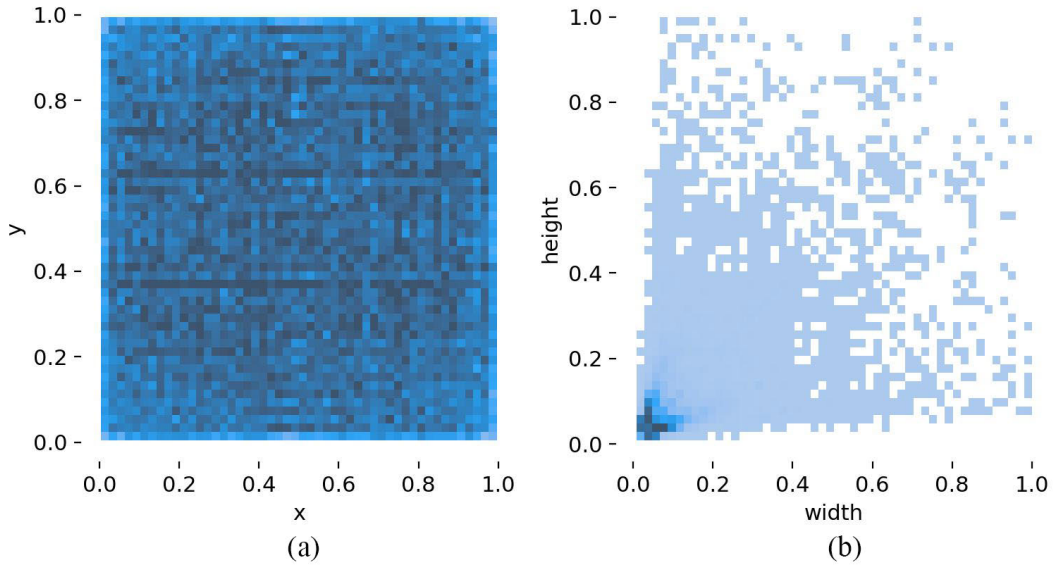

FIGURE 2. Label distribution and sizes, (a) label positions, (b) label size.

In contrast, test inputs are adaptively scaled to the setting size without any augmentation to maintain a rapid inference.

The backbone extracting feature maps is extremely vital to the performances of deep learning models. In YOLOv8, the backbone is composed of elementary convolution layers, newly proposed C2f modules and the spatial pyramid pooling- fast (SPPF) block.

a: C2F MODULE

The architecture of C2f module can be found in Fig. 4. The main components are convolutions followed by batch normalization and SiLU [38] activation function and several Bottleneck blocks. There are two different types of bottleneck block. When the add parameter is set to true, the bottleneck involves residual shortcut addition which is similar to the Resnet block [39]. Otherwise, the bottleneck is simply the stack of two convolutions.

According to the architecture, it can be found that the C2f module integrates the merits of Efficient layer aggregation networks (ELAN) [40] and CSPNet [41]. ELAN adopts less transition layers to shorten the shortest gradient path of the flow. Inspired by this improved stack tactic, C2f module adds more parallel branches to obtain richer gradient information without stretching the gradient path. Compared with C3 block in YOLOv5 [42], C2f module is able to get higher accuracy and more reasonable latency.

b: SPPF

In YOLO series, SPP was used to observe feature maps with multi-scale receptive fields, assisting the model obtain more abundant information. SPPF, whose structure is shown in Fig. 5, was proposed in YOLOv5 [42] and still used in YOLOv8.

TABLE 4. Amounts of each labels in the training set.

Label	connector	resistor	pads	emi filter	capacitor	resistor network	text
Amount	5493	21406	3588	560	27110	674	92228
Label	pins	ic	test point	capacitor jumper	clock	component text	switch
Amount	3244	3686	3157	30	331	9496	574
Label	led	button	diode	electrolytic capacitor	transistor	ferrite bead	jumper
Amount	2276	854	756	2210	935	305	798
Label	inductor	zener diode	fuse	resistor jumper	potentiometer	diode zener array	transformer
Amount	717	72	80	70	90	30	4
Label	display	battery	heatsink	buzzer			
Amount	9	5	16	14			

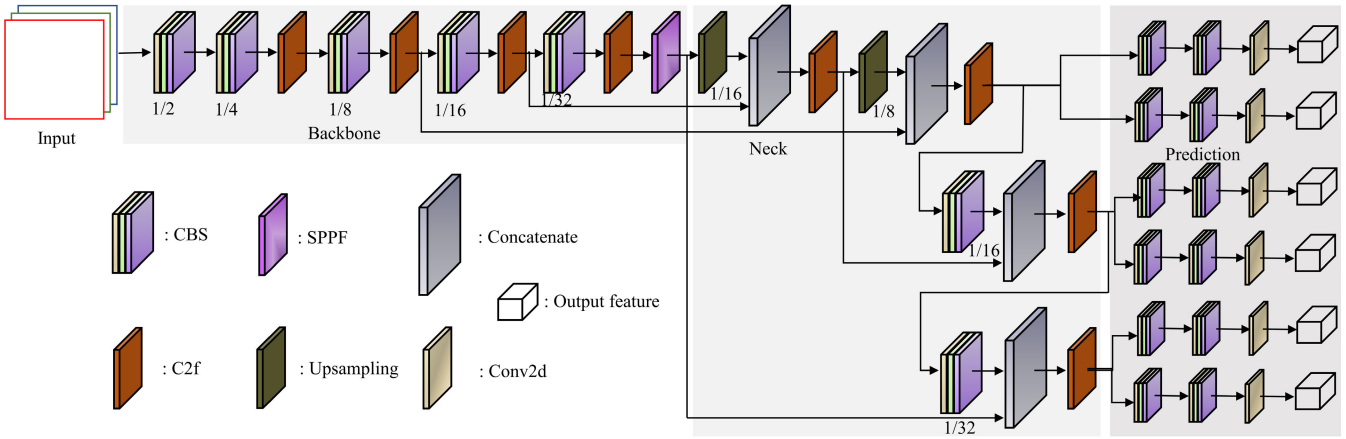

FIGURE 3. The architecture of YOLOv8.

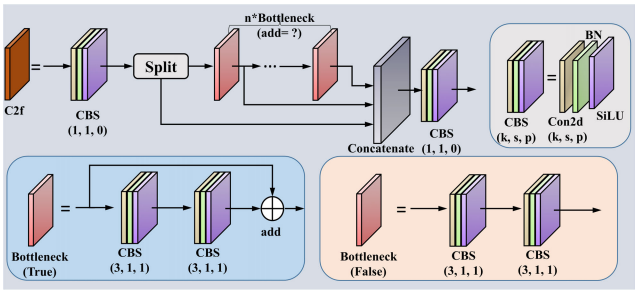

FIGURE 4. The structure of C2f module (Note: k means the kernel size, s means stride and p means padding of convolution).

