IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 November 2023, accepted 26 November 2023, date of publication 5 December 2023,
date of current version 14 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339561

TOPICAL REVIEW

A Comprehensive Review of Deep Learning-Based PCB Defect Detection

XING CHEN[1], **YONGLEI WU**[2], **XINGYOU HE**[2], **AND WUYI MING**[3]

[1]School of Artificial Intelligence and Software Engineering, Nanyang Normal University, Nanyang 473061, China
[2]Henan Key Laboratory of Intelligent Manufacturing of Mechanical Equipment, Zhengzhou University of Light Industry, Zhengzhou 450002, China
[3]Guangdong Provincial Key Laboratory of Digital Manufacturing Equipment, Guangdong HUST Industrial Technology Research Institute, Dongguan 523808, China

Corresponding author: Wuyi Ming (mingwuyi@gmail.com)

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515140066, in part by the Henan Provincial Key Scientific and Technological Project 222102220011, and in part by the Guangdong Provincial Key Laboratory of Manufacturing Equipment Digitization under Grant 2023B1212060012.

ABSTRACT A printed circuit board (PCB) functions as a substrate essential for interconnecting and securing electronic components. Its widespread integration is evident in modern electronic devices, spanning computers, cell phones, televisions, digital cameras, and diverse apparatus. Ensuring product quality mandates meticulous defect inspection, a task exacerbated by the heightened precision of contemporary circuit boards, intensifying the challenge of defect detection. Conventional algorithms, hampered by inefficiency and limited accuracy, fall short of usage benchmarks. In contrast, PCB defect detection algorithms rooted in deep learning hold promise for achieving heightened accuracy and efficiency, bolstered by their adeptness at discerning novel defect types. This review presents a comprehensive analysis of machine vision-based PCB defect detection algorithms, traversing the realms of machine learning and deep learning. It commences by contextualizing and elucidating the significance of such algorithms, followed by an extensive exploration of their evolution within the machine vision framework, encompassing classification, comparison, and analysis of algorithmic principles, strengths, and weaknesses. Moreover, the introduction of widely used PCB defect detection datasets and assessment indices enhances the evaluation of algorithmic performance. Currently, the detection accuracy can exceed 95% at an Intersection over Union (IoU) of 0.5. Lastly, potential future research directions are identified to address the existing issues in the current algorithm. These directions include utilizing Transformers as a foundational framework for creating new algorithms and employing techniques like Generative Adversarial Networks (GANs) and reinforcement learning to enhance PCB defect detection performance.

INDEX TERMS Deep learning, defect detection, machine vision, neural networks, PCB.

I. INTRODUCTION

In recent years, electronic products have witnessed widespread utilization and popularity, driven by the rapid expansion of the global economy and the swift evolution of information technology [1], [2]. With continuous upgrades and enhanced functionalities of electronic products, the requisites for performance and quality in printed circuit boards (PCBs) are on the rise [3]. Functioning as foundational underpinnings and pivotal components within electronic products [4], PCBs must demonstrate robust stability, pronounced resistance to interference, and excel in characteristics encompassing high-speed transmission, heightened integration levels, and compact dimensions [5]. Furthermore, PCB layout encompasses the strategic arrangement of PCBs, constituting a critical stage in electronic device manufacturing. Key concepts in the PCB layout process include:

1) Layout: This involves planning and organizing the placement of electronic components, connection lines, holes, and other specific elements on a PCB. The quality of the

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

layout directly influences circuit performance, electromagnetic interference (EMI), thermal performance, and maintenance challenges.

2) Component placement: An essential phase in the layout process, which includes the positioning of electronic components (such as chips, resistors, capacitors, connectors, etc.) on the PCB. Proper component placement is integral to ensuring optimal circuit performance and signal transmission.

3) Routing: Upon completion of the layout, the wiring routes must be established to interconnect the various components and form the circuit. Routing requires consideration of factors like signal integrity, power supply, ground, and signal transmission lines.

4) Power distribution: The layout process must include a rational plan for the distribution of power and ground lines to ensure that electronic components receive a stable power supply and to minimize noise and interference within the circuit.

Therefore, PCB layout is a pivotal step in the design and manufacture of electronic products. It demands that engineers consider multiple factors concurrently—performance, heat dissipation, and maintainability—to ensure the rational layout and connection of electronic components. This careful planning ensures that the PCB meets the design requirements and facilitates smooth manufacturing. Hence, ensuring defect detection and implementing quality control measures for PCBs assume paramount significance in the domain of PCB manufacturing and production processes [6].

Nevertheless, due to the diminishing size of components, amplified component density, and intricate and diverse nature of PCB manufacturing, PCBs become susceptible to a range of factors, including mechanical friction, electrostatic interference, and chemical corrosion, during production. These factors can readily give rise to a spectrum of defects, such as missing holes, mouse bites, open circuits, shorts, spurs, spurious copper, and broken holes [1], [7]. Figure 1 illustrates a variety of PCB defect types. These defects substantially undermine the quality and performance of PCBs. Failure to detect and address these issues promptly can directly impede the regular operation and service life of electronic devices, potentially resulting in severe accidents and incidents. Consequently, swiftly and accurately detecting and locating PCB defects during the manufacturing process holds paramount significance.

Presently, the detection of PCB defects stands as a crucial concern within the electronics manufacturing sector. Various inspection methods have been researched and developed to address diverse PCB defects and align with the demands of industrial production. These include the functional test method, visual inspection technology, instrument on-line inspection method, and manual visual subjective determination method [9], [10], [11]. Among these, the functional test method involves utilizing a fault simulator to assess the circuit board's functionality and identify potential defects. Although offering commendable reliability and accuracy, this approach necessitates specialized equipment and intricate testing procedures, resulting in a laborious and time-intensive process. Visual inspection technology primarily relies on artificial intelligence algorithms, utilizing image processing and pattern recognition techniques for swift and automated identification of PCB defects [12]. However, its effectiveness relies on ongoing optimization and enhancement of the detection algorithm. The instrument on-line detection method serves primarily to oversee PCB defects via various instruments, including the high-voltage detection and insulation hydrocarbon detection methods. This approach offers the benefits of convenient operation and rapid detection but occasionally faces challenges of misjudgment. The artificial visual subjective judgment method predominantly hinges on experiential and perceptual assessment, rendering it susceptible to human influence; nevertheless, it finds applicability in certain specialized inspection scenarios.

Despite the array of PCB defect detection methods, each approach presents inherent limitations and deficiencies. The selection of the suitable method must align with specific requirements and circumstances. In recent years, propelled by the continual progress and integration of technology, deep learning-based algorithms have progressively gained prominence across diverse industries. Within the industrial sector, these algorithms find utility in processing and positioning tasks [13], [14], while also capable of identifying defects in 3C products [15], [16], [17]. Similarly, within the medical field, deep learning-based algorithms aid doctors in analyzing medical images for diagnostic support [18], [19]. In agriculture, these algorithms contribute to crop monitoring [20], [21], [22], [23]. Additionally, in the military domain, deep learning-based algorithms analyze remote sensing images to facilitate swift positioning [24], [25], [26]. Simultaneously, PCB defect detection algorithms based on deep learning are poised to emerge as pivotal tools within the electronic manufacturing industry, offering a broader scope of application and developmental opportunities.

A standard PCB typically comprises several distinct layers:

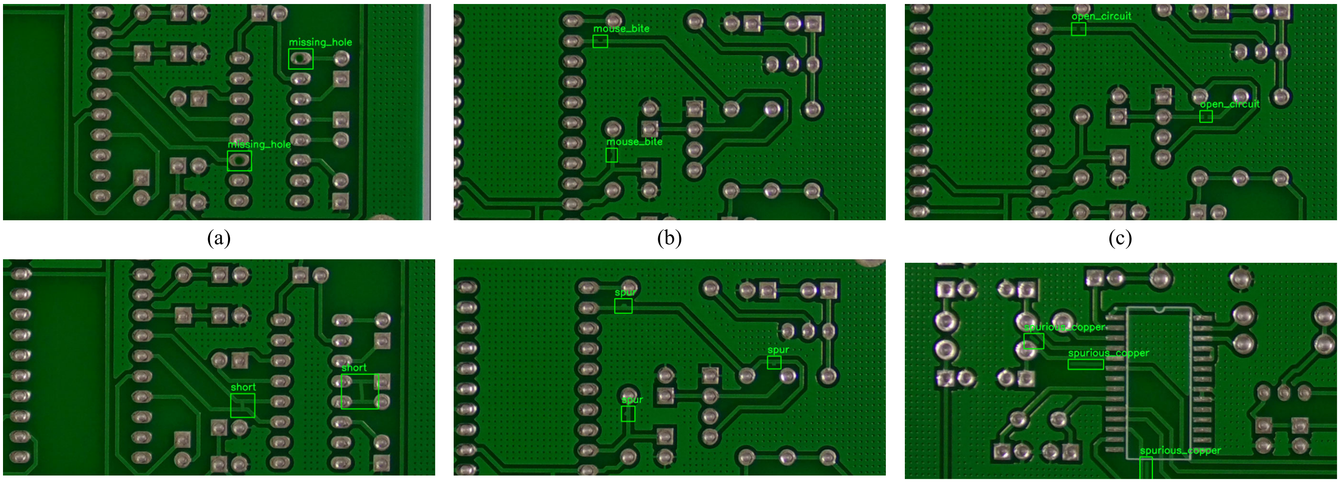
1) Bottom copper layer: This foundational metal layer provides ground connections and layouts for various circuit components. It frequently includes a soldermask layer to safeguard the circuit traces and a copper layer.

2) Top copper layer: This uppermost metal layer hosts the primary circuit components, wires, and signal paths, often including pads for connecting electronic elements.

3) Inner layers: Beyond the bottom and top copper layers, a PCB may feature one or more inner layers composed of glass fiber and copper foil, which facilitate signal or power transmission.

4) Signal layers: These layers transmit signals between electronic components, encompassing data, control signals, and clock signals.

5) Power layers (power planes): These layers supply power connections, including power and ground, typically



(a) (b) (c)

(d) (e) (f)

FIGURE 1. Variety of PCB defect types: (a) Missing hole [8]; (b) Mouse bite [8]; (c) Open circuit [8]; (d) Short [8]; (e) Spur [8]; (f) Spurious copper [8].

distributing power and ground connections to ensure board power stability.

6) Ground layer (ground planes): This dedicated layer provides a stable ground connection to minimize signal interference and enhance circuit performance.

7) Pad layer: Positioned on top of the top and bottom copper layers, the solder mask layer protects the circuit traces and copper layer to prevent short circuits, typically featuring pad openings to allow electronic component soldering.

8) Silkscreen layer: This layer commonly includes labels, component pin markings, and additional component information to aid assembly and maintenance personnel in correctly identifying and handling electronic components.

These layers' configurations can differ according to PCB design requirements. As this review focuses on deep learning-based PCB defect detection—predominantly based on machine vision applications—defect detection usually pertains to surface layers, including:

1) Top copper layer: Here, the deep learning model can identify pad defects, such as false soldering, short circuits, open circuits, offsets, and other issues.

2) Bottom copper layer: This layer also accommodates the layout of pads and electronic components, with defect detection similar to the top copper layer.

3) Pad layer: In this layer, the deep learning model can detect defects within the pad opening area.

4) Silkscreen layer: This layer, containing information about component markings and pin markings, allows deep learning models to detect issues such as broken and shifted lettering.

Moreover, some PCBs may incorporate specialized layers like soldermasks, inner layers, etc., which could be used for detecting specific defects.

This review delivers a methodical analysis and thorough synthesis of the research literature concerning PCB defect detection within the last decade. Its primary focus rests on methods and algorithms employing deep learning techniques to enhance the efficacy of PCB defect detection. The efficacy and impact of these algorithms in practical applications are expounded in detail. Additionally, a thorough elucidation of the fundamental principles underpinning deep learning and a concise overview of the Transformer model is furnished to ensure readers attain a comprehensive grasp of the foundational aspects of the domain. Moreover, this paper introduces widely employed datasets and evaluation metrics utilized for PCB defect detection. Ultimately, drawing upon the extant literature, prevailing algorithms are scrutinized and deliberated upon, while potential developmental pathways within this realm are postulated.

II. FUNDAMENTALS OF DEEP LEARNING

A. BASIC KNOWLEDGE

Coined in 1986, the term deep learning (DL) initially found its place within the realm of machine learning, later extending to the domain of artificial neural networks in 2000. Deep neural networks are structured with numerous hidden layers that progressively abstract data features [27], [28]. This architecture empowers computers to autonomously acquire higher-level abstract features, thus addressing a gamut of tasks such as classification, regression, clustering, and generation [29], [30]. In contrast to conventional machine learning methods, deep learning dispenses with the necessity for pre-designed manual features, automating the process of feature extraction and learning through extensive data training. An inherent advantage of deep learning lies in its efficiency in processing voluminous datasets, culminating in remarkable outcomes when presented with ample data. Nonetheless, challenges persist, encompassing the requisition for substantial labeled data, the expenditure of costly computational resources, and the interpretability of models. Consequently, the training process assumes paramount significance, especially in applications like PCB defect detection. This phase necessitates data preprocessing, the construction of a problem-relevant network model, and the formulation of the corresponding loss function. Lately, the strides achieved in the realm of deep learning predominantly owe their origins to pioneering advancements in network models, loss functions, and activation functions. As technology continues its relentless evolution, deep learning is poised to embrace a wider scope of developmental opportunities in the future, positioning itself as an integral component of the artificial intelligence landscape. Subsequently, attention will be directed towards convolutional neural networks (CNNs) and the recently surpassing Transformer architecture. These network structures have demonstrated remarkable accomplishments across varied domains, thereby unveiling novel avenues for the proliferation of deep learning applications.

B. CONVOLUTIONAL NEURAL NETWORK

CNN [31] is a distinct neural network architecture that finds extensive application in processing various forms of data, including images, speech, and natural language. Key constituents within CNN encompass the convolutional layer, pooling layer, and fully connected layer. Skillful integration of these layers enables the processing of data and extraction of features. CNNs have found widespread application within the realm of computer vision (CV), undertaking diverse tasks like image classification, target detection, semantic segmentation, image caricature, style migration, and generative adversarial networks (GANs) [32]. These applications have yielded outstanding outcomes. Simultaneously, the network architecture of CNN has undergone continual refinement and optimization, leading to the emergence of several exemplary structures, including AlexNet, VGG, GoogLeNet, and ResNet, among others. These established networks exhibit commendable performance across diverse CV tasks, serving as pivotal benchmarks in both research and application domains.

