WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

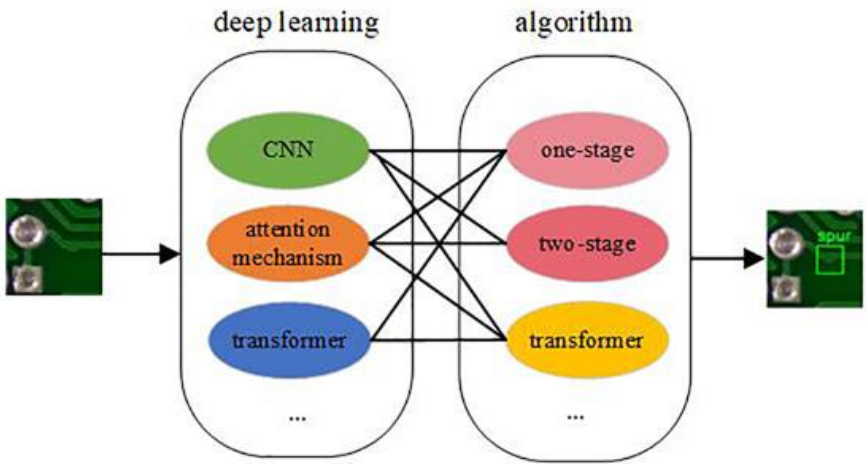
A Comprehensive Study of PCB Defect Detection Via Deep Learning

Mr. Nakul Ashok Gade[1], Ms. Priya Gulabrao Deshmukh[2], Mr. Sachin Ashok Surywanshi[3], Ms. Ankita Kamlakar Pangavhane[4]

[1] Assistant Professor MVPS's KBTCOE Nashik
[2] Lecturer MVPS's RSM POLY, Nashik
[3] Lecturer MVPS's RSM POLY, Nashik
[4] Lecturer MVPS's RSM POLY, Nashik

ABSTRACT:

Electronic components must be connected and secured using a printed circuit board (PCB), which serves as a substrate. Its extensive incorporation can be seen in a variety of contemporary electronic devices, including computers, smartphones, televisions, digital cameras, and other equipment. Defect detection is made more difficult by the increased accuracy of modern circuit boards, which makes it necessary to do thorough defect inspection in order to ensure product quality. Due to their limited accuracy and inefficiency, conventional algorithms are unable to meet usage benchmarks. On the other hand, deep learning-based PCB defect detection algorithms have the potential to achieve higher accuracy and efficiency due to their ability to identify new types of defects. This review explores the domains of machine learning and deep learning to provide a thorough examination of machine vision-based PCB defect detection algorithms. It begins by placing these algorithms in context and explaining their importance, then delves deeply into how they have developed within the machine vision framework, including categorization, comparison, and analysis of algorithmic principles, strengths, and shortcomings. Furthermore, the evaluation of algorithmic performance is improved by the addition of commonly used PCB fault detection datasets and assessment indices. At an Intersection over Union (IoU) of 0.5, the detection accuracy can currently surpass 95%. Finally, several avenues for future research are noted in order to tackle the problems with the current method. In order to improve PCB defect detection performance, these directions include using Transformers as a basic framework for developing new algorithms and applying strategies like reinforcement learning and Generative Adversarial Networks (GANs). Because PCBs are so small, it might be difficult to find flaws within them. However, improvements in deep learning approaches have greatly improved defect identification using deep learning-based methods. The algorithms used over the last 10 years, including CNN, attention mechanism, transformer, and hybrid approaches combining these approaches, are thoroughly compiled and examined in this paper.



1. INTRODUCTION :

In recent years, the utilization and popularity of electronic products have surged, largely due to the rapid growth of the global economy and the swift advancements in information technology. As electronic products undergo continuous upgrades and enhancements, the demands for performance and quality in printed circuit boards (PCBs) are increasing. Serving as essential foundations and critical components in electronic devices, PCBs must exhibit strong stability, significant resistance to interference, and excel in attributes such as high-speed transmission, increased integration levels, and compact size. Additionally, the layout of PCBs involves the strategic organization of electronic components, connection lines, holes, and other specific elements, representing a vital phase in the manufacturing of electronic devices. Important aspects of the PCB layout process include:

Layout: This entails the careful planning and arrangement of electronic components, connection lines, holes, and other specific elements on a PCB. The quality of the layout has a direct impact on circuit performance, electromagnetic interference (EMI), thermal efficiency, and maintenance considerations.

Component placement is a critical stage in the layout process, involving the strategic positioning of electronic components, including chips, resistors, capacitors, and connectors, on the printed circuit board (PCB). Effective component placement is vital for achieving optimal circuit performance and ensuring efficient signal transmission.

Routing follows the completion of the layout, where the wiring paths are established to connect the various components and create the circuit. This process necessitates careful consideration of factors such as signal integrity, power supply, grounding, and signal transmission lines.

Power distribution is another essential aspect of the layout process, requiring a well-thought-out plan for the allocation of power and ground lines. This ensures that electronic components receive a reliable power supply while minimizing noise and interference within the circuit.

Consequently, PCB layout represents a crucial step in the design and production of electronic products. It requires engineers to simultaneously consider various factors, including performance, heat dissipation, and maintainability, to achieve a logical arrangement and connection of electronic components. This meticulous planning guarantees that the PCB adheres to design specifications and supports efficient manufacturing. Therefore, the detection of defects and the implementation of quality control measures in PCB manufacturing and production processes are of utmost importance.

However, as component sizes decrease and density increases, along with the complexity and variety of PCB manufacturing, these boards become vulnerable to numerous factors such as mechanical wear, electrostatic interference, and chemical corrosion during production. These issues can lead to a range of defects, including missing holes, mouse bites, open circuits, shorts, spurs, spurious copper, and broken holes. Such defects significantly compromise the quality and performance of PCBs.

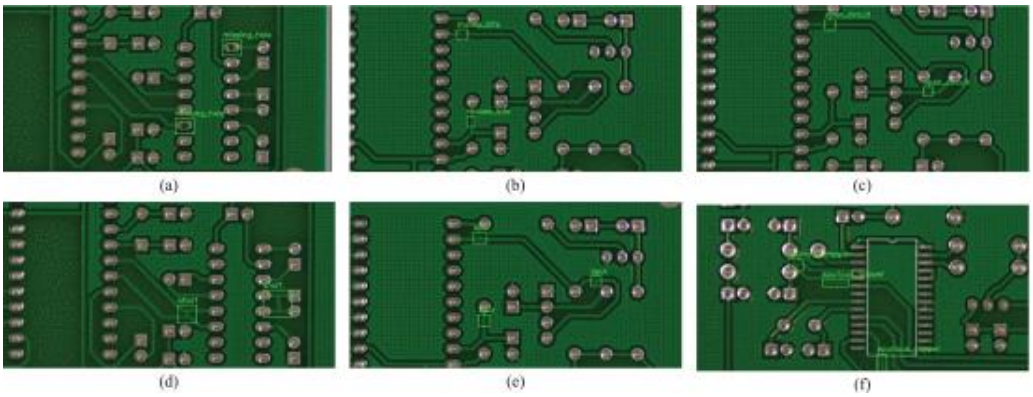


FIGURE 1. - Variety of PCB defect types: (a) Missing hole [8]; (b) Mouse bite [8]; (c) Open circuit [8]; (d) Short [8]; (e) Spur [8]; (f) Spurious copper [8].