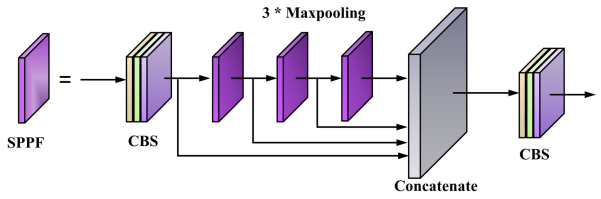

FIGURE 5. The architecture of SPPF.

2) THE NECK

The core insight of neck end is the multi-level feature fusion through FPN. The fusion between deep features and shallow features makes the model maintain stronger semantic information and more image details simultaneously, beneficial to address the matter of scale variance. The feature pyramid network used in YOLOv8 is PANet [43], whose structure can be found in Fig. 6. Compared with traditional FPN that fuses feature maps from the top to bottom, PANet introduces another fusion from the bottom to top. The overall characterization ability of the model is enhanced.

3) THE PREDICTION HEAD

Decoupled prediction heads are used in YOLOv8, as shown in Fig. 7. In previous YOLO series, for the feature maps fed into

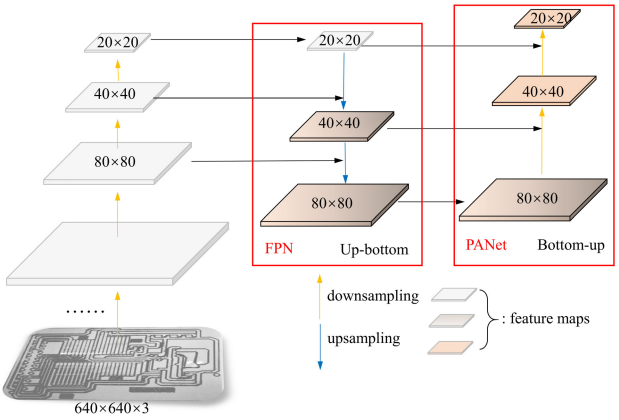

FIGURE 6. The PANet structure used in YOLOv8.

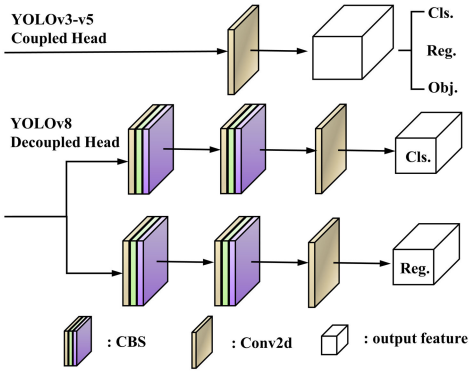

FIGURE 7. The structure of one Decoupled Head.

the head, the predictions of classification and bounding box regression are done simultaneously, sharing the parameters of the previous layer. However, YOLOv8 decouples them to predict separately. The decoupled prediction head can accelerate the model convergence and improve the detection accuracy [44].

C. THE PROPOSED METHOD

Although YOLOv8 has obtained significant detection performance promotions on the public COCO dataset [45], the generalization abilities on custom datasets have not been widely verified. Among YOLOv8 series models with different sizes, the lightest YOLOv8-nano has the highest detection speed with comparable accuracy. However, due to the densely packed PCB components with large scale variance and diverse appearances, it still needs to be further improved to accomplish fleet and precise detection. Here, concerning several improvements on YOLOv8-nano, we propose a novel detection model for PCB components. As illustrated in Fig. 8, the main improvements involve the backbone, which is modified by introducing Ghost convolution modules and C2Focal modules. Moreover, a new Sig-IoU loss function for bounding box regression is used to replace the original CIoU loss.

1) GHOST CONVOLUTION

Feature maps generated by traditional convolution modules have many similar pairs, like a ghost of each another. These ghost pairs are called redundant information, which are vital for a deep learning model to produce a comprehensive understanding of the input images. Thus, instead of abandoning these similar pairs, Han et al. [27] proposed a novel Ghost module to retain them in a cost-efficient way. In the Ghost module, output feature maps are considered as the combination of intrinsic features and ghost features. The intrinsic feature maps are generated by common convolutions, but the ghost feature maps are generated by cheap linear transformations on each intrinsic feature map.

The architecture of Ghost convolution module used in our model is presented in Fig. 9. The cheap linear operations can be achieved by Depthwise convolution, whose computational cost is much less than the ordinary convolution [28]. In Depthwise convolution, the input feature map is divided into g groups according to input channel, and then ordinary convolution is performed for each group, where g equals to the number of input channel.

Given an input feature map $F_{in} \in R^{c_{in}\times h\times w}$, where h and w are the height and width of the input feature, and c_{in} is the number of input channels, the action of an ordinary convolution layer can be formulated as

$$F_{out} = F_{in} * f + b \tag{1}$$

where * represents the convolution action, b means the bias, $F_{out} \in R^{h'\times w'\times c_{out}}$ means the output feature map, h' and w' are the height and width of the output feature, and c_{out} is the number of output channels. Additionally, $f \in R^{c_{in}\times k\times k\times c_{out}}$ represents the convolution filter, where $k\times k$ means the kernel size. The number of FLOPs of this process is $c_{out}\times h'\times w'\times c_{in}\times k\times k$ and the amount of parameters of this procedure is $c_{out}\times c_{in}\times k\times k$.

Concerning the Ghost convolution, for the primary ordinary convolution producing half output channels, the FLOPs and the amount of parameters are $1/2c_{out}\times h'\times w'\times c_{in}\times k\times k$ and $1/2c_{out}\times c_{in}\times k\times k$ respectively. The FLOPs of Depthwise convolution could be $1\times 1/2c_{out}\times h'\times w'\times k_c\times k_c$, where k_c is the kernel. The amount of parameters is $1\times 1/2c_{out}\times k_c\times k_c$. Therefore, the FLOPs ratio between ordinary convolution and Ghost convolution is

$$\begin{aligned} r_f &= \frac{c_{out}\times h'\times w'\times c_{in}\times k\times k}{\frac{1}{2}c_{out}\times h'\times w'\times c_{in}\times k\times k+\frac{1}{2}c_{out}\times h'\times w'\times k_c\times k_c} \\ &= \frac{c_{in}\times k\times k}{\frac{1}{2}c_{in}\times k\times k+\frac{1}{2}k_c\times k_c}\approx 2 \end{aligned} \tag{2}$$

where $k_c\times k_c$ has the similar magnitude as that of $k\times k$. The parameter amount ratio is

$$\begin{aligned} r_p &= \frac{c_{out}\times c_{in}\times k\times k}{\frac{1}{2}c_{out}\times c_{in}\times k\times k+\frac{1}{2}c_{out}\times k_c\times k_c} \\ &= \frac{c_{in}\times k\times k}{\frac{1}{2}c_{in}\times k\times k+\frac{1}{2}k_c\times k_c}\approx 2 \end{aligned} \tag{3}$$