In 1994, Lecun et al. introduced LeNet, one of the pioneering CNNs for handwriting font recognition. This seminal work played a pivotal role in shaping the subsequent evolution of CNNs. Krizhevsky introduced AlexNet in 2012 [33]. Building upon the foundations laid by LeNet, AlexNet effectively adopted the Rectified Linear Unit (ReLU) [34] as the activation function for CNN. This choice proves superior to Sigmoid [35] for networks with increased depth, addressing challenges like the vanishing gradient problem. Concurrently, CUDA (Compute Unified Device Architecture) technology facilitated the acceleration of deep convolutional network training, leveraging the robust parallel computing prowess of GPUs (Graphic Processing Units) to manage extensive operations during training. The remarkable achievement of winning the 2012 ImageNet competition attests to the efficacy of this approach, particularly notable for conventional machine learning classification algorithms. The success of AlexNet stands as empirical evidence showcasing the superior performance of CNNs in addressing large-scale image classification challenges. In 2014, Simonyan et al. [36] introduced VGG, a novel network architecture. In contrast to AlexNet, VGG boasts a deeper network structure and employs diminutive convolutional kernels to curtail parameter counts. This prowess was affirmed by its attainment of second place in the 2014 ILSVRC competition. Similarly, in that year, Szegedy introduced a novel deep learning architecture named GoogLeNet [37]. While prior networks predominantly pursued increased depth for enhanced training outcomes, this approach presented challenges like overfitting, gradient vanishing, and gradient explosion. GoogLeNet introduced the inception module, which not only optimizes computational resource utilization but also enhances feature extraction within the confines of the same computation load, thereby amplifying its effectiveness. However, as the number of network layers increases, the model becomes more challenging to train and is susceptible to issues like gradient vanishing. In 2015, He et al. introduced residual networks (ResNet) [38], accompanied by the design of a residual module tailored to mitigate the gradient vanishing issue arising from heightened network depth in deep neural networks. This architectural enhancement facilitates the training of deeper networks with greater ease. In 2016, Cai et al. [39] initially proposed a CNN leveraging cascade learning, which yielded superior detection results. This further underscores the significant potential of CNNs in PCB defect detection.

In summary, propelled by the swift advancement of deep learning, CNNs have achieved significant milestones within the realm of CV, fundamentally transforming image recognition and classification endeavors. This review centers on the canonical architectures of CNNs, encompassing LeNet, AlexNet, VGG, GoogLeNet, and ResNet. These architectures have garnered significant academic attention and concurrently exhibited exceptional prowess in industrial and practical contexts. Through a comprehensive comparison and analysis of these network architectures, a more profound comprehension of the influence exerted by diverse design strategies on CNN performance can be attained. This understanding will be instrumental in guiding future research endeavors within the CNN domain. Table 1 provides an overarching depiction of the strengths (including key attributes and innovations) and limitations characterizing the aforementioned CNNs. This review aims to furnish readers with a lucid comprehension of both the strengths and limitations of these networks.

The CNNs discussed above have witnessed a progression from modest layer counts to substantial depths, reaching dozens or even hundreds of layers. Simultaneously, the refinement of these networks has matured, evolving in tandem with advancements in activation functions, the integration of maximum pooling layers, and other techniques. Fig.2 (a) illustrates the schematic structure of the AlexNet network. Notably, AlexNet rescales all input images to dimensions of 32 × 32 before subjecting them to successive convolutional operations. Subsequently, the outcomes undergo compression via a fully connected layer, yielding a

TABLE 1. A comparative analysis of strength and boundedness among classical algorithms employed in CNNs.

Reference	Model	Strength	Boundedness
Lecun, [31] (1994)	LeNet	An early instance of a CNN, setting the foundation for the evolution of CNNs.	Due to its relatively shallow network structure and simplistic convolutional layer design, LeNet might exhibit suboptimal performance when confronted with intricate images and demanding visual tasks.
Krizhevsky, [33] (2012)	AlexNet	Employing ReLU as the activation function for CNNs and leveraging CUDA acceleration for training not only enhances network performance but also introduces a novel training approach.	Requires large computational and storage resources and is prone to overfitting when dealing with limited amounts of data.
Simonyan, [36] (2014)	VGG	A straightforward and consistent network structure, improved feature representation capabilities, and seamless transfer learning potential.	The high number of parameters and the high computational resource requirements make it easy to suffer from overfitting problems.
Szegedy, [37] (2014)	GoogleNet	By incorporating the Inception module, GoogLeNet adeptly captures information within images characterized by augmented depth and width.	The challenges encompass network complexity, dependence on substantial training data volumes, and feature redundancy.
He, [38] (2015)	ResNet	This issue of gradient vanishing and explosion has been effectively addressed, enabling the training of exceedingly deep network models.	-

1 × 10 vector that encodes character weights. In Fig.2 (b), the extended AlexNet architecture is depicted. This iteration involves deeper network structures and integrates the Max Pooling technique, inspired by LeNet. Furthermore, GPU acceleration is employed during training, culminating in the model's remarkable success in the ImageNet competition. Fig.2 (c) and 2(d) showcase the Inception module, a hallmark of both VGG and GoogLeNet, co-introduced in a single year. VGG employs a block-oriented approach to progressively extend the network, a straightforward yet efficient strategy. In contrast, GoogLeNet pursues network depth expansion through width enhancement, introducing the Inception module. Over time, this module has undergone iterative enhancements, resulting in impressive accuracy gains. Despite these advancements, the prevalent issue of excessive network depth during that era, contributing to suboptimal outcomes, remained unaddressed. In Fig.2 (e), the schematic representation of the residual module, a significant innovation by ResNet, is showcased. Within this module, the output encapsulates not only the network's result but also the sum of the original input, fostering a residual architecture. This structural paradigm recurs in subsequent network iterations, ultimately establishing itself as a fundamental element in the landscape of CNNs.

However, none of the aforementioned approaches considers the inherent human tendency to selectively concentrate on specific information while disregarding other visible data during the process of visual perception. The introduction of the attention mechanism, in turn, presents a more effective solution.

C. ATTENTION MECHANISM

The concept of the Attention Mechanism originated from observations in human vision. In 2014, Mnih et al. [40] noted that despite using multiple GPUs and priors to process image data, CNNs still required extended training periods. This concept was later merged with human observation images to implement the attention mechanism in image recognition, confirming its viability. This marked the inaugural integration of the attention mechanism into deep learning. Subsequently, this concept was extended to the domain of natural language processing (NLP) [41], [42]. In 2017, Google Research [43] introduced the Transformer model, founded on the concept of attention, which achieved remarkable results and disrupted the NLP field. Similarly, the realm of CV witnessed the emergence of numerous novel attention models. Hu et al. [44] introduced the SENet (Squeeze-and-Excitation Network), integrating attention mechanisms into feature channels. SENet autonomously learns the significance of various channels, using this acquired information to amplify valuable features and attenuate less relevant ones for the current task. SE attention mechanism module as shown in figure Fig.3 (a). Nonetheless, SENet overlooks the feature space aspect. Consequently, Woo et al. [45] introduced the convolutional block attention module (CBAM), which merges the attention mechanism in both feature channels and feature space dimensions, enhancing network performance without substantially increasing the parameter count. This module's schematic depiction can be found in Fig.3 (b). However, the prior attention mechanisms exhibited increased complexity, leading to heightened model intricacy. Therefore, Wang et al. [46] introduced efficient channel attention for deep convolutional neural networks (ECA-Net), characterized by a reduced parameter count and evident performance improvements. The schematic diagram of the ECA module is presented in Fig.3 (c). While earlier attention mechanisms mainly addressed inter-channel information, disregarding spatial location details or lacking long-range relationship capabilities, Hou and colleagues [47] proposed coordinate attention (CA). This method captures inter-channel information and considers orientation-based positional information, enhancing the model's ability to accurately localize and identify targets. Additionally, Coordinate Attention boasts a lightweight and flexible nature, permitting seamless

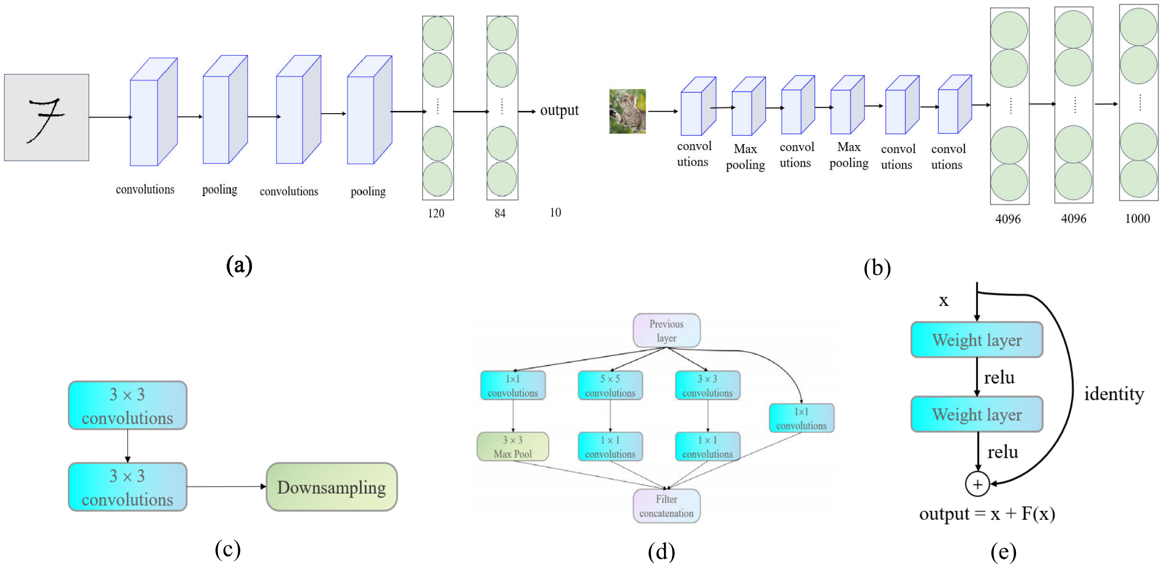


FIGURE 2. **Diagrams illustrating the network structures of classical CNNs: (a) LeNet; (b) AlexNet; (c) VGG block; (d) GoogleNet Inception module; (e) ResNet residual structure.**

integration into networks while delivering substantial performance enhancements. The CA attention module is displayed in Fig.3 (d).

In PCB defect detection, the majority of areas are typically defect-free, with only a minor proportion exhibiting defects. The attention mechanism, by directing more focus toward these defect-laden areas, facilitates more efficient defect detection. Consequently, incorporating an attention mechanism into the PCB defect detection model proves advantageous.

D. TRANSFORMER

The Transformer architecture, introduced by Google Research [43], represents a significant breakthrough in the realm of natural language processing [48]. Traditional sequential models, including recurrent neural networks (RNNs) and CNNs, often encounter issues such as gradient vanishing and explosion when handling lengthy sequences. In contrast, Transformer overcomes these challenges through the incorporation of a self-attention mechanism, enhancing its capability to capture extensive dependencies within sequences. Consequently, Transformer has emerged as the fundamental architecture within the NLP domain, and its core structure is depicted in Fig.4 (a).

As the Transformer gained prominence in the NLP field, scholars increasingly extended its application to the CV domain, exemplified by ViT (Vision Transformer) [49], [50]. Attention-based approach to complement CNN's limitations in ultra-long sequence modeling. This method transforms the input image dimensions [H, W, F] into two dimensions To Transformer's spatial location information deficiency, Bello extended the concept from [51] and bolstered Attention's prowess by incorporating relative position coding along the width and height dimensions. Subsequently, [52] advocated for replacing the convolution module in ResNet entirely with Attention-based relative position embedding to achieve a fully Attention-powered image model. Fig.5 (b) illustrates the local attention layer at k = 3. Progressing from CNN to Transformer modules, Carion et al. [53] introduced the End-to-End Object Detection with Transformers (DETR) model, marking the inaugural replacement of CNN with Transformer. DETR exhibited promising results in target detection, segmentation, and classification, though it lagged behind then-state-of-the-art CNN in certain aspects. DETR's model structure is outlined in Fig.5 (c). With the surge of researchers investigating the Transformer's role in CV tasks, Transformer's advancement in the CV domain has outpaced CNN's performance in large-scale models [50], [54], [55].

In PCB defect detection, defects are often minuscule, rendering traditional methods ineffective for their identification. Recently, the Transformer model, with its heightened sensitivity to small targets and complex defects in diverse environments, has been more suitable for PCB defect detection. Thus, employing the Transformer model for PCB defect detection could potentially enhance performance.

E. SUMMARY

This section delves into the fundamental concepts of deep learning, encompassing CNNs, attention mechanisms, and the transformer model. CNNs, originating in the last century, only truly began to thrive a little over a decade ago with the surge in computational power and accessibility to ample labeled data. This led to ground-breaking results in numerous domains and they have now become integral to PCB defect detection and machine vision. The attention mechanism, a versatile module, can be efficiently utilized in both

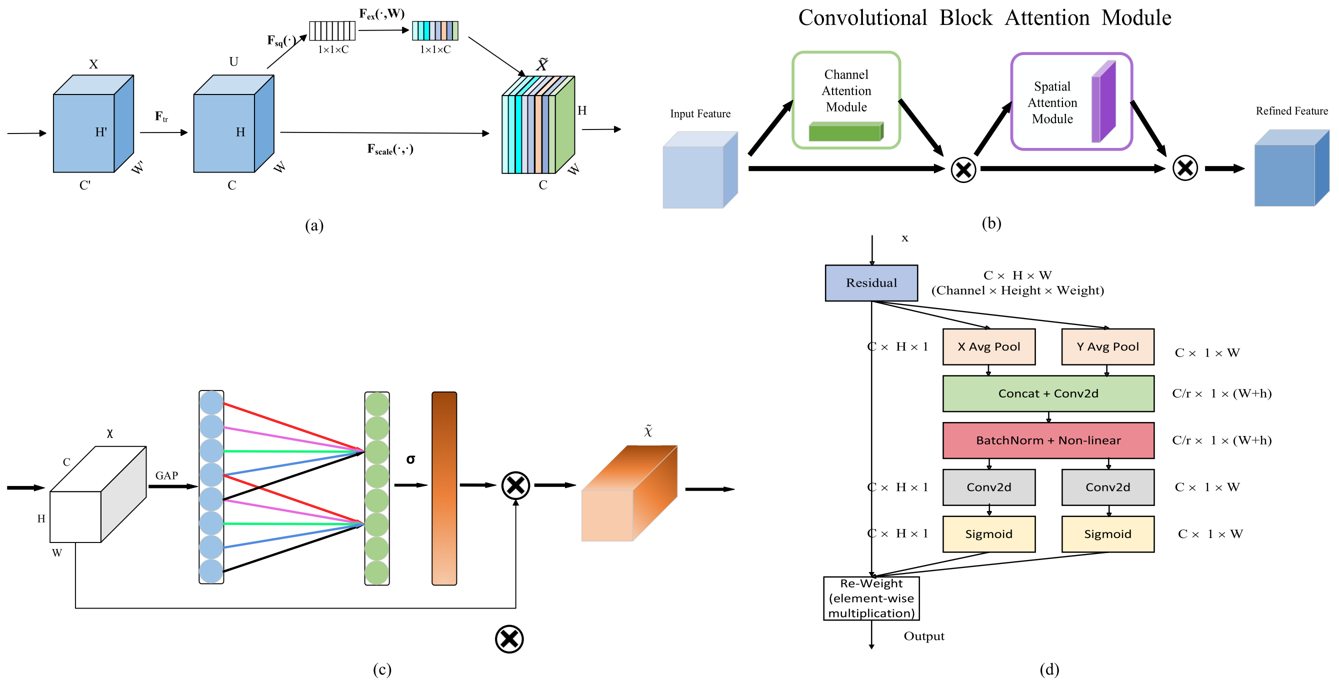


FIGURE 3. Diagram illustrating attention mechanisms: (a) SE attention mechanism module; (b) CBAM attention mechanism module; (c) ECA attention mechanism module; (d) CA attention mechanism module.