Currently, the identification of PCB defects is a significant issue in the electronics manufacturing industry. A variety of inspection techniques have been researched and developed to tackle different types of PCB defects and meet the requirements of industrial production. These techniques include the functional test method, visual inspection technology, instrument on-line inspection method, and manual visual subjective assessment. The functional test method employs a fault simulator to evaluate the functionality of the circuit board and detect potential defects. While this method is known for its reliability and accuracy, it requires specialized equipment and complex testing procedures, making it a labor-intensive and time-consuming process. Visual inspection technology predominantly utilizes artificial intelligence algorithms, incorporating image processing and pattern recognition techniques for the rapid and automated detection of PCB defects. However, its success is contingent upon continuous optimization and improvement of the detection algorithms. The instrument on-line detection method is primarily used to monitor PCB defects through various instruments, such as high-voltage detection and insulation hydrocarbon detection methods. This approach offers the advantages of ease of use and quick detection, although it can sometimes encounter issues with misjudgment. The artificial visual subjective judgment method relies mainly on experiential and perceptual evaluations, making it vulnerable to human bias; nonetheless, it is applicable in certain specialized inspection contexts.

Despite the variety of methods available for detecting defects in printed circuit boards (PCBs), each technique has its own limitations and shortcomings. The choice of the most appropriate method should be tailored to meet specific needs and conditions. In recent years, driven by ongoing advancements and the integration of technology, deep learning algorithms have increasingly gained traction across various sectors. In the industrial realm, these algorithms are utilized for processing and positioning tasks, while also demonstrating the capability to detect defects in consumer electronics. In the medical sector, deep learning algorithms assist healthcare professionals in interpreting medical images for diagnostic purposes. In agriculture, they play a role in monitoring crop health. Furthermore, in the military sector, these algorithms analyze remote sensing images to enable rapid positioning. Concurrently, deep learning-based algorithms for PCB defect detection are set to become essential tools in the electronic manufacturing industry, providing a wider range of applications and opportunities for development.

A standard printed circuit board (PCB) generally consists of several distinct layers:

Bottom copper layer: This foundational layer serves to establish ground connections and layouts for various circuit components. It often includes a solder mask layer to protect the circuit traces and a copper layer.

Top copper layer: This uppermost layer accommodates the primary circuit components, wires, and signal paths, frequently featuring pads for connecting electronic elements.

Inner layers: In addition to the top and bottom copper layers, a PCB may contain one or more inner layers made of glass fiber and copper foil, which facilitate the transmission of signals or power.

Signal layers: These layers are responsible for transmitting signals between electronic components, including data, control signals, and clock signals.

Power layers (power planes): These layers are responsible for providing power connections, including both power and ground, and are designed to distribute these connections effectively to maintain stability across the board.

Ground layer (ground planes): This specialized layer ensures a reliable ground connection, which helps to reduce signal interference and improve overall circuit performance.

Pad layer: Located above the top and bottom copper layers, the solder mask layer safeguards the circuit traces and copper layer from short circuits, typically featuring openings for pads to facilitate the soldering of electronic components.

Silkscreen layer: This layer generally contains labels, markings for component pins, and other relevant information to assist assembly and maintenance personnel in accurately identifying and managing electronic components.

The arrangement of these layers may vary based on the specific requirements of the PCB design. This review emphasizes deep learning-based PCB defect detection, primarily utilizing machine vision applications, where defect detection is mainly focused on surface layers, including:

Top copper layer: In this layer, the deep learning model is capable of detecting pad defects, such as incorrect soldering, short circuits, open circuits, offsets, and various other issues.

Bottom copper layer: This layer also features the arrangement of pads and electronic components, with defect detection processes akin to those applied to the top copper layer.

Pad layer: This layer enables the deep learning model to identify defects in the pad opening region.

Silkscreen layer: This layer contains details regarding component and pin markings, facilitating the detection of problems such as misaligned or damaged lettering by deep learning models.

Additionally, certain PCBs may feature specialized layers, including solder masks and inner layers, which can be utilized for identifying specific defects.

This review provides a systematic examination and comprehensive synthesis of the research literature on PCB defect detection over the past decade. It primarily emphasizes methods and algorithms that leverage deep learning techniques to improve the effectiveness of PCB defect detection. The performance and implications of these algorithms in real-world applications are discussed in detail. Furthermore, a thorough explanation of the core principles of deep learning, along with a brief overview of the Transformer model, is provided to ensure that readers gain a solid understanding of the foundational concepts in this field. The paper also presents commonly used datasets and evaluation metrics for PCB defect detection. Ultimately, based on the existing literature, current algorithms are analyzed and discussed, while potential avenues for future development in this area are proposed.

Fundamentals of Deep Learning

A. Basic Knowledge

The term deep learning (DL), introduced in 1986, initially emerged within the field of machine learning and later expanded into artificial neural networks by 2000. Deep neural networks are characterized by multiple hidden layers that progressively refine data features. This architecture enables computers to independently learn higher-level abstract features, thereby facilitating a variety of tasks including classification, regression, clustering, and generation. Unlike traditional machine learning techniques, deep learning eliminates the need for manually crafted features, automating the extraction and learning of features through extensive training on large datasets. A significant advantage of deep learning is its capability to efficiently process large volumes of data, leading to impressive results when sufficient data is available. However, challenges remain, such as the requirement for large amounts of labeled data, the high cost of computational resources, and the interpretability of the models. As a result, the training process is critically important, particularly in applications like PCB defect detection. This phase involves data preprocessing, the development of a relevant network model, and the establishment of an appropriate loss function. Recent advancements in deep learning can be attributed to significant improvements in network models, loss functions, and activation functions. As technology continues to advance, deep learning is expected to explore broader developmental avenues in the future, solidifying its role as a vital element of the artificial intelligence landscape. Attention will subsequently focus on convolutional neural networks (CNNs) and the emerging Transformer architecture.

In 1994, Lecun et al. unveiled LeNet, one of the first convolutional neural networks (CNNs) designed for handwriting font recognition. This groundbreaking research significantly influenced the future development of CNNs. In 2012, Krizhevsky presented AlexNet, which built upon the groundwork established by LeNet. AlexNet incorporated the Rectified Linear Unit (ReLU) as its activation function, a choice that proved to be more effective than the Sigmoid function for deeper networks, effectively mitigating issues such as the vanishing gradient problem. At the same time, the advent of CUDA (Compute Unified Device Architecture) technology enabled the acceleration of training deep convolutional networks by harnessing the powerful parallel computing capabilities of GPUs (Graphics Processing Units) to handle extensive operations during the training process. The notable success of AlexNet in winning the 2012 ImageNet competition serves as a testament to the effectiveness of this methodology, particularly in comparison to traditional machine learning classification algorithms. The achievements of AlexNet provide clear evidence of the enhanced performance of CNNs in tackling large-scale image classification tasks. In 2014, Simonyan et al. introduced VGG, a new network architecture characterized by a deeper structure and the use of smaller convolutional kernels to reduce the number of parameters. This capability was validated by VGG's second-place finish in the 2014 ILSVRC competition. That same year, Szegedy also presented a new deep learning architecture known as GoogLeNet. While earlier networks primarily focused on increasing depth to improve training results, this approach often led to challenges such as overfitting, vanishing gradients, and gradient explosions. GoogLeNet addressed these issues by introducing the inception module, which optimizes the use of computational resources while enhancing feature extraction within the same computational load, thereby increasing its overall effectiveness.

As the number of layers in a network increases, training the model becomes increasingly complex, leading to challenges such as gradient vanishing. In 2015, He et al. introduced residual networks (ResNet), which included a specially designed residual module aimed at addressing the gradient vanishing problem associated with deeper networks in deep neural networks. This architectural innovation simplifies the training process for deeper networks. In 2016, Cai et al. proposed a convolutional neural network (CNN) that utilized cascade learning, resulting in enhanced detection outcomes. This further highlights the considerable potential of CNNs in the detection of defects on printed circuit boards (PCBs).

In conclusion, driven by the rapid progress in deep learning, CNNs have reached significant achievements in the field of computer vision, fundamentally altering the landscape of image recognition and classification. This review focuses on the foundational architectures of CNNs, including LeNet, AlexNet, VGG, GoogLeNet, and ResNet. These architectures have attracted considerable scholarly interest and have demonstrated remarkable effectiveness in both industrial and practical applications. By conducting a thorough comparison and analysis of these network architectures, a deeper understanding of how various design strategies impact CNN performance can be achieved. This insight will be crucial for guiding future research efforts in the CNN field.