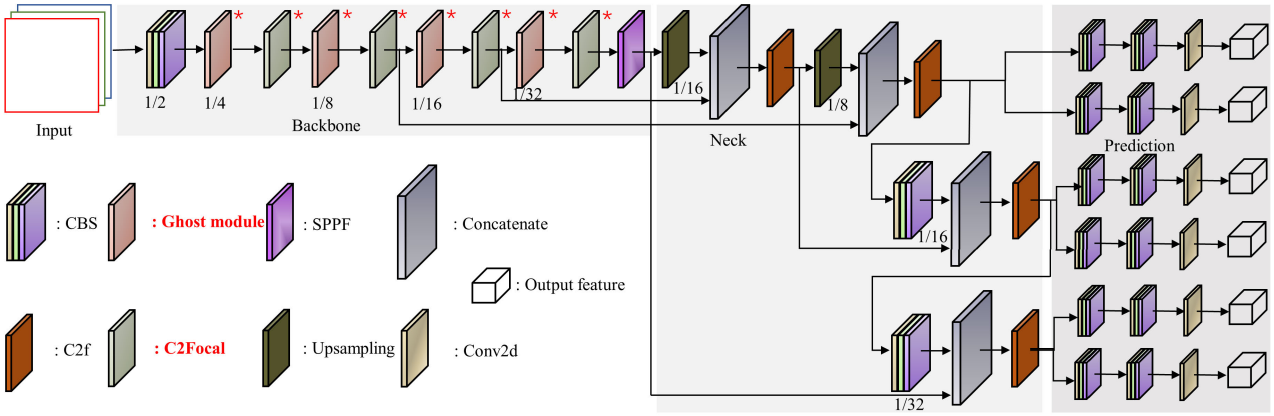

FIGURE 8. The architecture of our proposed model (Note: the marks * indicate where the modifications were implemented).

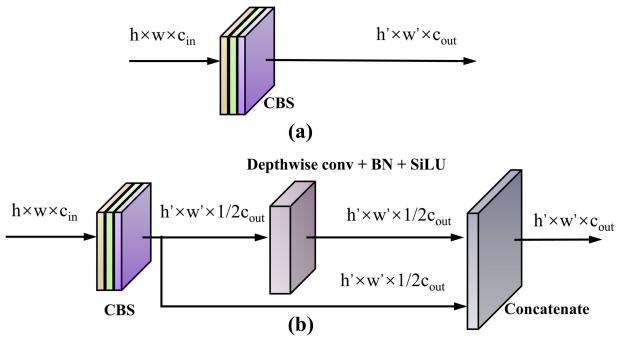

FIGURE 9. A presentation of convolution and Ghost convolution module. (a) ordinary convolution, (b) Ghost convolution.

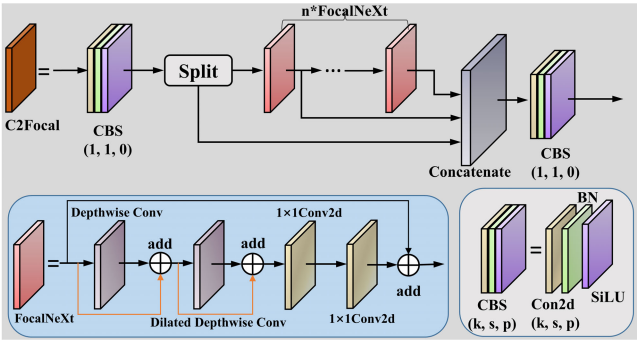

FIGURE 10. The architecture of proposed C2Focal module.

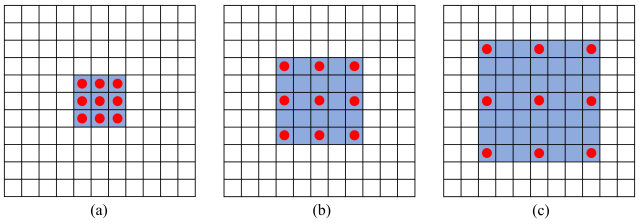

FIGURE 11. Dilated convolution, (a) normal convolution (dilation rate = 1), (b) dilated convolution with dilation rate of 2, (c) dilated convolution with dilation rate of 3 (Note: red circle indicates the element of kernel, the blue area means receptive field).

As the consequence, Ghost convolution module produces feature maps with same number of channels as that produced by ordinary convolution. However, the FLOPs and parameter amount have been cut down to half of ordinary convolution, theoretically contributing to speeding up the module.

2) C2FOCAL MODULE

It is always challenging to detect densely distributed targets at various scales. The common solution for detecting various scale targets is to produce and fuse feature maps with various dimensions. For example, features with strides of 8, 16, 32, 64 and 128 can be used to detect targets of corresponding scales. In the proposed network, only 3 different scale features with strides of 8, 16 and 32 are produced and used for final prediction. The reason is generating deeper features with different scales introduces extra huge parameters, as the numbers of channels of the deep features gradually increase when the feature maps scale down. Therefore, instead of generating deeper features, the FocalNeXt block proposed by Zhang, et al. [26] is introduced to enlarge the receptive field of the model, handling scale variance issue for dense object detection.

The FocalNeXt block, whose structure can be found in Fig. 10, was built based on ConvNeXt block [46]. The core insight is that FocalNeXt block involves dilated convolution and two jump joins to expand the receptive field of neurons. Unlike the normal convolution, dilated convolution introduces a hyper-parameter called the "dilation rate", which defines the spacing between the values of the convolution kernel, as shown in Fig. 11. The normal convolution covers a 3×3 area in the input image, same as the kernel size. While, the dilated convolution with dilation rate of 2 covers a 5×5 region in the input image, which equals to the normal convolution operation with kernel size of 5×5. Thus, dilated convolution can increase the receptive field without increasing the amount of parameters.

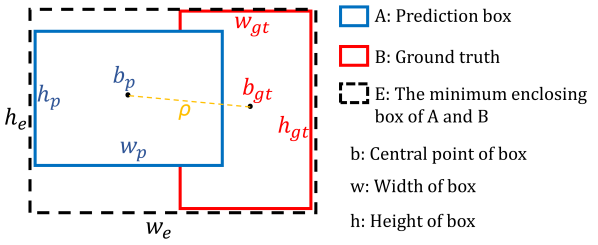

FIGURE 12. The schematic diagram of Sig-IoU loss.

Here, we propose C2Focal module by introducing the FocalNeXt block to replace the original C2f module in YOLOv8. The architecture of our proposed C2Focal module is illustrated in Fig. 10. The dilated convolutions with different dilation rates in the FocalNeXt blocks expand the receptive field and maintain coarse-grained global features. The normal convolutions and jump connections retain fine-grained local information. Thus, the proposed C2Focal module is able to fuse both coarse-grained and fine-grained features. Benefiting from the introduction of C2Focal module, our model has achieved competitive results, meanwhile only introducing insignificant additional computation cost.