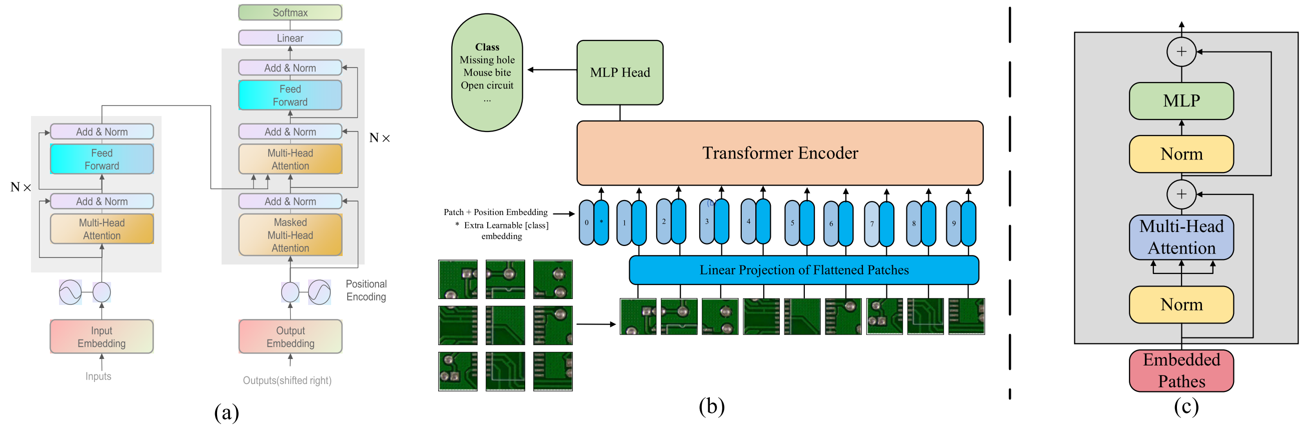


FIGURE 4. Transformer models: (a) The transformer model introduced by Google Research for NLP applications; (b) Fundamental architecture of ViT; (c) Transformer encoder configuration within ViT.

CNNs and transformers. Its role is to facilitate the model in prioritizing important features while disregarding irrelevant background information, thereby enhancing the detection of PCB defects. The transformer model is a deep learning construct built upon the self-attention mechanism. Despite its late introduction, the transformer model has witnessed rapid growth due to the extensive research conducted by a myriad of scholars. It has found widespread application in numerous tasks within NLP and CV, delivering outstanding results. The potential for further development of the transformer model is undeniable.

III. PCB DEFECT DETECTION METHOD BASED ON DEEP LEARNING

In the domain of PCB defect detection, detection algorithms are categorized into two primary groups: two-stage algorithms and single-stage algorithms [57], [58]. Apart from the aforementioned CNN-based methods, a category of PCB defect detection algorithms based on Transformers has also emerged. Notable two-stage algorithms encompass Region CNN (R-CNN) [59], Fast Region CNN (Fast R-CNN) [60], Faster Region CNN (Faster R-CNN) [61], and Mask Region CNN (Mask R-CNN) [62], among others. These methods partition PCB defect detection into two phases: firstly, region proposal (RP), which involves generating pre-selected boxes potentially containing the objects to be detected; subsequently, sample classification is conducted using CNNs.

The R-CNN algorithm starts by generating a sequence of candidate regions (Region Proposals) within the input image. It then performs CNN feature extraction on these regions and forwards the extracted features to classifiers and bounding box regressors for target detection. While possessing high

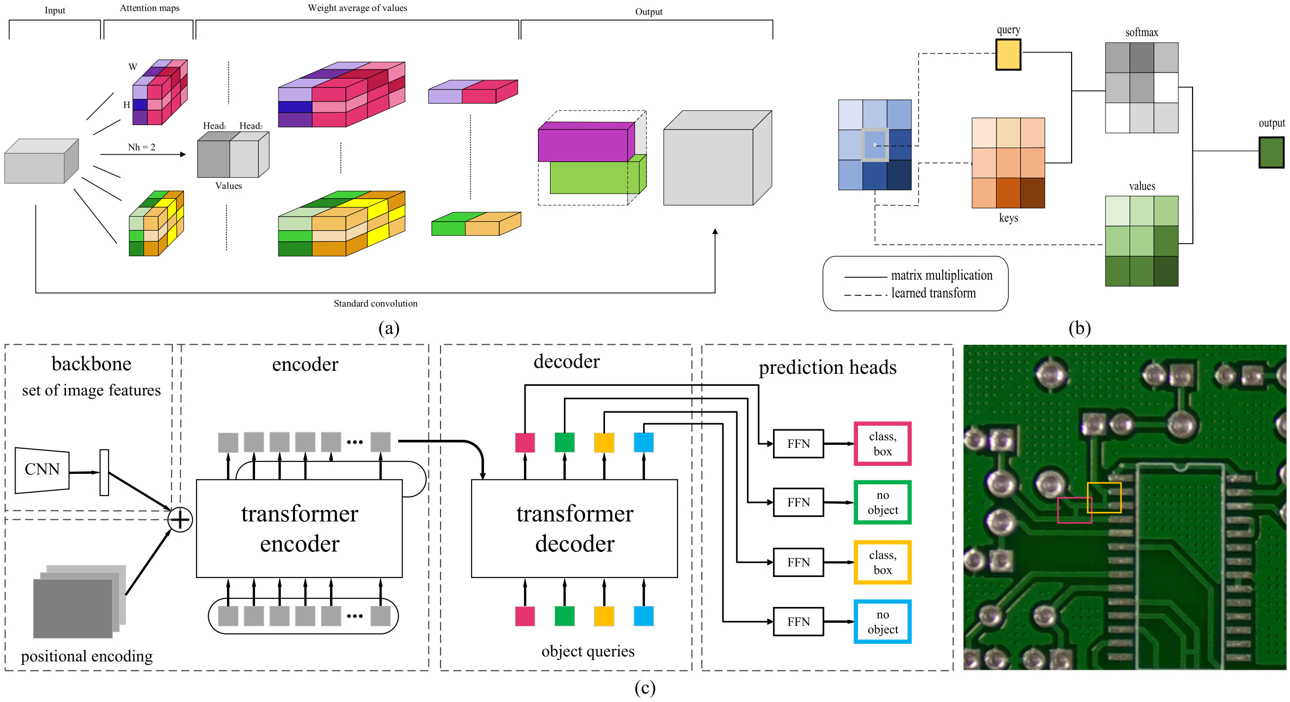


FIGURE 5. Transformer models in CV: (a) Schematic of the proposed attention-augmented convolutional architecture; (b) Substitution of attention for convolution in ResNet with kernel size k=3; (c) Pioneering alternative to CNN for the DETR model.

accuracy, its speed is impeded by the need for separate CNN feature extraction for each candidate region. The algorithm's structure is depicted in Fig. 6 (a). Faster R-CNN, an upgraded version of R-CNN, introduces the region proposal network (RPN), a learnable network designed for rapid candidate region generation. Faster R-CNN seamlessly integrates the RPN with subsequent classifiers and bounding box regressors, resulting in an end-to-end target detection network. This integration enables candidate region generation and feature extraction within the same network, significantly enhancing detection speed. The configuration of Faster R-CNN is shown in Fig.6 (b).

The other category encompasses single-stage algorithms, prominently represented by single shot multiBox detector (SSD) [63] and You Only Look Once (YOLO) [64], [65], [66], [67], [68], [69]. These methods redefine PCB defect detection as a regression problem. The YOLO algorithm introduces a distinctive perspective, treating target detection as a regression challenge. It accomplishes this by partitioning the input image into a grid and concurrently predicting class and bounding box attributes for multiple targets within each grid cell. YOLO achieves target detection through single-shot forward propagation, rendering it exceptionally rapid. However, its performance in detecting small targets might be comparatively less effective. The structure of YOLO is depicted in Fig. 6 (c). The SSD algorithm also falls under the single-stage target detection approach. Similar to YOLO, it simultaneously forecasts target categories and bounding boxes on feature maps at varying scales, adapting to diverse target scales via multiple anchor frames of differing dimensions. This design enables SSD to excel in detecting small and multi-scale targets. Its structure is illustrated in Fig. 6 (d). In comparison to YOLO, SSD exhibits slightly slower performance, although it may offer heightened accuracy in specific contexts.

Comparing the two-stage and single-stage algorithms, the two-stage algorithm has higher accuracy but is time-consuming and not suitable for real-time detection tasks. Whereas, the single-stage algorithm is faster but relatively less accurate. Transformer-based PCB target detection algorithms have also emerged, and unlike traditional CNNs, these methods are based on Transformer and show excellent detection accuracy, but the detection speed is slower, the training requires higher computational resources, and it requires a large amount of data support [70].

A. BASED ONE-STAGE ALGORITHM

Addressing the issues of poor stability and low accuracy in PCB defect detection models, Xin et al. [71] introduced an enhanced YOLOv4 model. This approach incorporates a mosaic data augmentation strategy during input processing and replaces the leaky rectified linear unit (Leaky-ReLU) activation function in the network backbone with the Mish activation function [72]. Additionally, the detection images are automatically segmented based on the average size of labeled boxes, thereby increasing the likelihood of including the target in the anchor frame. Similarly, to address the challenges of detecting small defects against complex backgrounds in PCBs, Zhang et al. [73] introduced a lightweight single-stage defect detection network. This network leverages

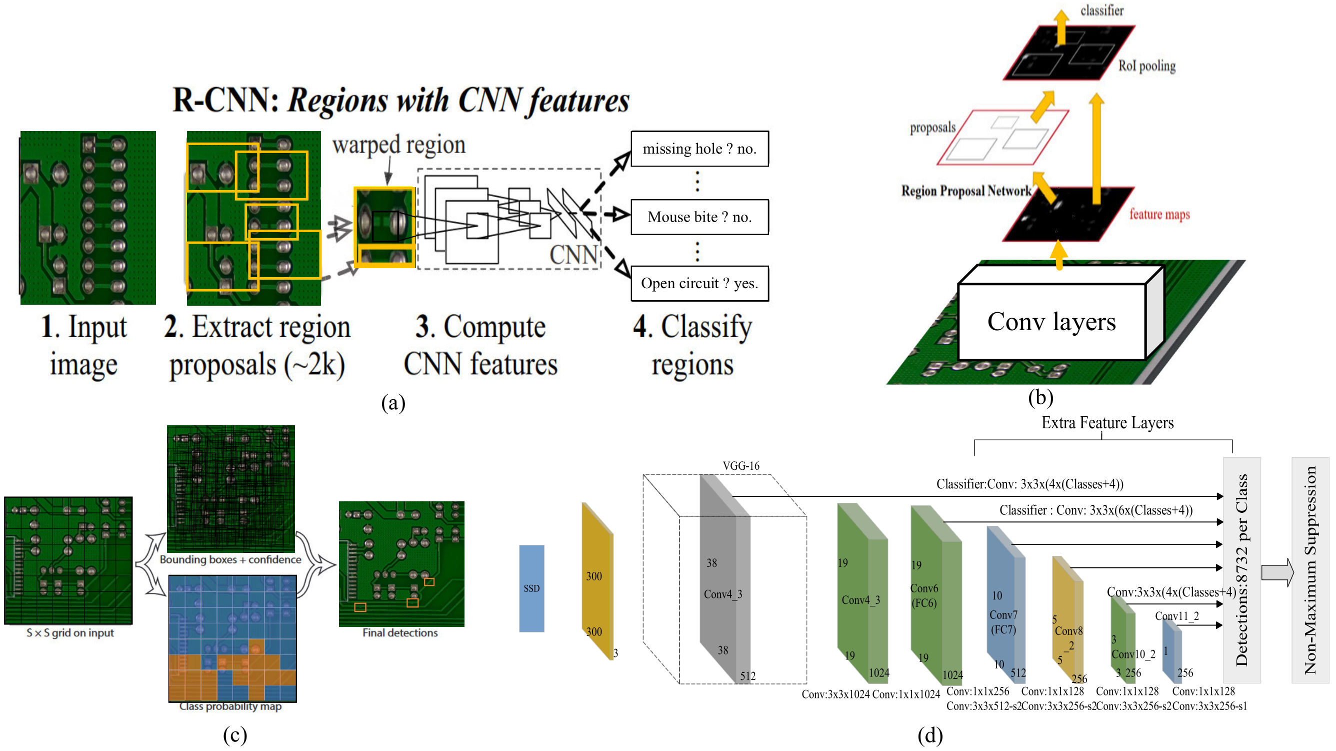


FIGURE 6. Traditional target detection algorithms: (a) Flowchart of R-CNN; (b) Fundamental structure of faster R-CNN; (c) Core process of the YOLO algorithm; (d) Network architecture schematic of the SSD algorithm.

the dual attention mechanism and a path-aggregation feature pyramid network (PAFPN) to enhance the detection of small defects. MobileNetV2, a lightweight backbone neural network, replaces ResNet101, considerably reducing model parameters. A dual attention mechanism is integrated to ensure effective feature extraction. The feature extraction capability is further enhanced by substituting PAFPN [74] for feature pyramid network (FPN) [75] in the neck. This improved model reduces inference time, and parameter count, and enhances detection accuracy. Jiang et al. [76] proposed enhancements to the SSD network model, introducing coordinated attention in the shallow network to better handle positional information, particularly for smaller targets. Li et al. [77] created a dataset for PCB assembly scene object detection, addressing anchor frame size-related detection issues. They performed a detailed analysis of effective receptive fields (ERF) [78] across the output layers, defining ERF ranges and introducing ERF-based anchor frame assignment rules to address anchor frame size challenges. Furthermore, they designed an improved atrous spatial pyramid pooling (ASPP) [79], [80], incorporated a channel attention module, and added contextual information to address challenges posed by small and hard-to-detect defects.