Table 1 presents a comprehensive overview of the strengths (including key features and innovations) and limitations of the aforementioned CNNs. The objective of this review is to provide readers with a clear understanding of both the advantages and drawbacks of these networks.

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.	-

The convolutional neural networks (CNNs) previously mentioned have evolved from having a limited number of layers to achieving considerable depths, with some networks comprising dozens or even hundreds of layers. Concurrently, the sophistication of these networks has advanced, paralleling improvements in activation functions, the incorporation of max pooling layers, and various other methodologies. Figure 2 (a) presents the schematic representation of the AlexNet architecture. It is noteworthy that AlexNet standardizes all input images to a size of 32×32 pixels prior to applying a series of convolutional operations. The results are then processed through a fully connected layer, producing a 1×10 vector that represents character weights. Figure 2 (b) illustrates the enhanced architecture of AlexNet, which features deeper network configurations and incorporates the Max Pooling technique, drawing inspiration from LeNet. Additionally, GPU acceleration is utilized during the training phase, leading to the model's significant achievements in the ImageNet competition. Figures 2 (c) and 2 (d) depict the Inception module, a defining feature of both VGG and GoogLeNet, which were introduced in the same year. VGG adopts a block-based methodology to gradually increase the network's depth, a simple yet effective approach. In contrast, GoogLeNet focuses on expanding network depth through an increase in width, introducing the Inception module. This module has seen continuous improvements over time, resulting in notable enhancements in accuracy. However, the persistent challenge of excessive network depth during this period, which led to less than optimal results, remained unresolved. Figure 2 (e) showcases the schematic of the residual module, a pivotal innovation introduced by ResNet. This module not only captures the output of the network but also includes the sum of the original input, thereby promoting a residual architecture. This structural concept has been reiterated in subsequent network designs, ultimately becoming a cornerstone in the evolution of CNNs.

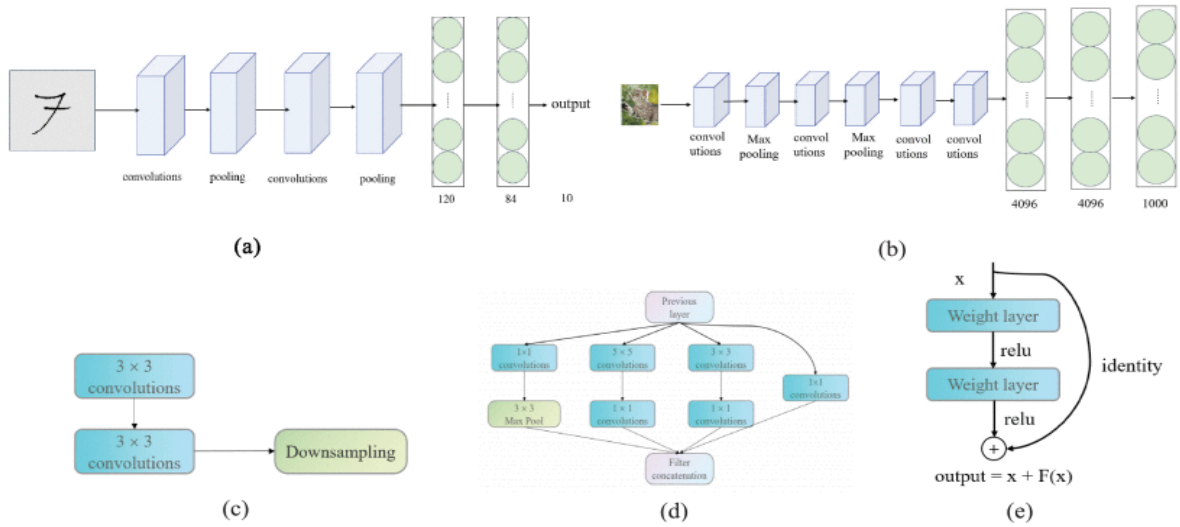


FIGURE 2. - Diagrams illustrating the network structures of classical CNNs: (a) LeNet; (b) AlexNet; (c) VGG block; (d) GoogleNet Inception module; (e) ResNet residual structure.

Nevertheless, none of the previously mentioned methods take into account the innate human inclination to focus selectively on certain information while ignoring other observable data during visual perception. The implementation of the attention mechanism offers a more efficient alternative.

C. Attention Mechanism

The Attention Mechanism concept emerged from insights gained from human visual processing. In 2014, Mnih et al. observed that even with the utilization of multiple GPUs and prior knowledge for image data processing, convolutional neural networks (CNNs) still necessitated prolonged training durations. This idea was subsequently combined with human visual observations to apply the attention mechanism in image recognition, validating its effectiveness. This represented the first incorporation of the attention mechanism into deep learning. The concept was later adapted for use in natural language processing (NLP). In 2017, Google Research unveiled the Transformer model, which is based on the attention concept and achieved outstanding results, significantly impacting the NLP landscape. Similarly, the field of computer vision (CV) saw the introduction of various innovative attention models. Hu et al. presented the SENet (Squeeze-and-Excitation Network), which integrates attention mechanisms within feature channels. SENet

autonomously determines the importance of different channels, utilizing this knowledge to enhance significant features while diminishing the influence of less relevant ones for the task at hand. The SE attention mechanism module is illustrated in Fig. 3 (a). However, SENet does not address the feature space dimension. In response, Woo et al. [45] developed the convolutional block attention module (CBAM), which combines the attention mechanism across both feature channels and feature space dimensions, thereby improving network performance without a significant increase in parameter count. A schematic representation of this module can be found in Fig. 3 (b). Nonetheless, the earlier attention mechanisms displayed increased complexity, resulting in more intricate models. Wang et al. [46] developed an efficient channel attention mechanism for deep convolutional neural networks, known as ECA-Net, which is distinguished by its lower parameter count and significant performance gains. The schematic representation of the ECA module can be found in Fig. 3 (c). In contrast to previous attention mechanisms that primarily focused on inter-channel information while neglecting spatial location details and long-range relationships, Hou and his team introduced coordinate attention (CA). This innovative approach not only captures inter-channel information but also incorporates orientation-based positional data, thereby improving the model's capability to accurately localize and identify targets. Furthermore, Coordinate Attention is characterized by its lightweight and adaptable design, allowing for easy integration into various networks while achieving considerable performance improvements. The CA attention module is illustrated in Fig. 3 (d).

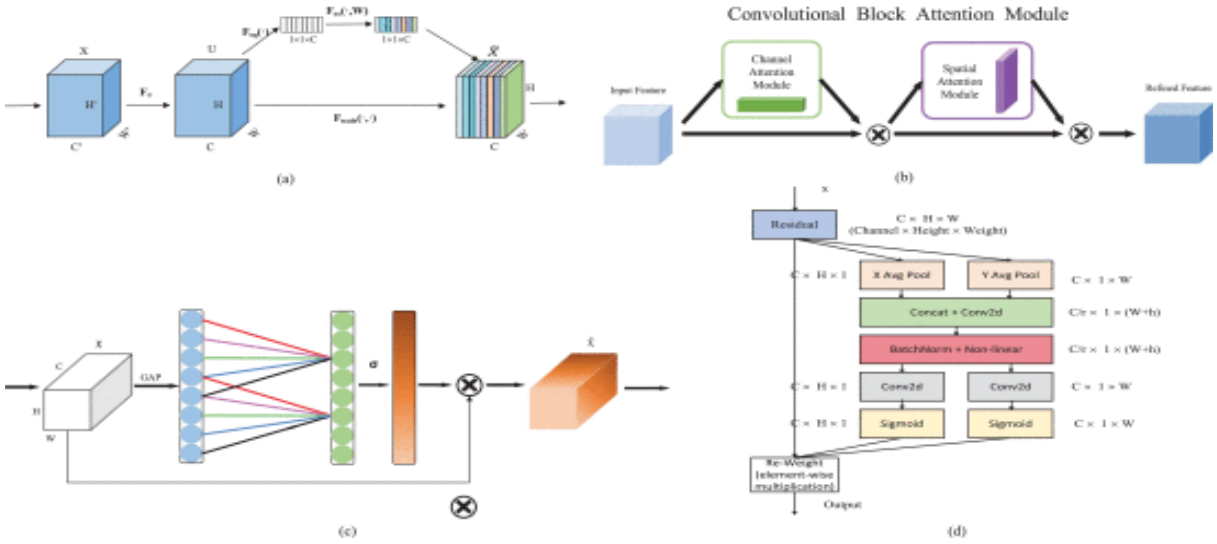


FIGURE 3. - Diagram illustrating attention mechanisms: (a) SE attention mechanism module; (b) CBAM attention mechanism module; (c) ECA attention mechanism module; (d) CA attention mechanism module.