3) SIG-IoU LOSS

The loss function of YOLOv8 consists of binary cross-entropy loss for classification and CIoU loss for bounding boxes regression. These two elements are summed with different weights to calculate the overall loss. The CIoU loss can be formulated as

$$\begin{aligned} l_{box} &= l_{IoU} + l_{dis} + l_{asp} \\ &= 1 - IoU + \frac{\rho^2(b_p, b_{gt})}{(w_e)^2 + (h_e)^2} + \alpha\nu \\ \nu &= \frac{4}{\pi^2}(arctan\frac{w_{gt}}{h_{gt}} - arctan\frac{w_p}{h_p})^2 \\ \alpha &= \frac{\nu}{(1 - IoU) + \nu} \\ IoU &= \frac{|A \cap B|}{|A \cup B|} \end{aligned} \tag{4}$$

where b represents the central point of box, ρ represents the Euclidean distance, w and h mean the width and height of corresponding box, as shown in Fig. 12.

The last term $\alpha\nu$ in Equation 4 denotes the discrepancy of the width-to-height ratio. The gradients of ν with respect to w_p and h_p can be calculated as follows

$$\begin{aligned} \frac{\delta\nu}{\delta w_p} &= \frac{8}{\pi^2}(arctan\frac{w_{gt}}{h_{gt}} - arctan\frac{w_p}{h_p}) \times \frac{h_p}{w_p^2 + h_p^2} \\ \frac{\delta\nu}{\delta h_p} &= -\frac{8}{\pi^2}(arctan\frac{w_{gt}}{h_{gt}} - arctan\frac{w_p}{h_p}) \times \frac{w_p}{w_p^2 + h_p^2} \end{aligned} \tag{5}$$

Thus, when the prediction bounding boxes are very tiny, that the width and height range from 0 to 1, the value of $w_p^2 + h_p^2$ will be extremely small, leading to the Gradient explosion. To avoid this issue, we design a novel Sig-IoU loss for bounding box regression, in which the arc-tangent function in the last term is replaced by Sigmoid function. The Sig-IoU loss is formulated as

$$\begin{aligned} l_{sig-box} &= l_{IoU} + l_{dis} + l_{asp} \\ &= 1 - IoU + \frac{\rho^2(b_p, b_{gt})}{(w_e)^2 + (h_e)^2} + \alpha\nu \\ \nu &= (\frac{1}{1 + e^{-w_{gt}/h_{gt}}} - \frac{1}{1 + e^{-w_p/h_p}})^2 \\ \alpha &= \frac{\nu}{(1 - IoU) + \nu} \end{aligned} \tag{6}$$

The detection results in following section show that our proposed Sig-IoU loss is superior to original CIoU loss.

TABLE 5. Experimental configuration.

Parameter	Configuration
CPU	Intel(R) Xeon(R) Gold 5320
GPU	NVIDIA RTX A4000 (16 GB)
Operation System	Ubuntu 20
torch	1.10.0
Python	3.8.10
CUDA	11.3
Pre-trained weight	No
Batch size	32
Input size	640 × 640 pixels
Optimizer	SGD
Learning rate	0.01
Weight decay	0.0005
Momentum	0.937
Epochs	100

TABLE 6. Confusion matrix.

Ground truth \ Prediction	True	False
True	True positive (TP)	False negative (FN)
False	False positive (FP)	True negative (TN)

IV. RESULTS AND DISCUSSIONS

A. EXPERIMENTAL CONFIGURATION

The detailed configuration used for training and verification experiments are tabulated in Table 5.

B. EVALUATION METRICS

Based on confusion matrix shown in Table 6, precision and recall are calculated to evaluate our proposed model. The values of precision and recall can be calculated by Equation 7.

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \end{aligned} \tag{7}$$

TABLE 7. Ablation results.

Models	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Detection time per image (ms)	FPS
YOLOv8-nano	85.2%	80.1%	85.8%	73.9%	8.8	113
+ C2Focal	85.3%	83.8%	87.2%	74.5%	9.6	104
+ Sig-IoU loss	84.4%	**85.3%**	87.5%	74.7%	9.5	105
+ Ghost Conv	**89.7%**	83.5%	**87.7%**	**75.3%**	9.1	110

In this paper, average precision (AP) is used to obtain a much sounder evaluation. The calculation of AP is formulated as

$$AP = \int_0^1 p(r)dr \tag{8}$$

where p represents precision and r means recall. Then, mAP is calculated based on different IoU thresholds.

C. EXPERIMENT RESULTS

1) ABLATION RESULTS

Ablation experiments for the improvements of Ghost convolution module, C2Focal module and Sig-IoU loss are executed to validate their benefits. We set YOLOv8-nano as the baseline to make the ablation. To test the performances of our proposed modifications, 1122 PCB component images are detected. We present the ablation results in Table 7. It indicates that our proposed model is much superior to original YOLOv8-nano on the task of diverse PCB component detection.

The introduction of C2Focal modules to replace the original C2f modules contributes to apparent boost to the detection metrics. As shown in Table 7, the recall metric markedly increases by 3.7% compared with the baseline, although the precision value has no evident enhancement yet. The metrics of mAP@0.5 and mAP@0.5:0.95 have significant increases of 1.4% and 0.6% respectively. The improvements in detection performances profit from the perception that FocalNeXt block can enlarge the receptive field of neurons and integrate fine-grained local and coarse-grained global features. Thus, the C2Focal module integrating FocalNeXt block mitigates the adverse impact from the scale variance to achieve preferable results. Moreover, only introducing little additional computation cost, the model with C2Focal modules still reaches the detection speed of 104 frames per second (FPS), which meets the requirement of industrial application.

Then, we use the Sig-IoU loss function for bounding box regression. As shown in Table 7, the introduction of Sig-IoU loss shows a little decrease in the precision. However, the recall is observably increased to 85.3% with an increment of 1.5%. The detection metrics of mAP@0.5 and mAP@0.5:0.95 also have certain rises of 0.3% and 0.2% respectively. As discussed previously, there are plenty of tiny components in our dataset, easily causing the gradient explosion to affect the converge of the model. The introduction of Sigmoid function into the loss copes with the gradient explosion issue successfully, which makes the model converge faster. Fig. 13 presents the convergence curves of CIoU and Sig-IoU loss for box regression. It demonstrates that our proposed Sig-IoU loss function reduces faster and lower, making prediction boxes approach ground truth quickly.