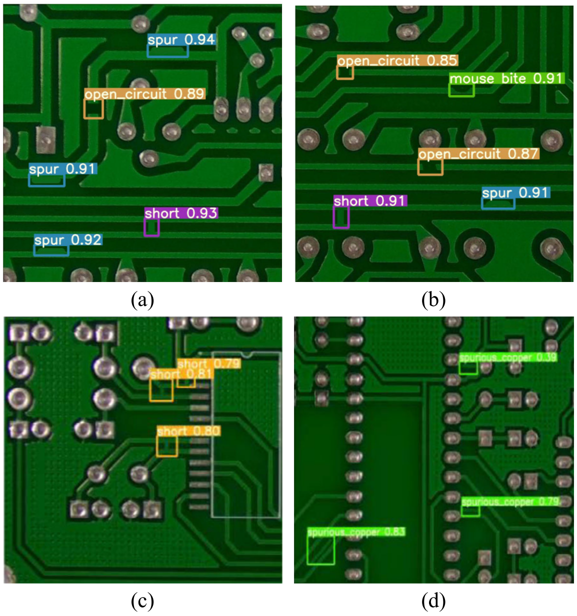
Due to the limited number of labeled PCB defect samples, the training process is influenced by unlabeled samples. To address this, Wan et al. [81] introduced a defect detection method with a data-expanding strategy (DE-SSD) and evaluated its performance on YOLOv5 using both labeled and unlabeled samples. This approach reduces the reliance on labeled samples, as it utilizes both labeled and unlabeled data. Moreover, a data-expanding strategy is proposed to mitigate the impact of unlabeled samples. This enhancement is particularly evident with small data volumes; however, the effectiveness diminishes as data volume increases. In another study, Wu et al. [82] presented GSC YOLOv5, a deep learning detection approach that integrates a lightweight network with a dual-attention mechanism. This modified algorithm employs lightweight Ghost Conv and Ghost Bottleneck structures to significantly reduce the model's parameter count and floating-point operations. Furthermore, SE and CBAM modules are incorporated into the network, resulting in enhanced accuracy and improved detection speed. Addressing concerns related to detection efficiency, memory consumption, and sensitivity to small defects, Xuan et al [83] adopted a novel cross stage partial network darknet (CSPDarkNet) as the YOLOX backbone network. This revised backbone comprises multiple inverted residual blocks and incorporates coordinated attention into the network architecture, significantly improving the model's capacity to detect small PCB defects. Notably, this modified model is lighter and more suitable for deployment on embedded systems. Zhao, Y. et al. [84] extended YOLOv5 by integrating adaptively spatial feature fusion (ASFF) [85] for feature fusion. This integration enables the adaptive fusion of varying levels of feature information across different spaces. Additionally, they introduced the global attention mechanism (GAM) [86] to enhance the model's information extraction capabilities.

Zheng and colleagues [87] introduced an enhanced fully CNN by integrating successive convolutional modules into the MobileNetV2 architecture. This augmentation, alongside an improved skip connection, contributes to heightened detection speed and accuracy in contrast to VGG-16 and ResNet-50 models. Lim and colleagues [88] developed a novel multi-scale feature pyramid network using YOLOv5, addressing the detection of minuscule PCB defects by leveraging contextual insights. The network also incorporates the CIoU loss function to precisely determine the spatial parameters, effectively capturing the exact positions of these imperfections. Yu and co-authors [89] introduced a lightweight, efficient network tailored for detecting minute PCB defects. Within the backbone network, they introduced diagonal feature pyramid (DFP), a mechanism for low-cost fusion of expansive feature maps, enhancing the detection of these subtle flaws. Additionally, they devised a multi-scale necking network to accommodate various scales of defects. Moreover, they introduced an adaptive localization loss function, enhancing the model's ability to discern these small-scale imperfections.

In summary, the aforementioned research has made significant advancements in the field of PCB defect detection using one-stage algorithms. These advancements encompass various novel approaches, including attention mechanisms, data enhancement techniques, and innovative backbone networks. These methods have found widespread application in PCB defect detection from images, yielding impressive outcomes across multiple dimensions such as enhanced accuracy and detection speed. A comparative analysis of these one-stage algorithms with specific alternative methods, along with their benefits and limitations, is presented in Table 2. It is crucial to acknowledge, however, that the one-stage algorithms are not exempt from limitations. These limitations encompass the potential for diminished performance when confronted with intricate and varied defective scenarios, the high reliance on limited sample data, and the imperative for further enhancements in coping with diverse dimensions, angles, and lighting conditions. Fig. 7 presents the results of PCB defect detection using four different algorithms.

B. BASED TWO-STAGE ALGORITHM

While the single-stage algorithm offers faster performance, the two-stage algorithm notably outperforms it in terms of detection accuracy. To achieve PCB defect detection via machine vision, Li et al. [90] introduced a faster-RCNN algorithm founded on VGG16. This algorithm incorporates data expansion and RGB data enhancement. Addressing the challenge of detecting tiny defects, which are inherently challenging to generate and detect in real-world scenarios, Ding et al. [8] introduced tiny defect detection network (TDD-net). This network employs a K-means algorithm to design appropriate anchor frames, enhances inter-feature map relationships, and utilizes online hard example mining (OHEM) [91] to refine region of interest (ROI) prediction. Addressing the limitations of template-dependent and computationally demanding traditional defect detection methods, Hu et al. [92] introduced an algorithm founded on Faster RCNN and Feature Pyramid FPN (Feature Pyramid Network). This algorithm initially employs ResNet50 with feature pyramids as its backbone network and subsequently integrates generative adversarial region proposal networks (GARPN) [93] to enhance anchor frame prediction accuracy. In the same year, Li et al. [94] introduced a feature pyramid-based network. They incorporated an SE module into ResNet-101 to enhance network expressive power, introduced a top-down structure to elevate overall feature levels, and employed ROI Align in lieu of ROI Pooling to mitigate the impact of dislocations on small object defect detection.



(a) (b) (c) (d)

FIGURE 7. Results of PCB defect detection using four different algorithms: (a) (b) Printed circuit boards defect detection method based on improved fully convolutional networks [87]; (c) (d) Printed circuit board quality detection method integrating lightweight network and dual attention mechanism [82].

In summary, when considering studies based on two-stage algorithms, it's evident that there are fewer scholarly works compared to those focused on one-stage algorithms. However, there's no doubt that two-stage-based algorithms offer significant advantages in terms of detection accuracy. Table 3 provides a comparison of two-stage-based algorithms, detailing the specific methods employed, along with their respective advantages and limitations. Recently, one-stage algorithms have flourished to meet real-time detection speed requirements. However, the difference in detection accuracy, compared to two-stage algorithms, is not significant. As a result, there are relatively fewer researchers focusing on this field. Nonetheless, utilizing two-stage algorithms remains a valuable choice for specific tasks

TABLE 2. A comparison of strength and boundedness in one-stage CNN algorithms.

Reference	Method	Strength	Boundedness
Xin, [71] (2021)	Enhancing data through mosaic techniques and Leaky-ReLU	Preprocessing the input image and analyzing it to Improve training results	Mosaic data augmentation was employed, extending the training duration.
Zhang, [73] (2021)	Dual attention mechanism and PAFPN	The network is lightweight, enhancing small defect detection, minimizing inference time, and reducing parameter count.	Longer inference time compared to certain models.
Jiang, [76] (2022)	CA and SE	Networks exhibit enhanced accuracy in predicting small targets.	The dataset is relatively small, comprising only five classes of PCB defect images, with a limited quantity.
Li, J. [77] (2022)	Optimise anchor box size, ERF, and ASPP	Effectively addresses anchor box size and enhances detection of small and hard-to-detect defects.	Without comparing it to other detection networks, such as Faster R-CNN or SSD, on the same dataset, it is challenging to assess the relative performance advantage of the model.
Wan, [81] (2022)	Semi-supervised learning and data augmentation	Accuracy improvement is more obvious when the amount of data is small	Accuracy improvement isn't evident with larger data sets, demanding prolonged training.
Wu, [82] (2022)	Lightweight Network with Dual Attention Mechanism	Efficiently reduces model parameters and flops.	There is a lack of exposition regarding the design and training procedures of the two attention mechanisms. Further investigation is needed to elucidate how these attention mechanisms collaborate and their impact on enhancing accuracy.
Xuan, [83] (2022)	CSPDarkNet and CA	Enhances the network's small PCB defect detection capability, while also offering a lightweight model for embedded system deployment.	The data set's imbalance in small defect samples may result in the model's inferior detection performance for small defects compared to moderate ones.
Zhao, [84] (2022)	ASFF and GAM	Enhanced feature extraction for models	Extended inference times and augmented model parameter count.
Zheng, [87] (2022)	MobileNetV2 full CNN	Offers advantages in detection speed and accuracy.	The paper lacks analysis regarding the adjustment of various hyperparameters and network structures.
Lim, [88] (2023)	Multi-scale feature pyramid network and CIoU loss	Enhanced detection of small or evolving PCB defects in real-time.	limited recognition of defect types.
Yu, [89] (2023)	DFP, Multi-scale neck networks, and adaptive localization loss function	Enhancing the network's small defect detection capability.	-

TABLE 3. A comparison of strength and boundedness in two-stage CNN algorithms.

Reference	Model	Strength	Boundedness
Li, [90] (2018)	VGG16, faster-RCNN, data expansion, and RGB data enhancement.	Enhanced accuracy.	Reduced Speed and Limited Elevation.
Ding, [8] (2019)	TDD-net, K-means, and OHEM	Enhanced small defect detection capability.	-
Hu, [92] (2020)	Faster-RCNN, FPN and GARPN	Improved precision in anchor frame prediction.	The real-time performance needs improvement.
Li, [94] (2020)	FPN, SE and ROI align	Enhanced expressive power and improved network detection.	The computational workload is substantial, and the inference time may not meet the real-time online inspection requirements of the PCB manufacturing process.

TABLE 4. A comparison of strength and boundedness in transformer algorithms.

Reference	Model	Strength	Boundedness
An, [95] (2022)	LPViT with the incorporation of a label smoothing strategy.	Enhanced accuracy.	Enhanced accuracy with reduced processing time.
Chen, [96] (2022)	Enhanced clustering algorithm utilizing Swin-Transformer.	Efficiently establishes correlation among image features with state-of-the-art accuracy.	Prolonged inference time.
Yang, [97] (2023)	SwinV2_TDD, MFSA and SA	Elevated accuracy and generalization capability.	Limited detection capability for minor defects.

demanding high detection accuracy and tailored datasets, without necessitating rapid detection speed. Fig. 8 presents the results of PCB defect detection using four different algorithms.

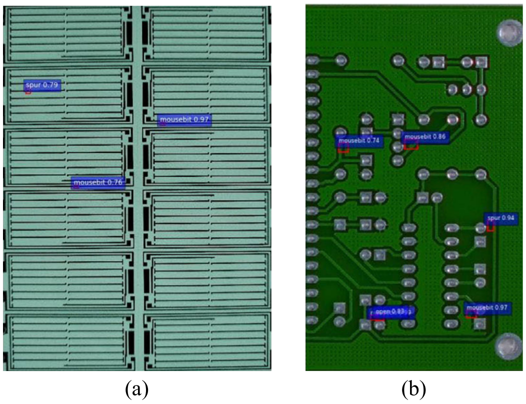


FIGURE 8. Results of PCB defect detection using four different algorithms: (a) (b) Detection of PCB surface defects with improved faster-RCNN and feature pyramid network [92].

C. BASED TRANSFORMER ALGORITHM

While Transformer has demonstrated proficiency in CV and NLP, it does exhibit limitations when employed in visual inspection tasks, notably involving time constraints and demanding equipment prerequisites. These limitations have resulted in a scarcity of literature concerning the implementation of Transformer for PCB defect detection. Nevertheless, notwithstanding these limitations, several studies have commenced efforts to address these challenges, aiming to unlock the full potential of Transformer in the realm of PCB defect detection. By surmounting the limitations of the transformer and effectively employing it in PCB defect inspection tasks, it is anticipated to offer a more efficient and accurate solution for industrial production.

An et al. [95] introduced label robust and patch correlation enhanced ViT (LPViT). In their work, a novel ViT model is presented, which is founded upon the principles of LPViT. This model prioritizes robustness while fully leveraging distinct regions of the PCB image across relationships. Additionally, certain blocks are randomly masked or substituted to enhance mutual understanding between different image regions. Ultimately, the model undergoes training via a label smoothing strategy, elevating its robustness. Chen [96] employs an enhanced clustering algorithm to generate appropriate anchor frames tailored to the PCB defective dataset in this study. Next, CNN was abandoned in favor of shifted window transformer (Swin-Transformer) for network feature extraction. Subsequently, the channel order in the feature map was adjusted to enable the network to efficiently prioritize more valuable information. Additionally, both convolutional and attentional mechanisms were integrated to enhance the network's feature extraction capacity. Yang et al. [97] introduced an enhanced YOLOv7 model. They achieved this enhancement by formulating the SwinV2_TDD module, which facilitates the extraction of local PCB information through the incorporation of an added convolutional layer. Next, the study introduces the MFSA mechanism, which augments each shuffle attention (SA) branch with a convolutional layer. This addition serves to amplify the depth of SA and bolster the flexibility of the attention mechanism. Fig. 9 presents the results of PCB defect detection using four different algorithms.

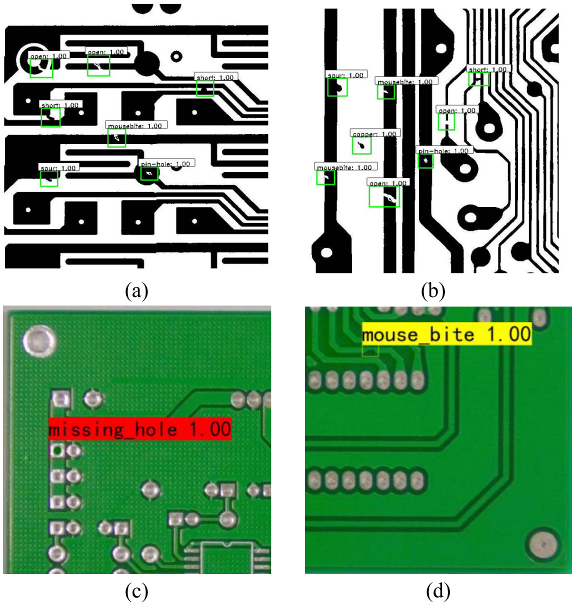


FIGURE 9. Results of PCB defect detection using four different algorithms: (a) (b) LPViT [95]; (c) (d) Transformer-YOLO [96].