In the context of PCB defect detection, it is observed that most regions are generally free of defects, with only a small fraction displaying imperfections. The implementation of an attention mechanism allows for increased emphasis on these defect-prone areas, thereby improving the efficiency of defect identification. As a result, the integration of an attention mechanism into the PCB defect detection model is highly beneficial. D. Transformer The Transformer architecture, developed by Google Research, marks a pivotal advancement in the field of natural language processing. Conventional sequential models, such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs), frequently face challenges like gradient vanishing and explosion when processing long sequences. In contrast, the Transformer effectively addresses these issues by utilizing a self-attention mechanism, which significantly enhances its ability to capture long-range dependencies within sequences. As a result, the Transformer has become the foundational architecture in the NLP field, with its essential structure illustrated in Fig. 4 (a).

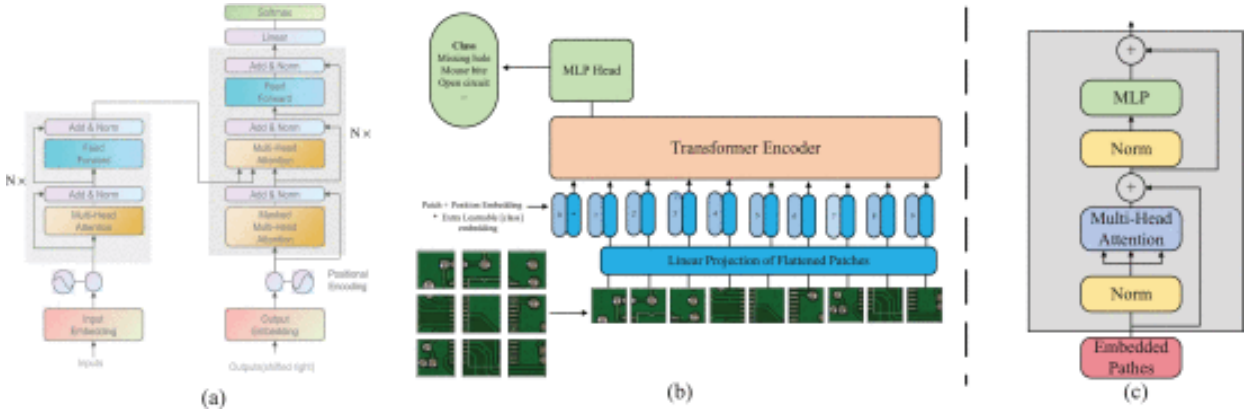


FIGURE 4. - Transformer models: (a) The transformer model introduced by Google Research for NLP applications; (b) Fundamental architecture of ViT; (c) Transformer encoder configuration within ViT.

Nevertheless, none of the previously mentioned methods take into account the innate human inclination to focus selectively on certain information while ignoring other observable data during visual perception. The implementation of the attention mechanism offers a more efficient alternative.

C. Attention Mechanism

The Attention Mechanism concept emerged from insights gained from human visual processing. In 2014, Mnih et al. observed that even with the

utilization of multiple GPUs and prior knowledge for image data processing, convolutional neural networks (CNNs) still necessitated prolonged training durations. This idea was subsequently combined with human visual observations to apply the attention mechanism in image recognition, validating its effectiveness. This represented the first incorporation of the attention mechanism into deep learning. The concept was later adapted for use in natural language processing (NLP). In 2017, Google Research unveiled the Transformer model, which is based on the attention concept and achieved outstanding results, significantly impacting the NLP landscape. Similarly, the field of computer vision (CV) saw the introduction of various innovative attention models. Hu et al. presented the SENet (Squeeze-and-Excitation Network), which integrates attention mechanisms within feature channels. SENet autonomously determines the importance of different channels, utilizing this knowledge to enhance significant features while diminishing the influence of less relevant ones for the task at hand. The SE attention mechanism module is illustrated in Fig. 3 (a). However, SENet does not address the feature space dimension. In response, Woo et al.developed the convolutional block attention module (CBAM), which combines the attention mechanism across both feature channels and feature space dimensions, thereby improving network performance without a significant increase in parameter count. A schematic representation of this module can be found in Fig. 3 (b). Nonetheless, the earlier attention mechanisms displayed increased complexity, resulting in more intricate models.

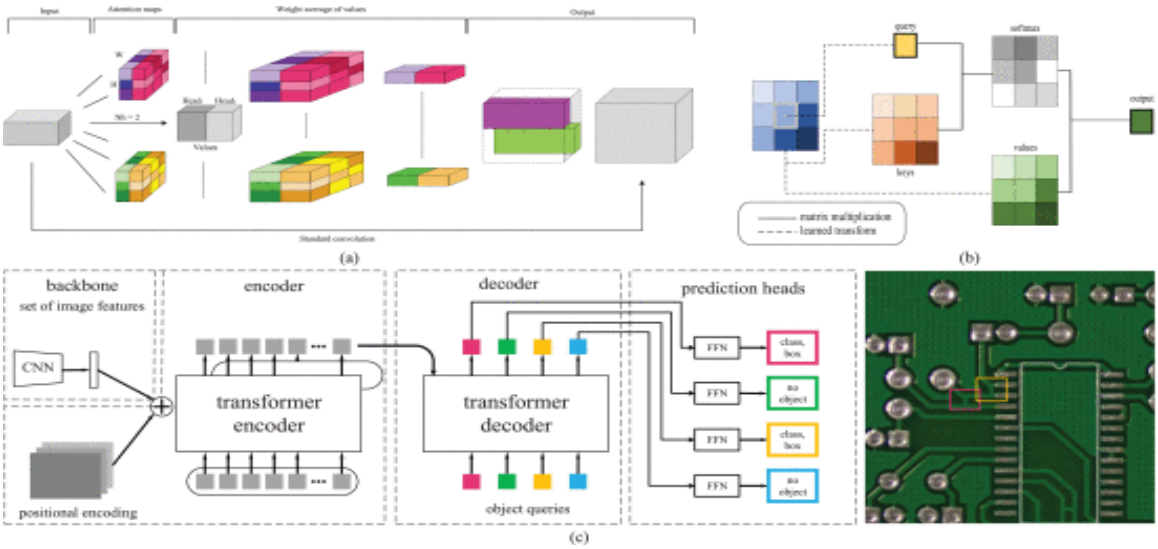


FIGURE 5. - Transformer models in CV: (a) Schematic of the proposed attention-augmented convolutional architecture; (b) Substitution of attention for convolution in ResNet with kernel size k=3; (c) Pioneering alternative to CNN for the DETR model.