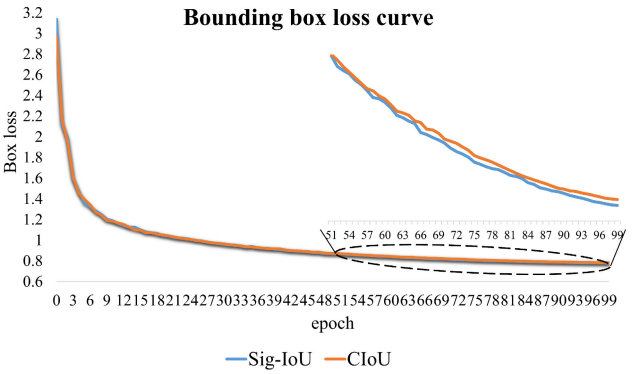

FIGURE 13. The bounding box loss curve.

After involving the Ghost convolutions, the detection performances of our model have marked enhancements too. The detection precision is significantly increased by 5.3% to reach 89.7%. Although the recall is reduced by 1.8%, it is still much larger than that of the baseline. The values of mAP@0.5 and mAP@0.5:0.95 generate the evident enhancements of 0.3% and 0.7%, respectively. Furthermore, Ghost convolution is able to generate equivalent feature maps using less parameters compared with common convolution, theoretically speeding up the detection process. Therefore, the inference speed has been increased to 110 FPS, which approaches nearly to the speed of 113 FPS obtained by original YOLOv8-nano.

Eventually, our proposed model significantly outperforms the YOLOv8-nano in the metrics of precision, recall and mAP. The values of precision and recall achieved by our method get 89.7% and 83.5% respectively, which exist 4.5% and 3.4% enhancements than that by YOLOv8-nano. The detection mAP@0.5 and mAP@0.5:0.95 are 87.7% and 75.3%, providing 1.9% and 1.4% growth separately than the baseline. When applying to the validation set, our method

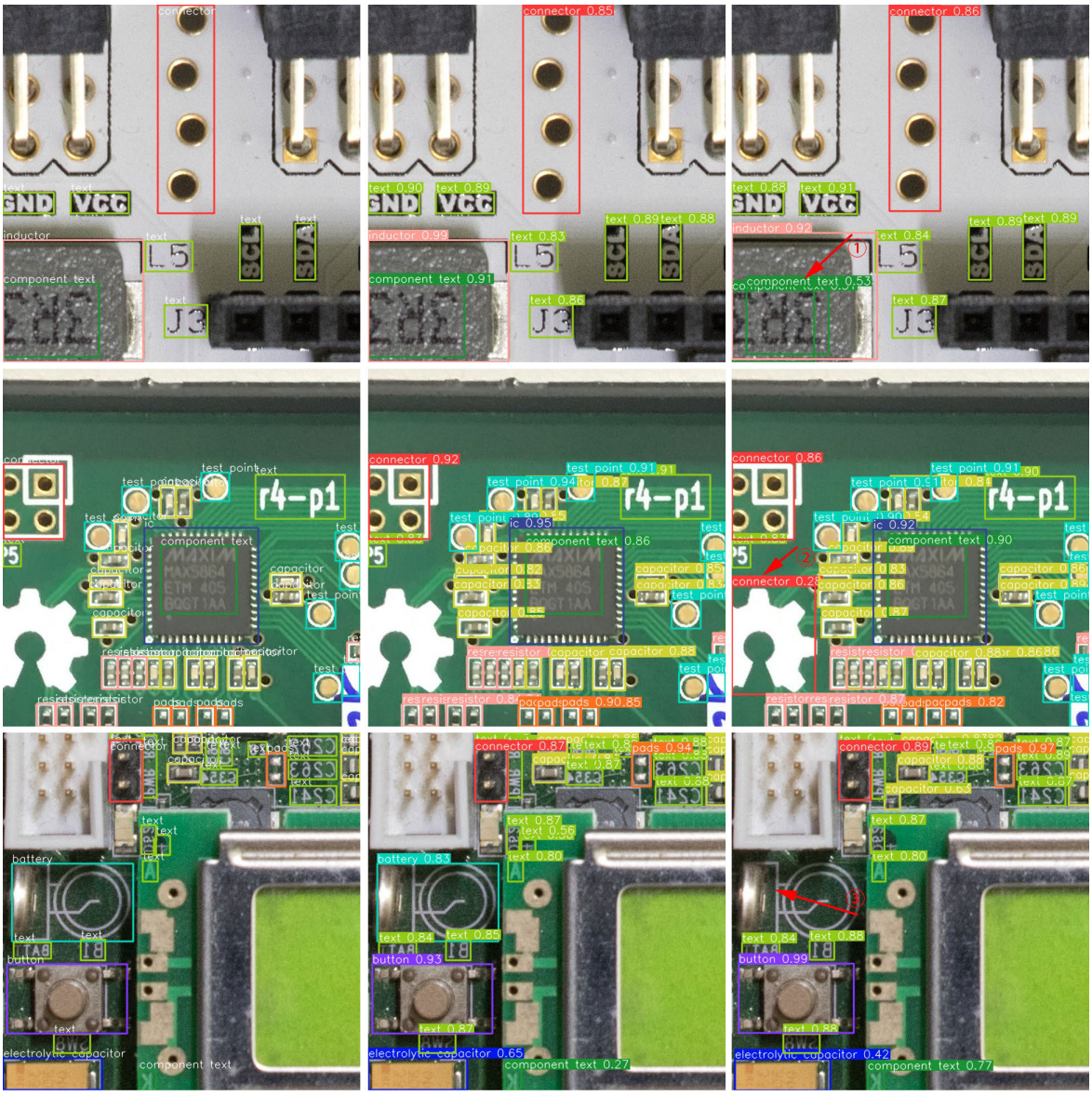

FIGURE 14. The detection examples by different models, (a) ground truths (b) results by our proposed model, (c) results by YOLOv8-nano (Note: arrow 1 indicates redundant prediction box, arrow 2 indicates misjudging the background as a connector, arrow 3 represents missing battery).

achieves 85.7% mAP@0.5 and 72.8% mAP@0.5:0.95, respectively. In addition, our model is lighter, only containing 2.6 million parameters, which is decreased by 19% compared with the baseline. Thus, our model can finish detecting one image within 9.1 ms averagely, that is 110 FPS. The inference speed of our proposed model is approximately comparable to the speed of the YOLOv8-nano, meeting the demand for industrial application.

The examples of PCB component detection results are illustrated in Fig. 14, verifying the advancement of our model. It indicates that YOLOv8-nano leaves redundant prediction boxes for the class of component text. Additionally, it misses the battery component or misjudges the background as a connector. In contrast, our proposed model avoids all the previous failures. Our model detects the densely distributed resistors, capacitors, texts etc. with high confidence scores, demonstrating the potential of our proposed improvements in the model.

2) PERFORMANCE COMPARISON RESULTS

We compare the performances between our model and other SOTAs, such as Faster R-CNN [17], YOLOv3-tiny [23], YOLOv4-tiny [31], YOLOv5-small [42], YOLOX-tiny [44], YOLOv7-tiny [47] and YOLOv8-nano [25]. The comparison metrics include the amount of model parameters, mAP and detection speed.