Despite transformers being infrequently used in PCB defect detection, the evolution of computer vision and computational capabilities has catalyzed the proposal of numerous transformer models. In 2021, Liu et al [55] trained a swin transformer v2 model, enriched with three billion parameters, and introduced a post-normalization technique and scaled cosine attention method. This approach achieved state-of-the-art (SOTA) results across multiple visual tasks. As of 2023, the swin transformer v2 backbone network continues to be a subject of extensive research, demonstrating exceptional.

Performance and further stimulating the evolution of large visual models. Today's ViTs have not explicitly leveraged features at different scales, which are notably crucial for visual inputs. To address this, Wang et al [98] proposed the crossformer in 2021, introducing a cross-scale embedding layer (CEL) and a long-short distance attention (LSDA). In 2023, they [99] further enhanced the crossformer by proposing the progressive group size (PGS) and amplitude cooling layer (ACL) to mitigate challenges associated with enlarging self-attention maps and amplitude explosion. Given the lack of a priori image information, ViT underperforms in dense prediction tasks. To remedy this, Chen et al [100] proposed the ViT-adapter in 2022, an ancillary network not requiring pre-training, enabling the basic ViT model to adapt to downstream dense prediction tasks without any architectural modifications. This significantly improved model performance for dense prediction tasks.

These novel transformer models, designed for visual macro-models, harness features at different scales and excel in dense prediction tasks. These tasks are also inherent in transformer-based PCB defect detection. We posit that these innovative transformer models and their methodologies can be applied to PCB defect detection tasks, enhancing the performance of existing models.

D. SUMMARY

In this section, we elucidate the one-stage, two-stage, and transformer-based algorithms. One-stage algorithms, characterized by high detection accuracy and speed, cater to enterprise needs for real-time PCB defect detection. However, their efficiency may diminish in more challenging defect scenarios. In contrast, the two-stage algorithm, despite being slower, excels in detecting intricate defects due to its superior accuracy, leading to its widespread adoption in factories for PCB defect detection. The transformer-based algorithm, distinct in structure from its counterparts, has also demonstrated commendable results in PCB defect detection and various industrial tasks. Despite its large parameter count, it effectively fulfills factory requirements for PCB defect detection. Concurrently, the transformer model exhibits impressive performance in other domains, indicating substantial potential for further development.

IV. ASSESSMENT METRICS, PCB DEFECT DATASETS, AND COMPARATIVE RESULTS

A. ASSESSMENT METRICS

PCB defect detection tasks often require assessing algorithmic localization accuracy, which measures the discrepancy between the target bounding box identified by the algorithm and the actual target bounding box [101]. For assessing algorithmic positional accuracy, a widely adopted metric is the intersection over union (IoU) [102], [103], defined as the ratio of the intersection area between the target frame and the actual frame to the union area. Specifically, in the context of the detection outcomes and real annotations depicted in Fig. 10 below, the intersection encompasses the overlapping portion of the two frames, while the union encompasses the combined area of both frames. The extent of overlap between detection outcomes and actual labeling is computed through intersection and union ratios, facilitating algorithmic positional accuracy evaluation as presented in formula (1). In practical detection scenarios, the intersection ratio threshold is commonly set to 0.5. If the intersection ratio between the targeted detection frame and the actual labeled frame exceeds 0.5, the algorithm is deemed to have successfully anticipated the target's location [104]. Modifying the intersection ratio threshold enables adjustments to the algorithm's positional accuracy and miss rate. Notably, distinct thresholds may necessitate establishment for varying tasks.

$$IoU = \frac{A \cap B}{A \cup B} \tag{1}$$

Equation (1) reveals that a greater IoU signifies the proximity of the estimated object region to the actual region, thus leading to higher accuracy in the detection results.

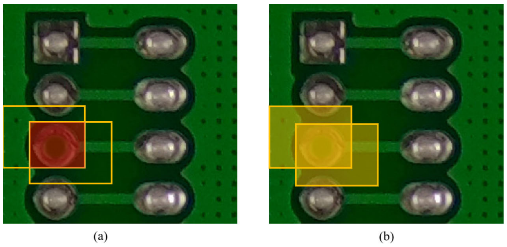


FIGURE 10. IoU schematic diagram: (a) Red represents the intersection region [8]; (b) Yellow represents union region [8].

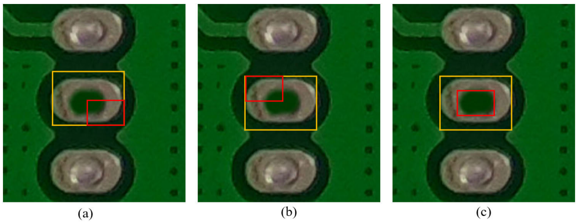


FIGURE 11. IoU limitations (while the IoU values are equal for the three images, it's noticeable that image (a) [8] and image (b) [8] exhibit lower accuracy. Image(c) demonstrates superior accuracy [8].)

Specifically, IoU's highest value, 1, signifies complete overlap between the actual object area and the inferred area, whereas its lowest value, 0, indicates no overlapping segment between the actual object area and the inferred area. However, there are instances where IoU may not precisely capture positional accuracy. For instance, in the three images shown in Fig.11, although the calculated IoUs for these images (Fig.11 (c)) are equal, it's evident that the third image is superior. Consequently, several enhanced iterations of IoU have emerged, including GIoU [102], DIoU [105], CIoU [105], EIoU [106], αIoU [107], and SIoU [108]. GIoU resolves the issue of the unavailability of gradient backpropagation for IoU's two frames without their intersection. Meanwhile, DIoU considers the prediction frame and real frame, building on the GIoU distance involving the centroid and the distance between the minimum enclosing frames. It also accounts for aspect ratio relationships based on DIoU. However, it does not address the gap in actual distance. Consequently, EIoU replaces DIoU's aspect ratio with the actual difference between width and height, along with their respective confidence levels. In 2021, αIoU was introduced, which involves only a single parameter α in comparison to other IoUs, yet it yields superior outcomes. The presence of multiple angles at equidistant points impacts the actual loss. Therefore, in 2022, SIoU was introduced to address the influence of angles. However, at present, a modified version of IoU is solely employed in the loss function, whereas IoU remains the standard for evaluation metrics.

Categorizing detection results in target detection tasks involves four distinct groups [109]: predicted values aligned with positive examples are labeled as P (Positive), those aligned with negative examples are labeled as N (Negative), values aligning with true values are T (True), and values

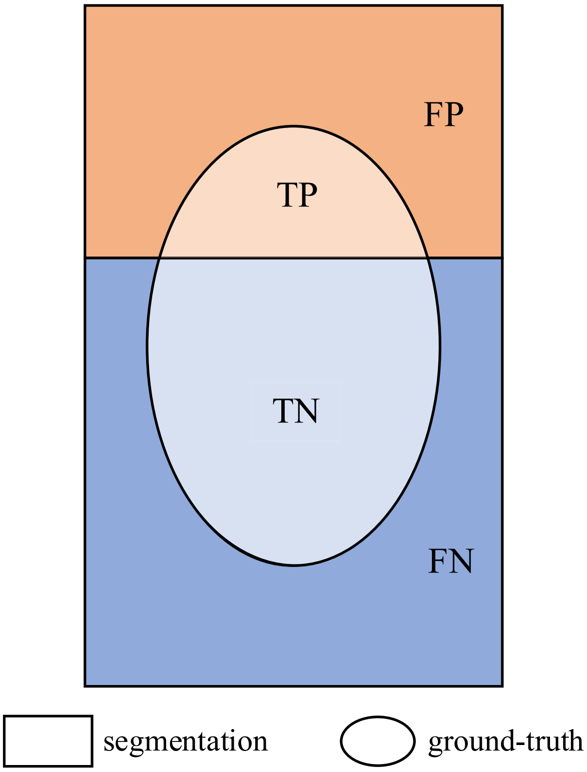


FIGURE 12. Schematic diagram of the recognition result.

opposing true values are F (False). Upon classification, the data can be organized into a confusion matrix to yield four distinct combination types.

Notably, As shown in Fig.12, TP represents the count of correctly detected targets, encompassing instances where the predicted positive sample aligns with the true positive sample; TN signifies the count of accurately identified background instances, where the predicted negative sample aligns with the true negative sample; FP designates the count of erroneously detected targets, represented by instances where the predicted positive sample mismatches the true negative sample; and FN accounts for the count of improperly missed targets, signifying cases where the predicted negative sample mismatches the true positive sample.

Evaluating algorithm performance becomes more precise through a comprehensive tally and comparison of predictions based on distinct types. Examples of assessment metrics include Accuracy (Acc), Precision [110], Recall [111], F1 Score [26], Average Precision [112], mAP [104], mAP$^{\text{Small}}$, mAP$^{\text{medium}}$, mAP$^{\text{large}}$ [113], and FPS (Frames Per Second) [67].

1) ACCURACY

The accuracy rate represents the proportion of correctly classified samples among the total samples, thus indicating the model's precision in classifying input data. The formula is depicted in Equation (2).

$$Acc = \frac{TP + TN}{TP + FN + FP + FN} \tag{2}$$

2) PRECISION

Precision refers to the ratio of correctly predicted positive samples to the total predicted positive samples by the model. The formula is depicted in Equation (3).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

3) RECALL

Recall, also known as the true positive rate, represents the ratio of correctly predicted positive samples to the actual total positive samples. The formula is illustrated in Equation (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

4) F1 SCORE

The F1 score provides a balanced mean between precision and recall, effectively harmonizing the model's accuracy and recall. The formula is depicted in Equation (5).

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{5}$$

5) AVERAGE PRECISION

Among the frequently employed evaluation metrics, Precision and Recall stand as crucial indicators for gauging model performance. However, a notable observation arises from the equation above. Precision and Recall exhibit a contradictory relationship: an increase in Precision often corresponds to a decrease in Recall, and vice versa. To address this challenge, the Average Precision (AP) metric is introduced to comprehensively evaluate the model's performance.

The term "average precision" specifically denotes the mean value resulting from the integration of accuracy rates across distinct thresholds, spanning a recall range from 0 to 1. For each category, a check-accuracy-recall curve is plotted, enabling the calculation of the area beneath this curve, referred to as the AP value. Consequently, a model's average precision corresponds to the mean AP values across all categories, constituting the mean average precision (mAP). Notably, the mAP metric stands as one of the most extensively employed performance measures in target detection, quantifying a model's proficiency in identifying multiple categories of targets.

The computation methods for mAP metrics vary. Specifically, AP0.5 represents the average accuracy when the intersection-over-union (IoU) threshold exceeds 50%, and AP0.5:0.95 corresponds to the average accuracy when the IoU threshold ranges from 50% to 95% in 5% increments. In real-world target detection tasks, models often need to detect targets from multiple categories, necessitating the calculation of AP values for each category and subsequent averaging to derive the mAP metric.

In conclusion, the mean average precision (mAP) stands as a crucial metric in assessing target detection model performance. By amalgamating model accuracy and recall, mAP offers insight into a model's ability to detect a multitude of diverse target categories. Consequently, this review employs mAP0.5:0.95 and mAP@0.5 (AP0.5) as the primary evaluation criterion. The formula is depicted in Equation (6).

$$AP = \int_{0}^{1} P(R)dr, mAP = \frac{1}{N}\sum_{(i=1)}^{N} AP_i \tag{6}$$

6) mAP^{Small}, mAP^{medium}, AND mAP^{large}

These three metrics correspond to average accuracies for objects of different sizes. Specifically, mAPS^{mall} pertains to objects with an area less than 32 × 32 pixels, mAP^{medium} corresponds to objects with an area more than 32 × 32 pixels and less than 96 × 96 pixels, and mAP^{large} relates to objects with an area greater than 96 × 96 pixels. These metrics provide a more detailed insight into the algorithm's detection accuracy across small, medium, and large targets.

7) FPS

FPS serves as a metric for assessing inference speed, representing the quantity of images that can be processed per second on specific hardware. FPS holds significance as a key metric for gauging model performance and its applicability in real-time scenarios. Through computation of the model's processing capacity for images in a single second, the model's real-time performance can be assessed. Elevated FPS values signify the model's capacity for swifter image processing, thereby enhancing real-time inference efficiency. This attribute is crucial for numerous applications demanding rapid response times, including real-time video analysis [64], autonomous driving systems [114], and real-time object recognition [64]. Consequently, researchers and developers strive to enhance a model's FPS value.

B. PCB DEFECT DATASETS

In the literature related to PCB defect detection mentioned earlier, many studies utilize proprietary datasets. Therefore, this subsection focuses on introducing several publicly available datasets, which possess certain evident advantages compared to proprietary datasets. Public datasets not only offer greater persuasiveness but also provide more accurate baselines. Additionally, they facilitate experiment reproducibility among other researchers. These publicly available datasets offer researchers more reliable standards and benchmarks, fostering performance comparison and method enhancement. Simultaneously, they provide a broader platform for progress and collaboration within the research community. Currently available datasets include PCB Defect, PCB Defect-Augmented [8], DEEP PCB [115], HRIPCB [116], and Micro-PCB, among others. These datasets exhibit varying characteristics such as different defect types, image quantities, and environmental factors. Models trained on different datasets yield varying accuracies and are suited for different scenarios. Detailed information regarding currently available publicly accessible PCB defect detection datasets is presented in Table 5.

C. RESULT COMPARING

This subsection will primarily focus on an in-depth analysis and meticulous comparison of the previously cited literature. By systematically summarizing and comprehensively evaluating the research outcomes from these references, we can attain a more comprehensive understanding of the latest advancements in the relevant field and unveil both commonalities and disparities therein. The results comparison encompasses various metrics, including mean average precision (mAP), frames per second (FPS), parameter quantity, and recall rate. These metrics collectively provide a comprehensive assessment of the performance of different algorithms or models in experimental settings. Furthermore, within the comparative experiments, detailed specifics of the utilized datasets will also be elaborated upon.