In the realm of PCB defect detection, the identification of defects is often challenged by their minute size, which renders conventional methods inadequate. Recently, the Transformer model has emerged as a more effective solution due to its enhanced sensitivity to small targets and intricate defects across various environments. Consequently, the application of the Transformer model in PCB defect detection holds the promise of significantly improving performance.

E. Summary

This section explores the essential principles of deep learning, including Convolutional Neural Networks (CNNs), attention mechanisms, and the Transformer model. Although CNNs were developed in the previous century, they only began to flourish over a decade ago, driven by advancements in computational capabilities and the availability of extensive labeled datasets. This progress has led to remarkable achievements across multiple fields, establishing CNNs as vital components in PCB defect detection and machine vision. The attention mechanism serves as a flexible module that can be effectively integrated into both CNNs and Transformers. Its primary function is to enable the model to focus on significant features while ignoring irrelevant background information, thereby improving the detection of PCB defects. The Transformer model, a deep learning architecture based on the self-attention mechanism, has experienced rapid advancement despite its relatively recent introduction, thanks to extensive research by numerous scholars. It has been widely adopted in various tasks within Natural Language Processing (NLP) and Computer Vision (CV), yielding exceptional results. The potential for further advancements in the Transformer model is unmistakable.

2. PCB Defect Detection Method Utilizing Deep Learning :

In the field of PCB defect detection, detection algorithms are primarily divided into two main categories: two-stage algorithms and single-stage algorithms. In addition to the previously mentioned CNN-based approaches, a new category of PCB defect detection algorithms utilizing Transformers has emerged. Prominent examples of two-stage algorithms include Region CNN (R-CNN), Fast Region CNN (Fast R-CNN), Faster Region CNN (Faster R-CNN), and Mask Region CNN (Mask R-CNN), among others. These methodologies divide the PCB defect detection process into two distinct phases: the first phase involves region proposal (RP), which generates pre-defined boxes that may contain the objects of interest; the second phase entails sample classification using CNNs.

The R-CNN algorithm initiates the process by producing a series of candidate regions (Region Proposals) from the input image. It then extracts features from these regions using CNNs and sends the extracted features to classifiers and bounding box regressors for object detection. Although it achieves high accuracy, the algorithm's speed is hindered by the necessity of performing separate CNN feature extraction for each candidate region. The structure of this algorithm is illustrated in Fig. 6 (a). An enhanced version, Faster R-CNN, incorporates a region proposal network (RPN), which is a learnable network

aimed at expediting the generation of candidate regions. Faster R-CNN effectively combines the RPN with subsequent classifiers and bounding box regressors, creating a comprehensive end-to-end target detection network. This integration allows for the simultaneous generation of candidate regions and feature extraction within a single network, thereby significantly improving detection speed. The configuration of Faster R-CNN is depicted in Fig. 6 (b).

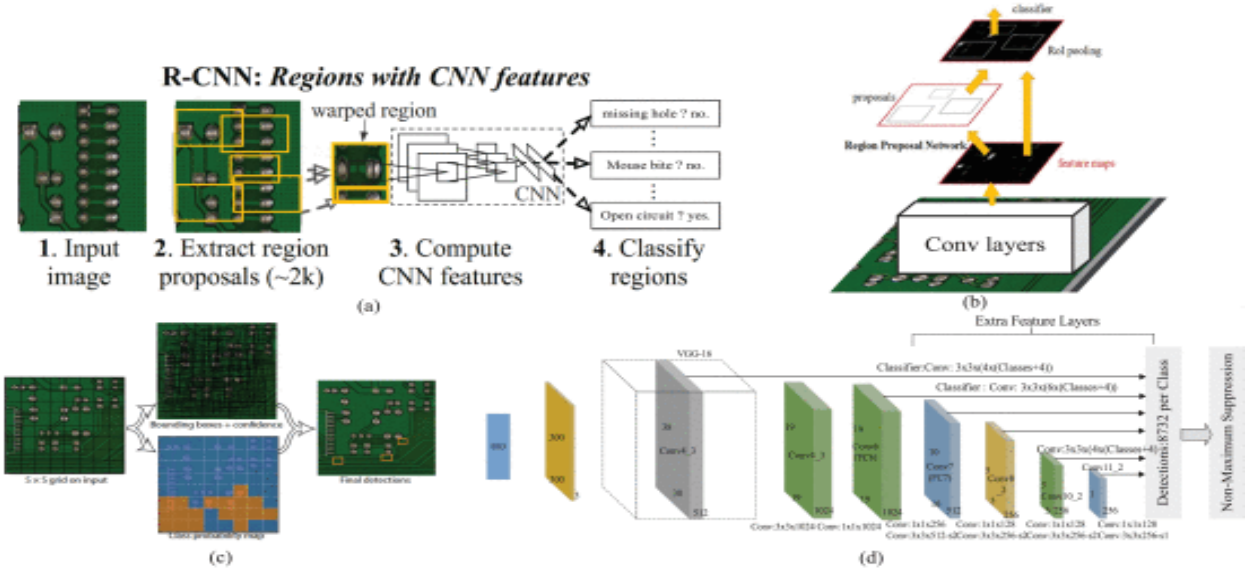


FIGURE 6.

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 category of single-stage algorithms includes notable examples such as the Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO). These approaches redefine the detection of PCB defects as a regression task. The YOLO algorithm offers a unique approach by framing target detection as a regression problem. It achieves this by dividing the input image into a grid and simultaneously predicting both class labels and bounding box parameters for multiple targets within each grid cell. YOLO's method of target detection relies on single-shot forward propagation, which contributes to its remarkable speed. However, it may be less effective in identifying smaller targets. The architecture of YOLO is illustrated in Fig. 6 (c). Similarly, the SSD algorithm is categorized as a single-stage target detection method. Like YOLO, it predicts target classes and bounding boxes concurrently on feature maps at various scales, accommodating different target sizes through multiple anchor boxes of varying dimensions. This design allows SSD to perform well in detecting small and multi-scale targets, as shown in Fig. 6 (d). Although SSD operates at a slightly slower pace compared to YOLO, it may provide improved accuracy in certain scenarios.

When comparing two-stage and single-stage algorithms, the former tends to offer greater accuracy but is more time-consuming, making it less suitable for real-time detection applications. In contrast, single-stage algorithms are faster but generally exhibit lower accuracy. Additionally, transformer-based algorithms for PCB target detection have emerged, which, unlike traditional convolutional neural networks (CNNs), leverage transformer architecture. These methods demonstrate excellent detection accuracy; however, they are characterized by slower detection speeds, higher computational resource requirements for training, and a need for substantial data support.

A. Enhanced One-Stage Algorithm

To tackle the challenges of inadequate stability and low precision in PCB defect detection models, Xin et al. [71] proposed an advanced YOLOv4 model. This model integrates a mosaic data augmentation technique during the input phase and substitutes the leaky rectified linear unit (Leaky-ReLU) activation function in the network's backbone with the Mish activation function. Furthermore, the detection images undergo automatic segmentation based on the average dimensions of labeled boxes, thereby increasing the probability of capturing the target within the anchor frame. In a similar vein, to confront the difficulties of identifying small defects against intricate backgrounds in PCBs, Zhang et al. developed a lightweight single-stage defect detection network. This network employs a dual attention mechanism alongside a path-aggregation feature pyramid network (PAFPN) to improve the detection of minor defects. The lightweight backbone neural network MobileNetV2 replaces ResNet101, significantly decreasing the number of model parameters. A dual attention mechanism is incorporated to facilitate effective feature extraction, which is further enhanced by replacing the feature pyramid network (FPN) with PAFPN in the neck of the model. This refined model not only reduces inference time and parameter count but also improves detection accuracy. Jiang et al. introduced modifications to the SSD network model, implementing coordinated attention in the shallow network to better manage positional information, especially for smaller targets. Li et al. developed a dataset for PCB assembly scene object detection, addressing issues related to anchor frame size. They conducted a comprehensive analysis of effective receptive fields (ERF) across the output layers, establishing ERF ranges and proposing ERF-based anchor frame assignment rules to mitigate anchor frame size challenges. Additionally, they designed an enhanced atrous spatial pyramid pooling (ASPP).