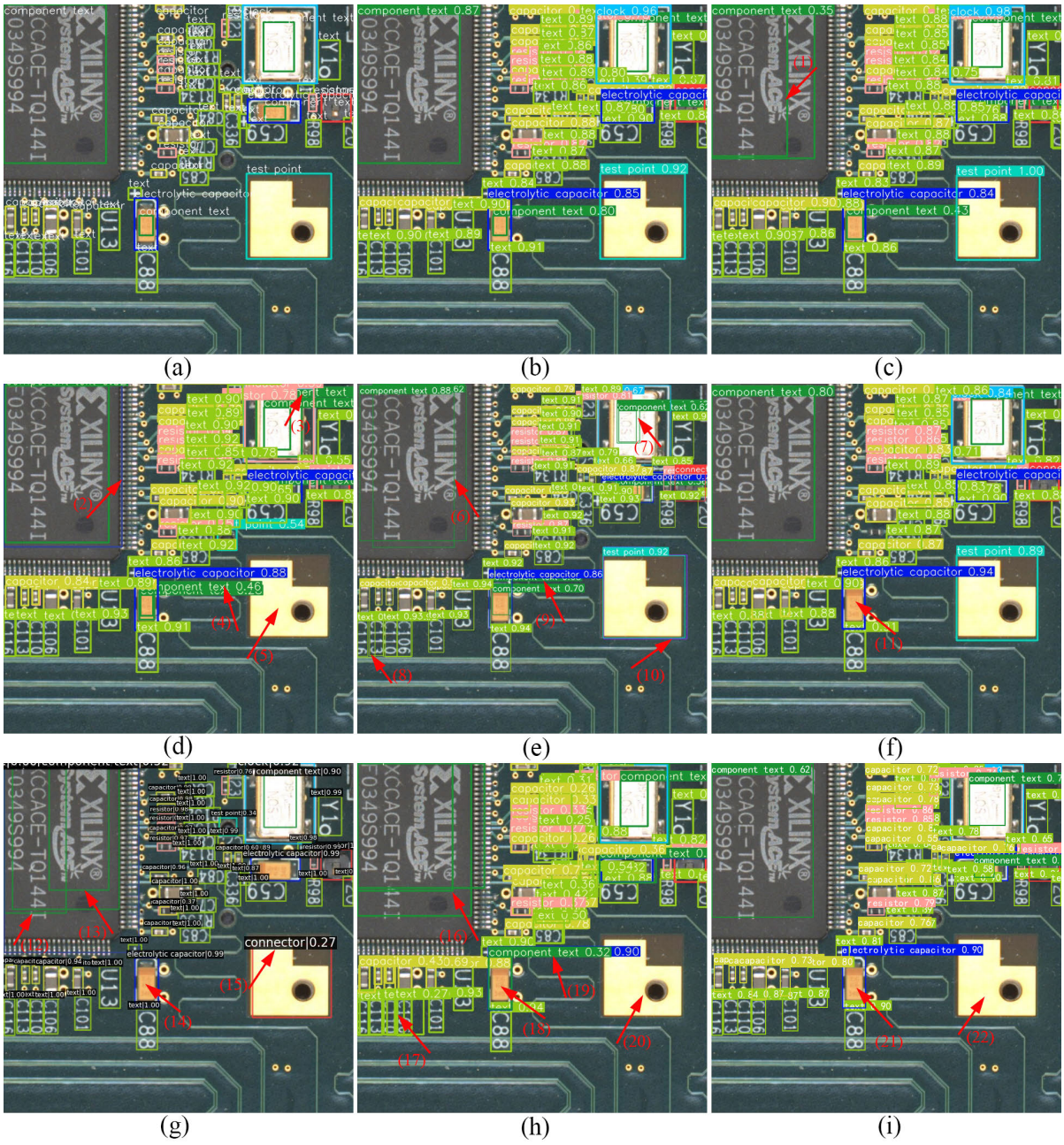

FIGURE 15. Detection results of PCB components by different methods, (a) ground truths, (b) the proposed model, (c) YOLOv8-nano, (d) YOLOv7-tiny, (e) YOLOv5-small, (f) YOLOv3-tiny, (g) Faster RCNN, (h) YOLOv4-tiny, (i) YOLOX-tiny.

According to the presentation in Table 8, our proposed model significantly outperforms other benchmarks, achieving the highest 87.7% mAP@0.5 and 75.3% mAP@0.5:0.95. Faster RCNN, as the noted two-stage CNN model, only obtains 64.9% mAP@0.5 and 49.71% mAP@0.5:0.95, which deeply falls behind our model. And it has vast parameters, leading to an extremely large latency. The inference speed of Faster RCNN is almost 10 times slower than our model so that it cannot execute real-time detection.

YOLO series are known for simultaneous high-speed and precise detection. Here, all of YOLOv3-tiny, YOLOv5-small and YOLOv8-nano achieve good performances. YOLOv7-tiny, YOLOv4-tiny and YOLOX-tiny perform poorly on our PCB component dataset, while YOLOv7-tiny and YOLOv4-tiny take the lead in detection speed. Concerning our proposed model, it undoubtedly exceeds other YOLO series in detection precision. Furthermore, benefiting by the introduction of Ghost convolution and C2Focal module, our model has quite limited amount of parameters, that is 2.6 million only. Although our model is not the fastest one, the inference speed of 110 FPS has been achieved, which is potential to be applied in the real-time PCB inspection industry.

TABLE 8. Comparison results between our model and other SOTAs.

Models	Parameters (M)	mAP@0.5	mAP@0.5:0.95	FPS
Faster RCNN	41.15	64.90%	49.71%	12
YOLOv3-tiny	12.15	85.7%	73.4%	82
YOLOv4-tiny	6.36	62.9%	45.4%	149
YOLOv5-small	7.10	78.6%	63.7%	104
YOLOX-tiny	5.06	59.48%	42.4%	92
YOLOv7-tiny	6.02	52.7%	38.1%	130
YOLOv8-nano	3.20	85.8%	73.9%	113
Our proposed model	**2.60**	**87.7%**	**75.3%**	110

TABLE 9. Explanations of bad predictions in Fig. 15.