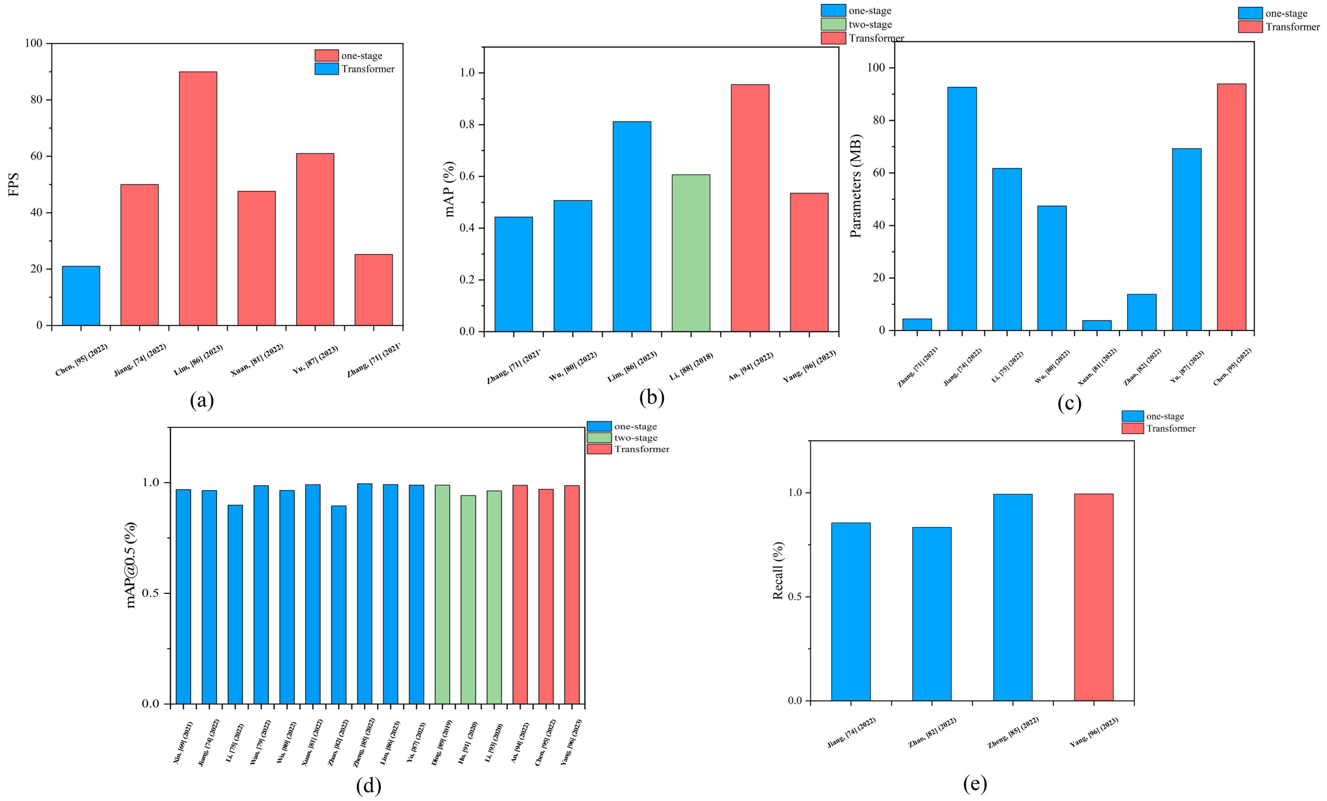
Various performance metrics from the aforementioned literature are presented in Fig.13. Fig.13 (a) illustrates a comparison of FPS across different algorithms, indicating that many of these algorithms are primarily single-stage models, achieving real-time detection goals, with some even reaching up to 90 FPS. Achieving a balance between inference speed and required accuracy is crucial in practical development to meet industrial demands. In Fig. 13 (b), a comparison of mAP reveals significant disparities, possibly arising from dataset diversity and difficulty levels. Nevertheless, all these algorithms yield commendable results. Fig. 13 (c) displays a comparison of parameters, where lower values generally correspond to reduced device demands and shorter inference times. Models with fewer parameters are suitable for deployment on mobile devices to fulfill specific requirements. The two studies with the lowest parameters often accomplished this through network optimization and pruning, effectively reducing parameters without notable accuracy compromise. Fig. 13 (d) presents mAP@0.5 comparisons, with the majority of algorithms achieving around 90% accuracy, meeting industrial practicality. In Fig. 13 (e), recall rates across different algorithms are shown, with Transformer-based models exhibiting higher recall rates, underscoring the significant developmental potential of Transformers in the future.

D. SUMMARY

This section articulates evaluation metrics pertinent to PCB defect detection, explores publicly accessible datasets, and compares these metrics to the algorithms discussed in section II. Evaluation metrics afford an assessment of a model's strengths and weaknesses, with a higher FPS implying a more rapid model inference, a greater mAP and mAP@0.5 signifying superior accuracy, and a lower parameters value reflecting a reduced model parameter count. Within the realm of PCB defect detection datasets, we concentrate on several public collections that range in size from over a thousand to more than 10,000 images. These public datasets facilitate the establishment of a baseline, simplify performance comparison across models, and simultaneously, the higher quality datasets foster advancements in PCB defect

TABLE 5. Overview of PCB datasets.

Dataset	Information
PCB Defect	The dataset, released by the Human-Computer Interaction Open Lab at Peking University, is a synthesized PCB dataset comprising a total of 1,386 images. It encompasses six distinct defect categories: missing holes, mouse bites, open circuits, short circuits, protrusions, and irregular copper patterns. This dataset is applicable for tasks related to detection, classification, and registration. The dataset can be accessed for download at the following address: http://robotics.pkusz.edu.cn/resources/dataset/.
PCB Defect-Augmented [8]	The dataset represents an enhanced version of the PCB Defect dataset, encompassing a total of 10,668 images along with corresponding annotation files. The original high-resolution images in the dataset were cropped into 600×600 sub-images and divided into a training set (9,920 images) and a test set (2,508 images). The dataset can be accessed and downloaded from the following link: https://www.dropbox.com/s/h0f39nyotddibsb/VOC_PCB.zip?dl=0.
DEEP PCB [115]	The DEEP PCB dataset comprises 1,500 pairs of images, with each pair consisting of an intact template image and an aligned test image. The annotations for the test images encompass six prevalent PCB defect types: open circuit, short circuit, mouse bite, protrusion, pinhole, and spurious copper. The dataset can be accessed and downloaded from the following link: https://github.com/tangsanli5201/DeepPCB.
HRIPCB [116]	HRIPCB: This dataset constitutes a synthesized collection of 1,386 PCB images, encompassing six distinct defect categories. The primary utilization of this dataset revolves around tasks related to PCB defect detection, classification, and registration. Within this dataset, a reference-based approach is employed for defect detection, coupled with an end-to-end CNN for defect classification, commonly referred to as the RBCNN method. The provision of this dataset furnishes researchers with an extensive resource to facilitate their exploration of pertinent research avenues. The dataset can be accessed and downloaded through the following link: http://robotics.pkusz.edu.cn/resources/dataset/.
Micro-PCB	The dataset comprises a collection of 8,125 high-resolution images representing 13 micro-PCBs. These images have an average dimension of 1949×2126 pixels (width×height). Captured under optimal lighting conditions, the micro-PCBs were photographed from 25 distinct camera angles. At each angle, they were captured in 5 different rotations, yielding 125 unique orientations per micro-PCB. Each of these orientations was photographed four times for training purposes. Additionally, a single micro-PCB of the same make and model was photographed once and used for testing. This ensures that no micro-PCB used for training is repeated in testing. While the micro-PCBs used for training closely resemble those used for testing, minute distinctions can be observed in certain cases. Overall, the dataset encompasses 500 training images and 125 testing images for each micro-PCB, resulting in a train/test split ratio of 6,500/1,625. The dataset can be accessed and downloaded from the following link: https://www.kaggle.com/datasets/frettapper/micropcb-images.



(a)

(b)

(c)

(d)

(e)

FIGURE 13. Comparison of performance indicators in various references: (a) FPS, (b) mAP, (c) Parameters, (d) mAP@0.5, (d) Recall.

detection. In the comparison of results, we enumerate the backbone, FPS, mAP, mAP@0.5, parameters, recall, and the dataset used in each paper for every model. Utilizing this data, we produce a suite of statistical charts, providing a clear visualization of each model's strengths and weaknesses. However, due to the use of varying datasets, these comparative results do not directly adjudicate the models' performance.

TABLE 6. Multiple result comparison of the above reference algorithm used in PCB defect detection.

Reference	BACKBONE	FPS	mAP@0.5	mAP	Parameters (MB)	Recall	Dataset
Xin. [71] (2021)	CSPDarknet-53	-	96.88%	-	-	-	PCB Defect
Zhang, [73] (2021)	MobileNet-V2	25.2		44.3%	4.42	-	The unpublished dataset consists of 1,455 training images and 624 test images, encompassing six distinct defect types.
Jiang, [76] (2022)	VGG-16	50	96.46%	-	92.65	85.59%	The undisclosed dataset comprises 2,010 annotated instances across five distinct defect categories, with each category containing 402 annotations.
Li, [77] (2022)	CSPDarknet-53	-	89.86%	-	61.73	-	The undisclosed dataset comprises 9,636 images, all sized uniformly at 4092x3000 pixels, encompassing a diverse range of 21 categories.
Wan, [81] (2022)	CSPDarknet-53	-	98.7%	-	-	-	DEEP PCB [116]
Wu, [82] (2022)	CSPDarknet-53	-	96.5%	50.7%	47.4	-	The undisclosed dataset comprises 520 training images, 150 validation images, and 23 test images, encompassing six distinct defect types.
Xuan, [83] (2022)	MC（CSPDarknet-53 and inverted residual block）	47.6	99.13%	-	3.79	-	The undisclosed dataset comprises 2,654 original images, encompassing seven distinct defect types.
Zhao, [84] (2022)	CSPDarknet-53	-	89.5%	-	13.8	83.4%	Provided by the Intelligent Robotics Open Laboratory at Peking University, the dataset comprises 693 images, spanning six distinct defect types.
Zheng, [87] (2022)	MobileNet-V2	-	99.60%	-	-	99.32%	The dataset is openly accessible and encompasses 700 images with a resolution of 2300x2300, categorized into four distinct types. It can be accessed at: https://robotics.pkusz.edu.cn/resources/dataset/.
Lim, [88] (2023)	CSPDarknet-53	90	99.17%	81.2%	-	-	PCB Defect-Augmented [8]
Yu, [89] (2023)	DFP-Net	61	98.9%	-	69.3	-	PCB Defect-Augmented [8]
Li, [90] (2018)	VGG-16	-	-	60.63%	-	-	The dataset, which remains confidential, comprises a total of 22,765 images.
Ding, [8] (2019)	ResNet-101	-	98.90%	-	-	-	Provided by the Intelligent Robotics Open Laboratory at Peking University, the dataset comprises 693 images, spanning six distinct defect types.
Hu, [92] (2020)	ResNet-50	-	94.2%	-	-	-	Provided by the Intelligent Robotics Open Laboratory at Peking University, the dataset comprises 693 images, spanning six distinct defect types.
Li, [94] (2020)	ResNet-101	-	96.3%	-	-	-	The undisclosed dataset comprises 1,540 images with a resolution of 985x825 pixels.
An, [95] (2022)	ViT	-	98.8%	95.5%	-	-	Micro-PCB
Chen, [96] (2022)	Swin Transformer	21	97.04	-	93.95	-	PCB Defect-Augmented [8]
Yang, [97] (2023)	SwinV2_TDD-YOLOv7		98.74	53.52%	-	99.49%	PCB Defect-Augmented [8]

V. DISCUSSION

In delving into the domain of PCB defect detection, ample room for development in this field becomes evident. Despite the existence of numerous exceptional algorithms, PCB images exhibit distinct dissimilarities compared to natural images, presenting unique challenges in defect detection. Moreover, PCB defects display prominent characteristics. Firstly, PCB images possess intricate structures and intricate details, stemming from the complex nature of circuit boards. These images contain a multitude of elements such as lines, solder points, and components, with intricate interconnections, demanding algorithms capable of comprehending and analyzing complex structures. Secondly, PCB defects frequently exhibit diversity and variations, with different defect types and locations manifesting distinctive features and forms in PCB images [117]. For instance, solder point defects can encompass scenarios like open soldering, short circuits, and displacement, while line defects may involve issues such as circuit breaks and wire corrosion. Furthermore, PCB images often suffer from noise and other interference factors due to the complexity of production environments and potential random factors during image acquisition. These interferences include noise, shadows, and variations in lighting conditions. Lastly, efficient and rapid PCB defect detection is imperative. In practical production settings, the urgency to fulfill production line requirements mandates the timely completion of PCB defect detection [89].

In the face of these challenges, we acknowledge that traditional methods have gradually demonstrated limited capabilities in addressing these issues. The steps are different between traditional machine vision methods and deep learning based methods. Conventional machine vision methods necessitate an initial pre-processing of the image to enhance its quality. Subsequently, the PCB board is isolated via image segmentation, followed by the extraction of features, such as shape, texture, color, and others, indicative of PCB defects within the image. The location of the defect in the PCB image is then identified and classified using machine learning algorithms, and the detection results undergo post-processing. In contrast, PCB defect detection grounded in deep learning can learn from substantial or even massive volumes of data. This method automatically discerns various levels of image features through CNNs or transformers, eliminating the need for manual image feature design and extraction. Deep learning employs an end-to-end training approach, treating feature extraction and classification or regression as an integrated process. This eliminates the requirement for a complex machine vision processing workflow. However, it is precisely behind these challenges that deep learning-based approaches begin to manifest remarkable superior performance. Relative to traditional methods, deep learning methods exhibit multiple advantages stemming from their exceptional capabilities in image processing and pattern recognition. Firstly, deep learning methods possess the capability of automated feature extraction, eliminating the cumbersome manual feature engineering process. In contrast, traditional methods often require significant time and effort to design manual feature extractors to capture key information in images. Secondly, deep learning methods demonstrate enhanced adaptability [4]. Given the diversity in PCB defect types and shapes, traditional methods struggle to encompass all defect variations. Furthermore, deep learning methods exhibit higher accuracy. Through training on extensive datasets, deep learning models learn from numerous samples, thereby enhancing detection accuracy. Conversely, traditional methods may be constrained by specific rules and limitations, preventing them from reaching the accuracy levels of deep learning models [77]. Lastly, deep learning methods can handle complex relationships. In PCB images, multiple defects may coexist with intricate interdependencies and interactions. Deep learning models adeptly capture these intricate relationships, resulting in more accurate detection of multiple defects.