The training process is affected by the scarcity of labeled PCB defect samples, leading to the influence of unlabeled samples. To tackle this issue, Wan et al. proposed a defect detection method known as the data-expanding strategy (DE-SSD), which was assessed using YOLOv5 with both labeled and unlabeled samples. This method lessens the dependence on labeled data by leveraging both types of samples. Additionally, a data-expanding strategy is suggested to alleviate the effects of unlabeled samples. This improvement is particularly noticeable with smaller datasets; however, its effectiveness tends to decrease as the volume of data increases. In a separate investigation, Wu et al. introduced GSC YOLOv5, a deep learning detection technique that

combines a lightweight network with a dual-attention mechanism. This revised algorithm utilizes lightweight Ghost Conv and Ghost Bottleneck structures, leading to a significant reduction in the model's parameter count and floating-point operations. Moreover, the incorporation of SE and CBAM modules into the network enhances both accuracy and detection speed. To address issues related to detection efficiency, memory usage, and sensitivity to minor defects, Xuan et al. implemented a novel cross stage partial network darknet (CSPDarkNet) as the backbone for YOLOX. This updated backbone features multiple inverted residual blocks and integrates coordinated attention into the architecture, greatly enhancing the model's ability to identify small PCB defects. Importantly, this modified model is lighter and more appropriate for deployment on embedded systems. Zhao et al. [84] further advanced YOLOv5 by incorporating adaptively spatial feature fusion (ASFF) for feature integration, allowing for the adaptive fusion of different levels of feature information across various spaces. They also introduced a global attention mechanism (GAM) to improve the model's information extraction capabilities. Zheng and his team presented an advanced fully convolutional neural network (CNN) by incorporating successive convolutional modules into the MobileNetV2 framework. This enhancement, along with an optimized skip connection, results in improved detection speed and accuracy when compared to the VGG-16 and ResNet-50 models. Lim and his associates created an innovative multi-scale feature pyramid network utilizing YOLOv5, specifically targeting the detection of tiny PCB defects by harnessing contextual information. The network also employs the CIoU loss function to accurately assess spatial parameters, effectively pinpointing the precise locations of these defects. Yu and his collaborators developed a lightweight and efficient network designed for the identification of small PCB defects. They introduced a diagonal feature pyramid (DFP) within the backbone network, facilitating low-cost fusion of extensive feature maps, which enhances the detection of these subtle imperfections. Furthermore, they established a multi-scale necking network to address defects of varying sizes and implemented an adaptive localization loss function to improve the model's capability to identify these small-scale flaws.

In conclusion, the previously discussed research has achieved notable progress in the domain of PCB defect detection through the utilization of one-stage algorithms. This progress includes a range of innovative strategies, such as attention mechanisms, data augmentation techniques, and advanced backbone networks. These methodologies have been extensively applied to PCB defect detection from images, resulting in remarkable improvements in various aspects, including accuracy and detection speed. A comparative evaluation of these one-stage algorithms against certain alternative methods, highlighting their advantages and drawbacks, is detailed in Table 2. It is important to recognize, however, that one-stage algorithms are not without their challenges. These challenges include the potential for reduced performance when faced with complex and diverse defect scenarios, a significant dependence on limited sample data, and the need for further advancements to effectively address variations in dimensions, angles, and lighting conditions. The outcomes of PCB defect detection utilizing four distinct algorithms are illustrated in Fig. 7.

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.	-

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; (c) (d) Printed circuit board quality detection method integrating lightweight network and dual attention mechanism.

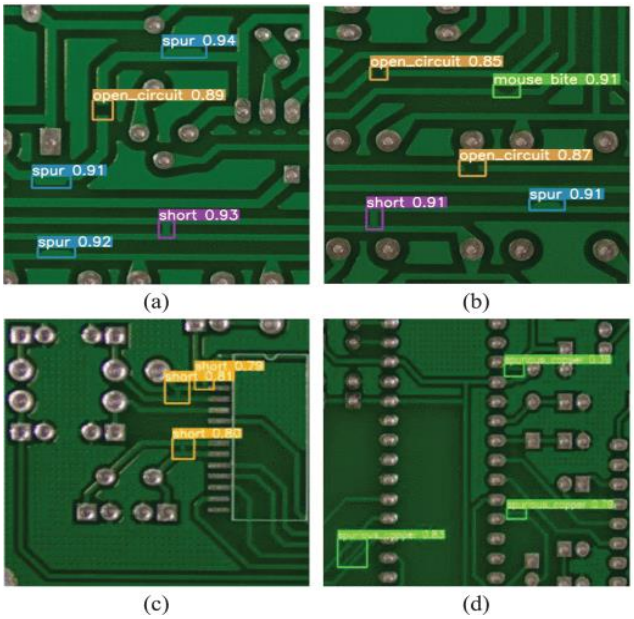


FIGURE 7.

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.

B. Based Two-Stage Algorithm

Although the single-stage algorithm provides quicker performance, the two-stage algorithm significantly surpasses it in detection accuracy. To facilitate PCB defect detection through machine vision, Li et al. proposed a faster-RCNN algorithm based on VGG16, which integrates data expansion and RGB data enhancement techniques. In response to the difficulties associated with identifying minute defects, which are particularly challenging to generate and detect in practical applications, Ding et al. [8] developed the tiny defect detection network (TDD-net). This network utilizes a K-means algorithm to create suitable anchor frames, improves inter-feature map relationships, and applies online hard example mining (OHEM) to enhance region of interest (ROI) predictions. To overcome the constraints of traditional defect detection methods that are template-dependent and computationally intensive, Hu et al. [92] introduced an algorithm based on Faster RCNN and Feature Pyramid Network (FPN). This approach initially employs ResNet50 with feature pyramids as its backbone and subsequently incorporates generative adversarial region proposal networks (GARPN) [93] to improve the accuracy of anchor frame predictions. In the same year, Li et al. presented a feature pyramid-based network that integrates an SE module into ResNet-101 to boost the network's expressive capabilities, introduces a top-down structure to enhance overall feature levels, and utilizes ROI Align instead of ROI Pooling to reduce the effects of dislocations on the detection of small object defects.

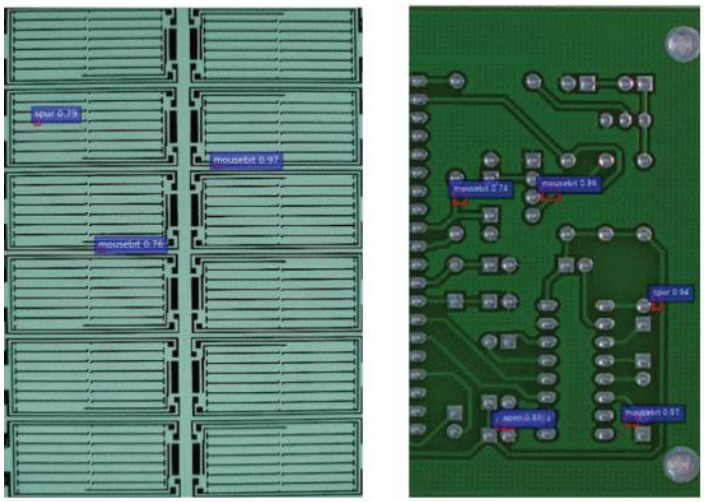
In conclusion, an examination of studies centered on two-stage algorithms reveals a scarcity of scholarly contributions in comparison to those dedicated to one-stage algorithms. Nevertheless, it is clear that two-stage algorithms provide considerable benefits regarding detection accuracy. Table 3 illustrates a comparison of two-stage algorithms, outlining the methodologies used along with their respective strengths and weaknesses. Recently, one-stage algorithms have gained prominence to satisfy the demands for real-time detection speeds. However, the disparity in detection accuracy when juxtaposed with two-stage algorithms is minimal. Consequently, there is a limited number of researchers concentrating on this area. Despite this, the application of two-stage algorithms remains a prudent option for particular tasks that require high detection accuracy and customized datasets, even when rapid detection speed is not a priority. Figure 8 displays the outcomes of PCB defect detection utilizing four distinct algorithms.