Models	Red Arrow	Explanation
YOLOv8-nano	(1)	superfluous prediction box for component text
YOLOv7-tiny	(2) (3) (4) (5)	superfluous prediction boxes for component text missing the test point
YOLOv5-small	(6) (7) (8) (9) (10)	superfluous prediction boxes for component text or text superfluous wrong prediction box for test point
YOLOv3-tiny	(11)	missing component text
Faster RCNN	(12) (13) (14) (15)	superfluous prediction boxes for component text missing component text wrong prediction for test point
YOLOv4-tiny	(16) (17) (18) (19) (20)	superfluous prediction boxes for component text or text missing component text wrong prediction for electrolytic capacitor missing test point
YOLOX-tiny	(21) (22)	missing component text missing test point

Fig. 15 compares the detection outputs produced by our proposed method and other SOTAs. None of Faster R-CNN and other YOLO series succeeds in detecting all the targets without errors. As shown in Fig. 15, all of YOLOv8-nano, YOLOv7-tiny, YOLOv4-tiny and YOLOv5-small produce superfluous prediction boxes for component text. Moreover, YOLOv7-tiny, YOLOv4-tiny and YOLOX-tiny miss the test point target sometimes. YOLOv5-small correctly labels the test point, but it also produces a false detection simultaneously that the test point is labeled as a button. For the small component text on the electrolytic capacitor, YOLOv3-tiny, YOLOv4-tiny and YOLOX-tiny have the trouble in finding out it. In addition, YOLOv4-tiny incorrectly labels the electrolytic capacitor as component text. Similarly, Faster R-CNN generates redundant bounding boxes or misses some targets sometimes. Worse than that, Faster R-CNN detects the test point as a connector incorrectly. For better understanding, the bad predictions in Fig. 15 are pointed out by red arrows, and the detailed explanations are tabulated in Table 9.

In contrast, profiting from the proposed refinements such as C2Focal module, Ghost convolution module and advanced bounding box regression loss, our method identifies all the PCB components favorably with notable confidences.

V. CONCLUSION

A creative deep learning model is proposed to detect densely packed PCB components efficiently and precisely. The proposed network beats conventional works, which are weighty in parameter and computation, and prolonged in detection time. A couple of improvements exist in our method. We introduce the FocalNeXt block to design the C2Focal module, combining both fine-grained local and coarse-grained global features concurrently. We replace the common convolution by the lightweight Ghost convolution to reduce the computation effectively without affecting the detection precision of the model. Besides, the bounding box regression loss function is upgraded to Sig-IoU loss, enhancing the localization accuracy and accelerating the convergence of the model.

Our model attains the highest detection precisions of 87.7% mAP@0.5 and 75.3% mAP@0.5:0.95 on our PCB component dataset. In addition, our model is very light in the amount of parameters. Furthermore, our model earns the detection speed of 110 FPS, which is several times faster than Faster RCNN. Thus, it is potential to be implemented in the real-time and high-precision PCB inspection industry. We believe our proposed method will boost the zero-error quality control for PCBs, plunging the production cost significantly.

In this paper, the proposed method not only pursues the precision, but also pays attention to the detection speed. High inference speed makes the proposed method possible for the real application in the PCB production line. Moreover, the proposed method outperforms other SOTAs in detection precision, demonstrating the effectiveness of out proposed modules and improvements. And our method detects 32 classes of dense PCB components, infinitely superior than other works. However, so many classes of components with diverse appearances and dimensions possibly make the accuracy not extremely high. Another limitation is the imbalanced PCB component dataset. Some classes of components are infrequent so that the detection performances for them are relatively poor. In addition, the policy that merges all the silkscreen numbers of electronic components in one class could also cause the data imbalance.

Therefore, future works could focus on the detection improvements for imbalanced PCB components. A definitely potential direction is improving the strategy for grouping 'text' to make the instance distribution more reasonable. Fox example, only these kinds of text instances with the same types of components can be merged in one class, such as text_connector, text_resistor etc. Another possible direction for improving the detection performance is to involve the angles of PCB components using arbitrary-oriented detectors. The regression of angular prediction can be transformed to a classification task [48], [49], [50], as small number of components are placed on the board with $\pm 45°$ angles only.

In addition, based on the achievements of this paper, PCB component defects detection or pixel-level segmentation could be developed further. Moreover, the proposed method is potential to be used for augmented reality application in factories. For example, the recognition information of PCB components can be projected in the user's field of view for better understanding of PCB assembly construction.

REFERENCES

[1] Y. Shi, Z. Xin, P. C. Loh, and F. Blaabjerg, "A review of traditional helical to recent miniaturized printed circuit board Rogowski coils for power-electronic applications," *IEEE Trans. Power Electron.*, vol. 35, no. 11, pp. 12207–12222, Nov. 2020.

[2] F. R. Leta and F. F. Feliciano, "Computational system to detect defects in mounted and bare PCB based on connectivity and image correlation," in *Proc. 15th Int. Conf. Syst., Signals Image Process.*, Jun. 2008, pp. 331–334.

[3] G. Qiang, Z. Shanshan, Z. Yang, and C. Mao, "Detection method of PCB component based on automatic optical stitching algorithm," *Circuit World*, vol. 41, no. 4, pp. 133–136, Nov. 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:62337549

[4] S.-C. Lin, C.-H. Chou, and C.-H. Su, "A development of visual inspection system for surface mounted devices on printed circuit board," in *Proc. IECON 33rd Annu. Conf. IEEE Ind. Electron. Soc.*, Nov. 2007, pp. 2440–2445.

[5] A. J. Crispin and V. Rankov, "Automated inspection of PCB components using a genetic algorithm template-matching approach," *Int. J. Adv. Manuf. Technol.*, vol. 35, nos. 3–4, pp. 293–300, Dec. 2007. [Online]. Available: https://api.semanticscholar.org/CorpusID:109271957

[6] S. Mashohor, J. R. Evans, and A. T. Erdogan, "Automatic hybrid genetic algorithm based printed circuit board inspection," in *Proc. 1st NASA/ESA Conf. Adapt. Hardw. Syst. (AHS)*, 2006, pp. 390–400.

[7] D. Li, C. Li, C. Chen, and Z. Zhao, "Semantic segmentation of a printed circuit board for component recognition based on depth images," *Sensors*, vol. 20, no. 18, p. 5318, Sep. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221843812

[8] H. M. Yin, "A template-matching-based fast algorithm for PCB components detection," *Adv. Mater. Res.*, vols. 690–693, pp. 3205–3208, May 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:60677124

[9] D.-Z. Wang, C.-H. Wu, A. Ip, C.-Y. Chan, and D.-W. Wang, "Fast multi-template matching using a particle swarm optimization algorithm for PCB inspection," in *Applications of Evolutionary Computing*. Berlin, Germany: Springer, 2008, pp. 365–370.

[10] N. Dong, C.-H. Wu, W.-H. Ip, Z.-Q. Chen, and K.-L. Yung, "Chaotic species based particle swarm optimization algorithms and its application in PCB components detection," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12501–12511, Nov. 2012.

[11] L. Chen, A. H. Sadka, J. Dong, and H. Zhou, "Object detection based on CNNs: Current and future directions," in *Recent Trends in Computer Applications*. Cham, Switzerland: Springer, 2018, pp. 17–32.