However, despite the significant achievements of deep learning in PCB defect detection, certain methods still exhibit more remarkable potential for future development. Firstly, methods based on CNNs offer substantial advantages in detection speed and accuracy. Employing CNN for feature extraction and classification of PCB images facilitates the effective detection of common defect types.

Furthermore, by enhancing network structures and augmenting sample data, continuous improvements in detection accuracy can be achieved [92]. Secondly, transfer learning methods can leverage pre-trained models like ImageNet for fine-tuning, and adapting to the unique features of PCB images. This approach reduces the demand for training samples, enhancing both training speed and detection effectiveness. Additionally, harnessing data augmentation through GANs can provide diverse PCB defect images for deep learning algorithms, thus bolstering model robustness. Lastly, PCB defect detection algorithms based on Transformers show promise. Although Transformer-based models exhibit exceptional performance in the field of CV, they are accompanied by considerable computational demands and sample requirements, limiting real-time detection advantages. With the continuous evolution of Transformer technology, models with comparable speed and accuracy to CNN have emerged. Given that Transformer-based models do not require non-maximum suppression (NMS) during inference, we hold confidence in their substantial potential for development.

Through an in-depth exploration of the PCB defect detection domain, we have not only recognized its vast potential for development but also gained a profound appreciation for the remarkable performance of deep learning methods within it. Among various approaches, CNNs, transfer learning, GANs, and Transformers exhibit substantial prospects. Particularly in the realm of PCB defect detection, deep learning methods stand out with their exceptional performance, addressing the limitations of traditional methods and demonstrating noteworthy advantages in automated feature extraction, adaptability, accuracy, and complex relationship handling. Despite these significant advancements, the field of PCB defect detection still confronts a spectrum of difficulties and challenges. As we look towards future development, it is evident that a mixture of challenges and opportunities lies ahead.

In addition to visible light detection, infrared imaging detection is also widely used in industry. Understanding the heat transfer mechanism is crucial for infrared detection. By deeply understanding the heat transfer model, the advantages and disadvantages of infrared imaging in various defect detection and classification applications can be analyzed more thoroughly [118]. At the same time, revealing the heat transfer mechanism helps to further improve the accuracy and applicability of infrared imaging. For example, the study of various heat transfer models, including electrical discharge machining (EDM) heat transfer [119], [120], laser heat transfer [121], [122], and wire-EDM heat transfer [123], [124], helps to have a more comprehensive understanding of the mechanism of infrared imaging. Furthermore, considering the differences in thermal gradients may be helpful for detecting defects in infrared imaging.

VI. OUTLOOK

In recent years, technologies based on deep learning have demonstrated remarkable capabilities across various domains, particularly in the realm of image processing. This comprehensive review aims to investigate and analyze the application of deep learning techniques to the detection of defects in PCBs, exploring their potential, methodologies, challenges, and future directions.

(1) Enormous Potential of Deep Learning in PCB Defect Detection: Deep learning models, such as CNNs and Transformer models, have made significant strides in PCB defect detection owing to their robust feature learning and representation capabilities. CNNs employ stacked convolution and pooling operations to learn local and global features from raw images, progressively abstracting and comprehending image content. While Transformer models have

achieved success in natural language processing, they have also demonstrated applicability in image processing tasks. These models leverage self-attention mechanisms to capture global information while retaining sensitivity to local details, providing substantial support for PCB defect detection.

(2) Diverse Detection Approaches: Deep learning-based PCB defect detection methods are categorized into one-stage and two-stage algorithms. One-stage algorithms directly predict defect locations and categories in raw images, offering real-time advantages. Two-stage algorithms first extract candidate regions from images and then classify these regions, achieving more precise defect localization. Additionally, Transformer-based methods have shown promising results in PCB defect detection by transforming image data into sequential data for processing, capturing features at different scales and levels.

(3) Significance of Evaluation: In PCB defect detection tasks, the choice of appropriate evaluation metrics is crucial for assessing model accuracy and stability. Common evaluation metrics include precision, recall, accuracy, and F1 score. Moreover, selecting suitable PCB defect datasets for training and testing is vital to ensuring model generalization. Thoughtful dataset selection better reflects real-world application scenarios and more accurately evaluates model performance.

(4) Overcoming Challenges and Issues: Despite the notable progress of deep learning in PCB defect detection, challenges persist. Firstly, PCB defects exhibit diverse complexity, posing challenges for designing robust deep learning models to detect various defects. Different defect types may possess distinct features and morphologies, making enhancing generalization a formidable task. Secondly, data imbalance and scarcity of defect samples impact model training and generalization, necessitating effective strategies for mitigation. Lastly, enhancing model interpretability and explainability is a pressing research direction in PCB defect detection, particularly for applications demanding rigorous model decisions.

(5) Future Development Prospects: Deep learning-based methods retain advantages over traditional methods. Deep learning models autonomously learn features from data, reducing the reliance on manual feature engineering. With hardware advancements and the accumulation of large-scale data, the potential of deep learning models becomes more pronounced. Hence, future research can focus on improving the robustness and generalization capabilities of deep learning models, exploring more effective data augmentation and defect sample generation methods to address challenges related to diversity and data scarcity. Meanwhile, employing crack analysis in tandem with deep learning is crucial for detecting PCB defects. Specifically, the utilization of neural networks in model training accurately identifies minuscule cracks, often imperceptible to the human eye, on PCB surfaces. This application is invaluable for early detection of potential issues, contributing significantly to ensuring product quality during production. Integration of deep learning with crack analysis enhances detection precision and streamlines quality control procedures. Additionally, increasing model interpretability and explainability, enhancing model decision transparency, and incorporating techniques such as reinforcement learning, transfer learning, and generative adversarial networks can further elevate the performance of PCB defect detection.

In conclusion, deep learning-based PCB defect detection holds vast developmental prospects and application potential in the future. Technological advancements coupled with in-depth research will drive improvements in quality control and production efficiency within the PCB manufacturing industry.

REFERENCES

[1] Q. Ling and N. A. M. Isa, "Printed circuit board defect detection methods based on image processing, machine learning and deep learning: A survey," *IEEE Access*, vol. 11, pp. 15921–15944, 2023, doi: 10.1109/ACCESS.2023.3245093.

[2] A. F. M. Hani, A. S. Malik, R. Kamil, and C.-M. Thong, "A review of SMD-PCB defects and detection algorithms," *Proc. SPIE*, vol. 8350, pp. 373–379, Jan. 2012, doi: 10.1117/12.920531.

[3] J.-H. Park, Y.-S. Kim, H. Seo, and Y.-J. Cho, "Analysis of training deep learning models for PCB defect detection," *Sensors*, vol. 23, no. 5, p. 2766, Mar. 2023, doi: 10.3390/s23052766.

[4] Q. Zhang and H. Liu, "Multi-scale defect detection of printed circuit board based on feature pyramid network," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Jun. 2021, pp. 911–914, doi: 10.1109/ICAICA52286.2021.9498174.

[5] D. Makwana and S. Mittal, "PCBSegClassNet—A light-weight network for segmentation and classification of PCB component," *Exp. Syst. Appl.*, vol. 225, Sep. 2023, Art. no. 120029, doi: 10.1016/j.eswa.2023.120029.

[6] I. A. Soomro, A. Ahmad, and R. H. Raza, "Printed circuit board identification using deep convolutional neural networks to facilitate recycling," *Resour., Conservation Recycling*, vol. 177, Feb. 2022, Art. no. 105963, doi: 10.1016/j.resconrec.2021.105963.

[7] M. Liukkonen, E. Havia, and Y. Hiltunen, "Computational intelligence in mass soldering of electronics—A survey," *Exp. Syst. Appl.*, vol. 39, no. 10, pp. 9928–9937, Aug. 2012, doi: 10.1016/j.eswa.2012.02.100.

[8] R. Ding, L. Dai, G. Li, and H. Liu, "TDD-Net: A tiny defect detection network for printed circuit boards," *CAAI Trans. Intell. Technol.*, vol. 4, no. 2, pp. 110–116, Jun. 2019, doi: 10.1049/trit.2019.0019.

[9] X. Wu, Y. Ge, Q. Zhang, and D. Zhang, "PCB defect detection using deep learning methods," in *Proc. IEEE 24th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2021, pp. 873–876, doi: 10.1109/CSCWD49262.2021.9437846.

[10] F. Raihan and W. Ce, "PCB defect detection USING OPENCV with image subtraction method," in *Proc. Int. Conf. Inf. Manage. Technol. (ICIMTech)*, Nov. 2017, pp. 204–209, doi: 10.1109/ICIMTech.2017.8273538.

[11] A. Raj and A. Sajeena, "Defects detection in PCB using image processing for industrial applications," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Apr. 2018, pp. 1077–1079, doi: 10.1109/ICICCT.2018.8473285.

[12] M. Moganti, F. Ercal, C. H. Dagli, and S. Tsunekawa, "Automatic PCB inspection algorithms: A survey," *Comput. Vis. Image Understand.*, vol. 63, no. 2, pp. 287–313, Mar. 1996, doi: 10.1006/cviu.1996.0020.

[13] Y. Tang, M. Chen, C. Wang, L. Luo, J. Li, G. Lian, and X. Zou, "Recognition and localization methods for vision-based fruit picking robots: A review," *Frontiers Plant Sci.*, vol. 11, p. 510, May 2020, doi: 10.3389/fpls.2020.00510.

[14] W. He, Z. Jiang, W. Ming, G. Zhang, J. Yuan, and L. Yin, "A critical review for machining positioning based on computer vision," *Measurement*, vol. 184, Nov. 2021, Art. no. 109973, doi: 10.1016/j.measurement.2021.109973.

[15] W. Ming, F. Shen, X. Li, Z. Zhang, J. Du, Z. Chen, and Y. Cao, "A comprehensive review of defect detection in 3C glass components," *Measurement*, vol. 158, Jul. 2020, Art. no. 107722, doi: 10.1016/j.measurement.2020.107722.

[16] W. Ming, F. Shen, H. Zhang, X. Li, J. Ma, J. Du, and Y. Lu, "Defect detection of LGP based on combined classifier with dynamic weights," *Measurement*, vol. 143, pp. 211–225, Sep. 2019, doi: 10.1016/j.measurement.2019.04.087.

[17] W. Ming, C. Cao, G. Zhang, H. Zhang, F. Zhang, Z. Jiang, and J. Yuan, "Review: Application of convolutional neural network in defect detection of 3C products," *IEEE Access*, vol. 9, pp. 135657–135674, 2021, doi: 10.1109/ACCESS.2021.3116131.

[18] W. He, Z. Shi, Y. Liu, T. Liu, J. Du, J. Ma, Y. Cao, and W. Ming, "Feature fusion classifier with dynamic weights for abnormality detection of amniotic fluid cell chromosome," *IEEE Access*, vol. 11, pp. 31755–31766, 2023, doi: 10.1109/ACCESS.2023.3257045.

[19] W. He, Y. Han, W. Ming, J. Du, Y. Liu, Y. Yang, L. Wang, Y. Wang, Z. Jiang, C. Cao, and J. Yuan, "Progress of machine vision in the detection of cancer cells in histopathology," *IEEE Access*, vol. 10, pp. 46753–46771, 2022, doi: 10.1109/ACCESS.2022.3161575.

[20] A. Kamilaris and F. X. Prenafeta-Boldu, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018, doi: 10.1016/j.compag.2018.02.016.

[21] D. Shen, S. Zhang, W. Ming, W. He, G. Zhang, and Z. Xie, "Development of a new machine vision algorithm to estimate potato's shape and size based on support vector machine," *J. Food Process Eng.*, vol. 45, no. 3, Mar. 2022, Art. no. e13974, doi: 10.1111/jfpe.13974.

[22] Y.-Y. Zheng, J.-L. Kong, X.-B. Jin, X.-Y. Wang, and M. Zuo, "CropDeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture," *Sensors*, vol. 19, no. 5, p. 1058, Mar. 2019, doi: 10.3390/s19051058.

[23] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, K. Kavita, M. F. Ijaz, and M. Wozniak, "A survey of deep convolutional neural networks applied for prediction of plant leaf diseases," *Sensors*, vol. 21, no. 14, p. 4749, Jul. 2021, doi: 10.3390/s21144749.

[24] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016, doi: 10.1109/MGRS.2016.2540798.

[25] W. Pei, Z. Shi, and K. Gong, "Small target detection with remote sensing images based on an improved YOLOv5 algorithm," *Frontiers Neurorobotics*, vol. 16, Feb. 2023, Art. no. 1074862.

[26] H. Tian, X. Fang, Y. Lan, C. Ma, H. Huang, X. Lu, D. Zhao, H. Liu, and Y. Zhang, "Extraction of citrus trees from UAV remote sensing imagery using YOLOv5s and coordinate transformation," *Remote Sens.*, vol. 14, no. 17, p. 4208, Aug. 2022, doi: 10.3390/rs14174208.

[27] W. Ming, P. Sun, Z. Zhang, W. Qiu, J. Du, X. Li, Y. Zhang, G. Zhang, K. Liu, Y. Wang, and X. Guo, "A systematic review of machine learning methods applied to fuel cells in performance evaluation, durability prediction, and application monitoring," *Int. J. Hydrogen Energy*, vol. 48, no. 13, pp. 5197–5228, Feb. 2023, doi: 10.1016/j.ijhydene.2022.10.261.

[28] W. He, Z. Li, T. Liu, Z. Liu, X. Guo, J. Du, X. Li, P. Sun, and W. Ming, "Research progress and application of deep learning in remaining useful life, state of health and battery thermal management of lithium batteries," *J. Energy Storage*, vol. 70, Oct. 2023, Art. no. 107868, doi: 10.1016/j.est.2023.107868.

[29] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[32] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.* Curran Associates, 2012, pp. 1–9. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[34] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. IEEE IWAENC*, vol. 15, Jun. 2011, pp. 315–323.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014, *arXiv:1409.4842*.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[39] N. Cai, Q. Ye, G. Liu, H. Wang, and Z. Yang, "IC solder joint inspection based on the Gaussian mixture model," *Soldering Surf. Mount Technol.*, vol. 28, no. 4, pp. 207–214, Sep. 2016, doi: 10.1108/ssmt-03-2016-0005.

[40] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," 2014, *arXiv:1406.6247*.

[41] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[42] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[44] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2017, *arXiv:1709.01507*.

[45] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," 2018, *arXiv:1807.06521*.

[46] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," 2019, *arXiv:1910.03151*.

[47] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," 2021, *arXiv:2103.02907*.

[48] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[49] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," 2020, *arXiv:2012.12877*.