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.



C. Transformer-Based Algorithm

The Transformer architecture has shown considerable effectiveness in both computer vision (CV) and natural language processing (NLP). However, it faces certain challenges when applied to visual inspection tasks, particularly those that involve strict time limitations and specific equipment requirements. These challenges have contributed to a limited body of research focused on the use of Transformers for detecting defects in printed circuit boards (PCBs). Despite these obstacles, various studies have begun to explore solutions aimed at harnessing the full capabilities of Transformers in PCB defect detection. By overcoming the inherent limitations of the Transformer and applying it effectively in PCB inspection tasks, it is expected to provide a more efficient and precise approach for industrial manufacturing.

An et al. proposed a label robust and patch correlation enhanced Vision Transformer (LPViT). Their research introduces a novel ViT model based on LPViT principles, emphasizing robustness while effectively utilizing various regions of the PCB image in relation to one another. To improve the mutual understanding among different image areas, certain blocks are randomly masked or replaced. The model is ultimately trained using a label smoothing technique, which enhances its robustness. In a separate study, Chen utilized an advanced clustering algorithm to create suitable anchor frames specifically designed for the PCB defective dataset. This study transitioned from using convolutional neural networks (CNNs) to employing a shifted window transformer (Swin-Transformer) for feature extraction. Furthermore, the order of channels in the feature map was modified to allow the network to prioritize more significant information effectively. Additionally, both convolutional and attention mechanisms were incorporated to improve the network's feature extraction capabilities. Yang et al. presented an enhanced version of the YOLOv7 model.

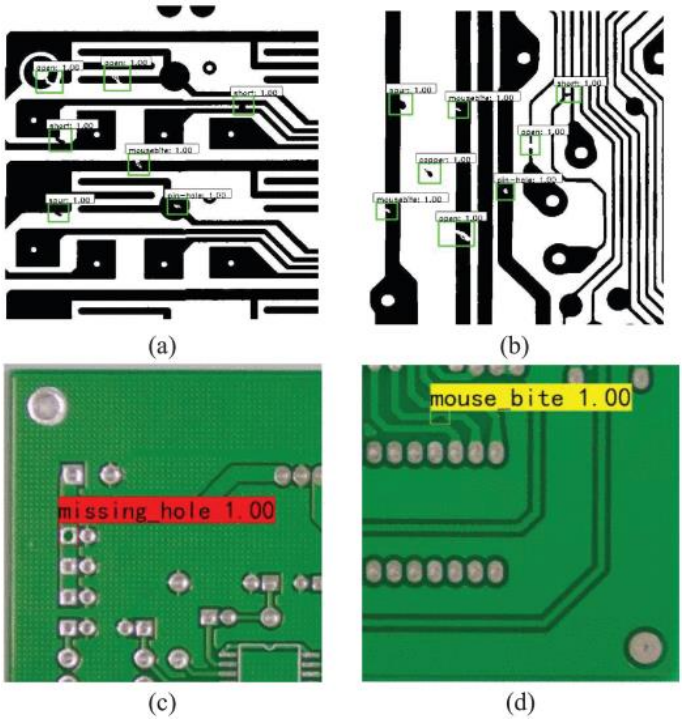


FIGURE 9. - Results of PCB defect detection using four different algorithms: (a) (b) LPViT; (c) (d) Transformer-YOLO.

Transformers have been rarely utilized in the detection of defects on printed circuit boards (PCBs); however, advancements in computer vision and computational power have led to the development of various transformer models. In 2021, Liu et al. trained a Swin Transformer V2 model, which comprises three billion parameters, and introduced techniques such as post-normalization and scaled cosine attention. This methodology achieved state-

of-the-art (SOTA) results across a range of visual tasks. As of 2023, the Swin Transformer V2 backbone network remains a focal point of extensive research, showcasing remarkable performance and further propelling the advancement of large visual models.

Current Vision Transformers (ViTs) have not effectively utilized features across different scales, which are essential for visual data. To tackle this issue, Wang et al. introduced the Crossformer in 2021, which features a cross-scale embedding layer (CEL) and long-short distance attention (LSDA). In 2023, they improved the Crossformer by implementing a progressive group size (PGS) and an amplitude cooling layer (ACL) to address challenges related to the expansion of self-attention maps and amplitude surges. Due to the absence of prior image information, ViTs tend to underperform in dense prediction tasks. To counter this, Chen et al. proposed the ViT-adapter in 2022, an auxiliary network that does not require pre-training, allowing the fundamental ViT model to adjust to downstream dense prediction tasks without any changes to its architecture. This significantly enhanced the model's performance in such tasks.

These innovative transformer models, tailored for large visual models, effectively leverage features at various scales and excel in dense prediction tasks, which are also critical in transformer-based PCB defect detection. We assert that these advanced transformer models and their methodologies can be effectively applied to PCB defect detection tasks, thereby improving the performance of existing models.

D. Summary

This section provides an overview of one-stage, two-stage, and transformer-based algorithms. One-stage algorithms are noted for their high detection accuracy and speed, making them suitable for enterprises requiring real-time detection of PCB defects. However, their effectiveness may be compromised in more complex defect situations. Conversely, the two-stage algorithm, while slower, is particularly adept at identifying intricate defects due to its enhanced accuracy, which has led to its prevalent use in factories for PCB defect detection. The transformer-based algorithm, which differs structurally from the other types, has also shown promising results in detecting PCB defects and performing various industrial tasks. Despite having a large number of parameters, it meets the demands of factories for PCB defect detection. Additionally, the transformer model has demonstrated strong performance in other fields, suggesting significant potential for future advancements.

3. Evaluation Metrics, PCB Defect Datasets, and Comparative Findings :

A. Evaluation Metrics

In tasks related to PCB defect detection, it is essential to evaluate the accuracy of algorithmic localization, which quantifies the difference between the bounding box predicted by the algorithm and the actual bounding box of the target. A commonly utilized metric for assessing positional accuracy is the intersection over union (IoU). This metric is defined as the ratio of the area of intersection between the predicted frame and the actual frame to the area of their union. Specifically, as illustrated in Fig. 10 below, the intersection refers to the area where the two frames overlap, while the union represents the total area covered by both frames. The degree of overlap between the predicted outcomes and the actual annotations is calculated using the intersection and union ratios, which aids in evaluating the algorithm's positional accuracy as shown in formula (1). In practical applications, the threshold for the intersection ratio is typically set at 0.5. If the intersection ratio between the predicted frame and the actual labeled frame exceeds this threshold, the algorithm is considered to have accurately identified the target's location. Adjusting the intersection ratio threshold allows for modifications in the algorithm's positional accuracy and false negative rate. It is important to note that different tasks may require the establishment of distinct thresholds.

$$IoU=A\cap BA\cup B(1)$$

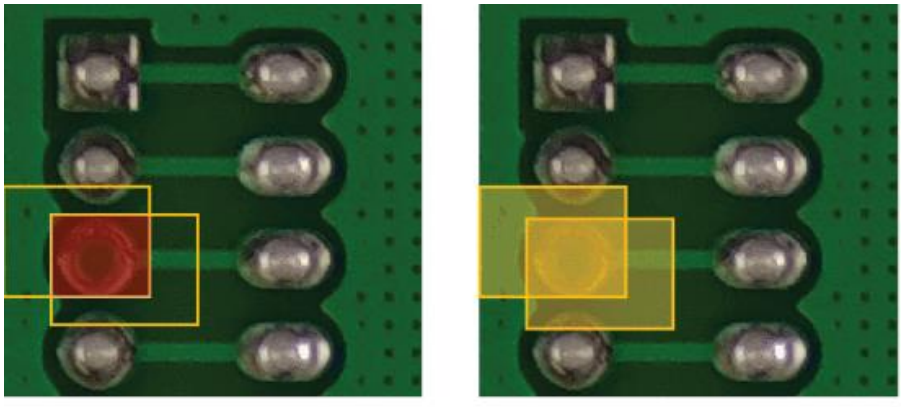


FIGURE 10. - IoU schematic diagram: (a) Red represents the intersection region [8]; (b) Yellow represents union region [8].