[12] K. Demir, M. Ay, M. Cavas, and F. Demir, "Automated steel surface defect detection and classification using a new deep learning-based approach," *Neural Comput. Appl.*, vol. 35, no. 11, pp. 8389–8406, Apr. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:254494333

[13] O. D. Pedrayes, D. G. Lema, R. Usamentiaga, and D. F. García, "Detection and localization of fugitive emissions in industrial plants using surveillance cameras," *Comput. Ind.*, vol. 142, Nov. 2022, Art. no. 103731. [Online]. Available: https://api.semanticscholar.org/CorpusID:249933653

[14] D. Kang, S. S. Benipal, D. L. Gopal, and Y.-J. Cha, "Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning," *Autom. Construct.*, vol. 118, Oct. 2020, Art. no. 103291. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580520300157

[15] R. Ali and Y.-J. Cha, "Attention-based generative adversarial network with internal damage segmentation using thermography," *Autom. Construction*, vol. 141, Sep. 2022, Art. no. 104412. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580522002850

[16] J. Lewis, Y.-J. Cha, and J. Kim, "Dual encoder–decoder-based deep polyp segmentation network for colonoscopy images," *Sci. Rep.*, vol. 13, no. 1, p. 1183, Jan. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:256080147

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[18] J. Li, J. Gu, Z. Huang, and J. Wen, "Application research of improved YOLO v3 algorithm in PCB electronic component detection," *Appl. Sci.*, vol. 9, no. 18, p. 3750, Sep. 2019.

[19] J. Shen, N. Liu, and H. Sun, "Defect detection of printed circuit board based on lightweight deep convolution network," *IET Image Process.*, vol. 14, no. 15, pp. 3932–3940, Dec. 2020.

[20] K. Zhang, "Using deep learning to automatic inspection system of printed circuit board in manufacturing industry under the Internet of Things," *Comput. Sci. Inf. Syst.*, vol. 20, no. 2, pp. 723–741, 2023.

[21] I.-C. Chen, R.-C. Hwang, and H.-C. Huang, "PCB defect detection based on deep learning algorithm," *Processes*, vol. 11, no. 3, p. 775, Mar. 2023.

[22] X. Liu, J. Hu, H. Wang, Z. Zhang, X. Lu, C. Sheng, S. Song, and J. Nie, "Gaussian-IoU loss: Better learning for bounding box regression on PCB component detection," *Expert Syst. Appl.*, vol. 190, Mar. 2022, Art. no. 116178. [Online]. Available: https://api.semanticscholar.org/CorpusID:244096237

[23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[24] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2018, pp. 1967–1976.

[25] Ultralytics. (2023). *YOLOv8*. Accessed: Aug. 10, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[26] G. Zhang, Z. Li, J. Li, and X. Hu, "CFNet: Cascade fusion network for dense prediction," 2023, *arXiv:2302.06052*.

[27] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[29] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8574–8586, Aug. 2022.

[30] J. S. S. V. Mamidi, S. Sameer, and J. Bayana, "A light weight version of PCB defect detection system using YOLO v4 tiny," in *Proc. Int. Mobile Embedded Technol. Conf. (MECON)*, Mar. 2022, pp. 441–445.

[31] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[32] Z. Lan, Y. Hong, and Y. Li, "An improved YOLOv3 method for PCB surface defect detection," in *Proc. IEEE Int. Conf. Power Electron., Comput. Appl. (ICPECA)*, Jan. 2021, pp. 1009–1015.

[33] C.-J. Li, Z. Qu, S.-Y. Wang, K.-H. Bao, and S.-Y. Wang, "A method of defect detection for focal hard samples PCB based on extended FPN model," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 12, no. 2, pp. 217–227, Feb. 2022.

[34] A. Evangelidis, N. Dimitriou, L. Leontaris, D. Ioannidis, G. Tinker, and D. Tzovaras, "A deep regression framework toward laboratory accuracy in the shop floor of microelectronics," *IEEE Trans. Ind. Informat.*, vol. 19, no. 3, pp. 2652–2661, Mar. 2023.

[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[36] C.-W. Kuo, J. D. Ashmore, D. Huggins, and Z. Kira, "Data-efficient graph embedding learning for PCB component detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 551–560.

[37] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[38] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, Nov. 2018.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[40] C.-Y. Wang, H. Liao, and I.-H. Yeh, "Designing network design strategies through gradient path analysis," *J. Inf. Sci. Eng.*, vol. 39, pp. 975–995, Nov. 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253420213

[41] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.

[42] Ultralytics. (2020). *YOLOv5*. Accessed: Aug. 10, 2023. [Online]. Available: https://github.com/ultralytics/yoloV5

[43] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[44] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[45] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. European Conf. Comput. Vis.*, 2014, pp. 740–755. [Online]. Available: https://api.semanticscholar.org/CorpusID:14113767
[46] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11966–11976.
[47] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.
[48] Z. Sun, X. Leng, Y. Lei, B. Xiong, K. Ji, and G. Kuang, "BiFA-YOLO: A novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images," *Remote Sens.*, vol. 13, no. 21, p. 4209, Oct. 2021.
[49] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," in *Proc. Eur. Conf. Comput. Vis.*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 677–694.
[50] X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan, "Dense label encoding for boundary discontinuity free rotation detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15814–15824.

QIN LING received the B.Eng. degree in mechanical engineering and automation, in 2013, and the master's degree in mechatronics engineering, in 2022. He is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, Universiti Sains Malaysia (USM). His research interests include image processing, deep learning, convolution neural networks, and artificial intelligence.

NOR ASHIDI MAT ISA received the B.Eng. degree (Hons.) in electrical and electronic engineering and the Ph.D. degree in electronic engineering (majoring in image processing and artificial neural network) from Universiti Sains Malaysia (USM), in 1999 and 2003, respectively. He is currently a Professor with the School of Electrical and Electronic Engineering, USM. He has published more than 188, 222, and 300 articles indexed in WoS-ISI (H-index 32), SCOPUS (H-index 38), and Google Scholar (H-Index 46), respectively. His research interests include intelligent systems, image processing, neural networks, computational intelligence, and medical image processing. Due to his outstanding achievement in research, he gained recognition, both national and internationally. He was recognized as a top 2% researcher in category—Citation Impact in Single Calendar Years 2019, 2020, and 2021, by Stanford University, USA, and Top Research Scientist Malaysia (TRSM) by Akademi Sains Malaysia (ASM), in 2020.

MOHD SHAHRIMIE MOHD ASAARI received the Bachelor of Engineering degree (Hons.) in electrical from Universiti Teknologi Mara, in 2009, the Master of Science degree in electrical and electronics from Universiti Sains Malaysia, in 2012, and the Ph.D. degree in science (physics) from Universiteit Antwerpen, Belgium, in 2019. In his Ph.D. research, he concentrates in plant-related study, where the project aims for early stress detection in plants using the analysis of close-range hyperspectral imaging. Currently, he is a Senior Lecturer with the School of Electrical and Electronics Engineering, Universiti Sains Malaysia. His research interests include image processing, computer vision, machine learning, and close-range hyperspectral imaging.