[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[51] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," 2018, *arXiv:1803.02155*.

[52] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Proc. Adv. Neural Inf. Process. Syst* Curran Associates, 2019., pp. 1–13. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/3416a75f4cea9109507cacd8e2f2aefc-Abstract.html

[53] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," May 2020, *arXiv:2005.12872*.

[54] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16. [Online]. Available: https://xueshu.baidu.com/usercenter/paper/show?paperid=1p7m006063070v10ua310e20pd632729&site=xueshu_se

[55] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, "Swin transformer v2: Scaling up capacity and resolution," 2021, *arXiv:2111.09883*.

[56] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," 2019, *arXiv:1904.09925*.

[57] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020, doi: 10.1007/s11263-019-01247-4.

[58] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[59] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[60] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[61] 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, doi: 10.1109/TPAMI.2016.2577031.

[62] 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, doi: 10.1109/ICCV.2017.322.

[63] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ''SSD: Single shot MultiBox detector,'' in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[64] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ''You only look once: Unified, real-time object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[65] J. Redmon and A. Farhadi, ''YOLO9000: Better, faster, stronger,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

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

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

[68] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, ''YOLOv6: A single-stage object detection framework for industrial applications,'' 2022, *arXiv:2209.02976*.

[69] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, ''YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,'' 2022, *arXiv:2207.02696*.

[70] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, ''Object detection in 20 years: A survey,'' *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023, doi: 10.1109/JPROC.2023.3238524.

[71] H. Xin, Z. Chen, and B. Wang, ''PCB electronic component defect detection method based on improved YOLOv4 algorithm,'' *J. Phys., Conf.*, vol. 1827, no. 1, Mar. 2021, Art. no. 012167, doi: 10.1088/1742-6596/1827/1/012167.

[72] D. Misra, ''Mish: A self regularized non-monotonic activation function,'' 2019, *arXiv:1908.08681*.

[73] Y. Zhang, F. Xie, L. Huang, J. Shi, J. Yang, and Z. Li, ''A lightweight one-stage defect detection network for small object based on dual attention mechanism and PAFPN,'' in *Proc. AIP. Conf.*, vol. 9, Oct. 2021, Art. no. 708097, doi: 10.3389/fphy.2021.708097.

[74] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ''Focal loss for dense object detection,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[75] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, ''Feature pyramid networks for object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[76] W. Jiang, S. Zhang, and W. Chen, ''Multi-target detection of PCB defects based on improved SSD,'' *Int. Core J. Eng.*, vol. 8, no. 6, pp. 319–325, 2022, doi: 10.6919/ICJE.202206_8(6).0045.

[77] J. Li, W. Li, Y. Chen, and J. Gu, ''Research on object detection of PCB assembly scene based on effective receptive field anchor allocation,'' *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–32, Feb. 2022, doi: 10.1155/2022/7536711.

[78] W. Luo, Y. Li, R. Urtasun, and R. Zemel, ''Understanding the effective receptive field in deep convolutional neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.* Curran Associates, 2016, pp. 1–9. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/c8067ad1937f728f51288b3eb986afaa-Abstract.html

[79] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, ''DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[80] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, ''Rethinking atrous convolution for semantic image segmentation,'' 2017, *arXiv:1706.05587*.

[81] Y. Wan, L. Gao, X. Li, and Y. Gao, ''Semi-supervised defect detection method with data-expanding strategy for PCB quality inspection,'' *Sensors*, vol. 22, no. 20, p. 7971, Oct. 2022, doi: 10.3390/s22207971.

[82] L. Wu, L. Zhang, and Q. Zhou, ''Printed circuit board quality detection method integrating lightweight network and dual attention mechanism,'' *IEEE Access*, vol. 10, pp. 87617–87629, 2022, doi: 10.1109/ACCESS.2022.3198994.

[83] W. Xuan, G. Jian-She, H. Bo-Jie, W. Zong-Shan, D. Hong-Wei, and W. Jie, ''A lightweight modified YOLOX network using coordinate attention mechanism for PCB surface defect detection,'' *IEEE Sensors J.*, vol. 22, no. 21, pp. 20910–20920, Nov. 2022, doi: 10.1109/JSEN.2022.3208580.

[84] Y. Zhao, H. Yang, and H. Feng, ''An improved YOLOv5 PCB defect detection,'' *Proc. SPIE*, vol. 12351, pp. 380–387, Nov. 2022, doi: 10.1117/12.2652341.

[85] S. Liu, D. Huang, and Y. Wang, ''Learning spatial fusion for single-shot object detection,'' 2019, *arXiv:1911.09516*.

[86] Y. Liu, Z. Shao, and N. Hoffmann, ''Global attention mechanism: Retain information to enhance channel-spatial interactions,'' 2021, *arXiv:2112.05561*.

[87] J. Zheng, X. Sun, H. Zhou, C. Tian, and H. Qiang, ''Printed circuit boards defect detection method based on improved fully convolutional networks,'' *IEEE Access*, vol. 10, pp. 109908–109918, 2022, doi: 10.1109/ACCESS.2022.3214306.

[88] J. Lim, J. Lim, V. M. Baskaran, and X. Wang, ''A deep context learning based PCB defect detection model with anomalous trend alarming system,'' *Results Eng.*, vol. 17, Mar. 2023, Art. no. 100968, doi: 10.1016/j.rineng.2023.100968.

[89] Z. Yu, Y. Wu, B. Wei, Z. Ding, and F. Luo, ''A lightweight and efficient model for surface tiny defect detection,'' *Int. J. Speech Technol.*, vol. 53, no. 6, pp. 6344–6353, Mar. 2023, doi: 10.1007/s10489-022-03633-x.

[90] Y. T. Li and J. I. Guo, ''A VGG-16 based faster RCNN model for PCB error inspection in industrial AOI applications,'' in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, May 2018, pp. 1–2, doi: 10.1109/ICCE-China.2018.8448674.

[91] A. Shrivastava, A. Gupta, and R. Girshick, ''Training region-based object detectors with online hard example mining,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 761–769, doi: 10.1109/CVPR.2016.89.

[92] B. Hu and J. Wang, ''Detection of PCB surface defects with improved faster-RCNN and feature pyramid network,'' *IEEE Access*, vol. 8, pp. 108335–108345, 2020, doi: 10.1109/ACCESS.2020.3001349.

[93] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, ''Region proposal by guided anchoring,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 2960–2969, doi: 10.1109/CVPR.2019.00308.

[94] D. Li, S. Fu, Q. Zhang, Y. Mo, L. Liu, and C. Xu, ''An improved PCB defect detector based on feature pyramid networks,'' in *Proc. 4th Int. Conf. Comput. Sci. Artif. Intell.*, Zhuhai China, Dec. 2020, pp. 233–239, doi: 10.1145/3445815.3445853.

[95] K. An and Y. Zhang, ''LPViT: A transformer based model for PCB image classification and defect detection,'' *IEEE Access*, vol. 10, pp. 42542–42553, 2022, doi: 10.1109/ACCESS.2022.3168861.

[96] W. Chen, Z. Huang, Q. Mu, and Y. Sun, ''PCB defect detection method based on transformer-YOLO,'' *IEEE Access*, vol. 10, pp. 129480–129489, 2022, doi: 10.1109/ACCESS.2022.3228206.

[97] Y. Yang and H. Kang, ''An enhanced detection method of PCB defect based on improved YOLOv7,'' *Electronics*, vol. 12, no. 9, p. 2120, May 2023, doi: 10.3390/electronics12092120.

[98] W. Wang, L. Yao, L. Chen, B. Lin, D. Cai, X. He, and W. Liu, ''CrossFormer: A versatile vision transformer hinging on cross-scale attention,'' 2021, *arXiv:2108.00154*.

[99] W. Wang, W. Chen, Q. Qiu, L. Chen, B. Wu, B. Lin, X. He, and W. Liu, ''CrossFormer++: A versatile vision transformer hinging on cross-scale attention,'' 2023, *arXiv:2303.06908*.

[100] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, ''Vision transformer adapter for dense predictions,'' 2022, *arXiv:2205.08534*.

[101] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, ''The PASCAL visual object classes (VOC) challenge,'' *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.

[102] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, ''Generalized intersection over union: A metric and a loss for bounding box regression,'' 2019, *arXiv:1902.09630*.

[103] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012, doi: 10.1109/TPAMI.2011.155.

[104] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[105] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," 2019, *arXiv:1911.08287*.

[106] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," 2021, *arXiv:2101.08158*.

[107] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, and X.-S. Hua, "Alpha-IoU: A family of power intersection over union losses for bounding box regression," 2021, *arXiv:2110.13675*.

[108] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.

[109] Z. Liu, Y. Gao, Q. Du, M. Chen, and W. Lv, "YOLO-extract: Improved YOLOv5 for aircraft object detection in remote sensing images," *IEEE Access*, vol. 11, pp. 1742–1751, 2023, doi: 10.1109/ACCESS.2023.3233964.

[110] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016, doi: 10.1016/j.isprsjprs.2016.03.014.

[111] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng, "R2-CNN: Fast tiny object detection in large-scale remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5512–5524, Aug. 2019, doi: 10.1109/TGRS.2019.2899955.

[112] J. Zhang, J. Lei, W. Xie, Z. Fang, Y. Li, and Q. Du, "SuperYOLO: Super resolution assisted object detection in multimodal remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5605415, doi: 10.1109/TGRS.2023.3258666.

[113] K. Fu, Z. Chang, Y. Zhang, G. Xu, K. Zhang, and X. Sun, "Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 161, pp. 294–308, Mar. 2020, doi: 10.1016/j.isprsjprs.2020.01.025.

[114] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang, "A comparative study of real-time semantic segmentation for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 700–70010, doi: 10.1109/CVPRW.2018.00101.

[115] S. Tang, F. He, X. Huang, and J. Yang, "Online PCB defect detector on a new PCB defect dataset," 2019, *arXiv:1902.06197*.

[116] W. Huang, P. Wei, M. Zhang, and H. Liu, "HRIPCB: A challenging dataset for PCB defects detection and classification," *J. Eng.*, vol. 2020, no. 13, pp. 303–309, Jul. 2020, doi: 10.1049/joe.2019.1183.

[117] V. U. Sankar, G. Lakshmi, and Y. S. Sankar, "A review of various defects in PCB," *J. Electron. Test.*, vol. 38, no. 5, pp. 481–491, Oct. 2022, doi: 10.1007/s10836-022-06026-7.

[118] X. Zhao, Y. Zhao, S. Hu, H. Wang, Y. Zhang, and W. Ming, "Progress in active infrared imaging for defect detection in the renewable and electronic industries," *Sensors*, vol. 23, no. 21, p. 8780, Oct. 2023, doi: 10.3390/s23218780.

[119] W. Ming, Z. Xie, J. Ma, J. Du, G. Zhang, C. Cao, and Y. Zhang, "Critical review on sustainable techniques in electrical discharge machining," *J. Manuf. Processes*, vol. 72, pp. 375–399, Dec. 2021, doi: 10.1016/j.jmapro.2021.10.035.

[120] W. Ming, S. Zhang, G. Zhang, J. Du, J. Ma, W. He, C. Cao, and K. Liu, "Progress in modeling of electrical discharge machining process," *Int. J. Heat Mass Transf.*, vol. 187, May 2022, Art. no. 122563, doi: 10.1016/j.ijheatmasstransfer.2022.122563.

[121] Z. Zhang, W. Qiu, G. Zhang, D. Liu, and P. Wang, "Progress in applications of shockwave induced by short pulsed laser on surface processing," *Opt. Laser Technol.*, vol. 157, Jan. 2023, Art. no. 108760, doi: 10.1016/j.optlastec.2022.108760.

[122] P. Wang, Z. Zhang, B. Hao, S. Wei, Y. Huang, and G. Zhang, "Investigation on heat transfer and ablation mechanism of CFRP by different laser scanning directions," *Compos. B, Eng.*, vol. 262, Aug. 2023, Art. no. 110827, doi: 10.1016/j.compositesb.2023.110827.

[123] Y. Chen, S. Hu, A. Li, Y. Cao, Y. Zhao, and W. Ming, "Parameters optimization of electrical discharge machining process using swarm intelligence: A review," *Metals*, vol. 13, no. 5, p. 839, Apr. 2023, doi: 10.3390/met13050839.

[124] G. Zhang, Z. Chen, Z. Zhang, Y. Huang, W. Ming, and H. Li, "A macroscopic mechanical model of wire electrode deflection considering temperature increment in MS-WEDM process," *Int. J. Mach. Tools Manuf.*, vol. 78, pp. 41–53, Mar. 2014, doi: 10.1016/j.ijmachtools.2014.01.004.

XING CHEN was born in Hubei, in 1979. He received the master's degree from the Huazhong University of Science and Technology, in 2008. He is currently a Teacher with Nanyang Normal University. He has published more than ten papers in academic journals, three papers in EI search, presided over two projects of the Ministry of Education, and guided students to win the first prize of Henan Province in their graduation papers. His research interests include intelligent control and security.

YONGLEI WU was born in Henan, China, in 2001. He is currently pursuing the master's degree with the College of Mechanical and Electrical Engineering, Zhengzhou University of Light Industry. His research interests include machine learning and deep learning. He has won the silver medal in the CCPC Henan Division, the second prize in the Blue Bridge Cup National Finals, and the second prize in the Henan Division of the National College Mathematics Competition.

XINGYOU HE was born in Henan, China, in 1999. He is currently pursuing the master's degree with the College of Mechanical and Electrical Engineering, Zhengzhou University of Light Industry. He has won the first prize in the final of the 2021 iCAN National Undergraduate Innovation and Entrepreneurship Competition and the second prize in the 2021 "Internet Plus" Undergraduate Innovation and Entrepreneurship Competition and the Seventh China National Standard "Internet Plus" Undergraduate Innovation and Entrepreneurship Competition in Henan.

WUYI MING was born in Hubei, China, in 1981. He received the bachelor's degree in engineering from Huazhong Agricultural University, in 2002, the master's degree in engineering from Zhengzhou University, in 2010, and the Ph.D. degree in engineering from the Huazhong University of Science and Technology, in 2014.

He has done the theoretical and applied research in EDM and deep learning. He is currently with the Guangdong Provincial Key Laboratory of Manufacturing Equipment Digitization. He has published nearly 100 SCI indexed articles (more than 1700 citations, H27) in journals, such as *Journal of Cleaner Production*, *Applied Surface Science*, *Journal of Materials Processing Technology*, *Measurement*, and IEEE ACCESS. His main research interests include digital technology and equipment for difficult-to-process materials. He won the Second Prize of the Henan (China) Science and Technology Progress Award, in 2018.