Equation (1) indicates that a higher Intersection over Union (IoU) value reflects a closer alignment between the predicted object region and the actual region, thereby enhancing the accuracy of detection outcomes. Specifically, an IoU value of 1 denotes complete overlap between the actual object area and the predicted area, while a value of 0 signifies no overlap at all. Nevertheless, there are situations where IoU may not accurately represent positional precision. For example, in the three images depicted in Fig. 11, despite the calculated IoUs for these images (Fig. 11 (c)) being identical, it is clear that the third image demonstrates superior quality. As a result, various improved iterations of IoU have been developed, including GIoU, DIoU, CIoU, EIoU, α IoU, and SIoU. GIoU addresses the challenge of gradient backpropagation for IoU's two frames when there is no intersection. On the other hand, DIoU incorporates both the predicted and actual frames, enhancing GIoU by considering the centroid distance and the distance between the minimum enclosing frames, while also factoring in aspect ratio relationships. However, it does not resolve the issue of actual distance. To address this, EIoU substitutes DIoU's aspect ratio with the actual differences in width and height, along with their respective confidence levels. In 2021, α IoU was introduced, which

utilizes a single parameter α, resulting in improved performance compared to other IoUs. The presence of multiple angles at equidistant points affects the actual loss, prompting the introduction of SIoU in 2022 to mitigate the impact of angles. Currently, a modified version of IoU is exclusively used in the loss function, while IoU continues to serve as the benchmark for evaluation metrics.

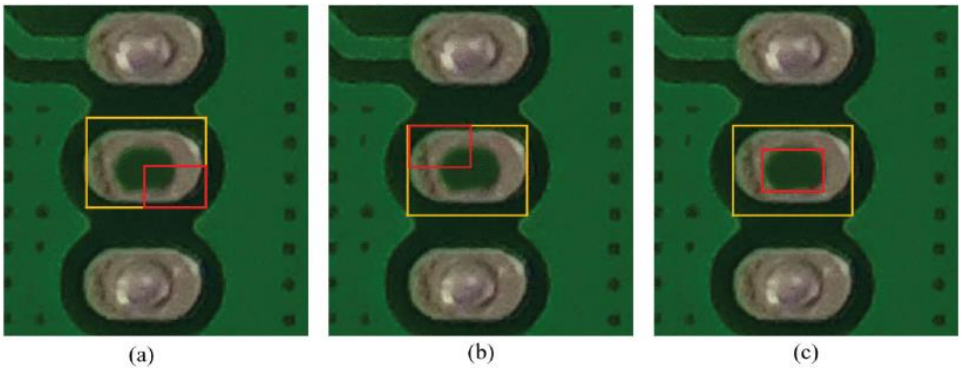


FIGURE 11. - IoU limitations (while the IoU values are equal for the three images, it's noticeable that image (a) [8] and image (b) [8] exhibit lower accuracy. Image(c) demonstrates superior accuracy [8].)

The categorization of detection outcomes in target detection tasks is divided into four distinct categories: predicted values that correspond with positive examples are classified as P (Positive), those that correspond with negative examples are classified as N (Negative), values that match true values are designated as T (True), and values that contradict true values are labeled as F (False). Once classified, the data can be structured into a confusion matrix, resulting in four unique combination types.

As illustrated in Fig. 12, TP indicates the number of accurately detected targets, which includes instances where the predicted positive sample corresponds with the true positive sample. TN represents the number of correctly identified background instances, where the predicted negative sample aligns with the true negative sample. FP refers to the number of incorrectly detected targets, characterized by instances where the predicted positive sample does not match the true negative sample. Lastly, FN accounts for the number of targets that were missed, indicating cases where the predicted negative sample does not align with the true positive sample.

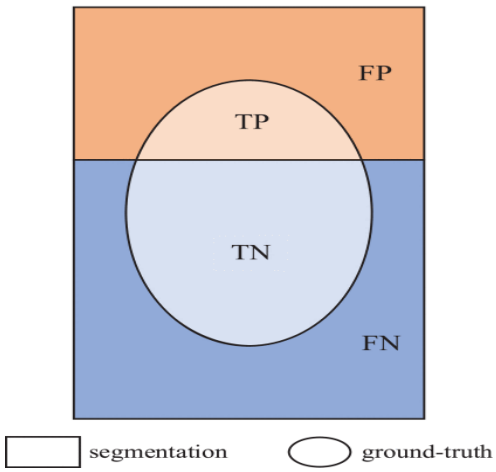


FIGURE 12. - Schematic diagram of the recognition result.

Assessing the performance of algorithms can be enhanced through a thorough enumeration and comparison of predictions across various categories. Notable evaluation metrics include Accuracy (Acc), Precision, Recall, F1 Score, Average Precision, mean Average Precision (mAP), mAPSmall, mAPmedium, mAPlarge, and Frames Per Second (FPS).

1) Accuracy

The accuracy metric indicates the ratio of correctly classified instances to the total number of instances, thereby reflecting the model's effectiveness in categorizing input data. The corresponding formula is represented in Equation (2).

$$Acc = (TP + TN) / (TP + FN + FP + TN)$$

2) Precision

Precision measures the proportion of true positive predictions relative to the total number of positive predictions made by the model. The formula is represented in Equation (3).

$$Precision = TP / (TP + FP)$$

3) Recall

Recall, often referred to as the true positive rate, quantifies the ratio of correctly identified positive instances to the actual number of positive instances. The formula is illustrated in Equation (4).

Recall = TP / (TP + FN)

4) F1 Score

The F1 score serves as a harmonic mean of precision and recall, effectively balancing the model's accuracy with its ability to identify positive instances. The formula is represented in Equation (5).

F1 = 2TP / (2TP + FP + FN)

5) Average Precision

Among the commonly utilized evaluation metrics, Precision and Recall are vital for assessing model performance. However, it is important to note the inverse relationship between these two metrics: an increase in Precision typically leads to a decrease in Recall, and vice versa. To mitigate this issue, the Average Precision (AP) metric is introduced to provide a more holistic evaluation of the model's performance.

The term "average precision" refers to the mean value derived from the integration of accuracy rates across various thresholds, covering a recall spectrum from 0 to 1. For each category, a precision-recall curve is generated, which allows for the calculation of the area under this curve, known as the AP value. Therefore, the average precision of a model is defined as the mean of the AP values across all categories, resulting in the mean average precision (mAP). This metric is widely recognized as one of the most significant performance indicators in target detection, measuring a model's effectiveness in recognizing multiple target categories.

The methodologies for calculating mAP metrics differ. Specifically, AP0.5 indicates the average accuracy when the intersection-over-union (IoU) threshold is greater than 50%, while AP0.5:0.95 reflects the average accuracy when the IoU threshold varies from 50% to 95% in increments of 5%. In practical target detection scenarios, models are often required to identify targets from various categories, which necessitates the computation of AP values for each category and their subsequent averaging to obtain the mAP metric.

In summary, the mean average precision (mAP) is an essential metric for evaluating the performance of target detection models. By integrating model accuracy and recall, mAP provides valuable insights into a model's capability to detect a wide array of target categories. Consequently, this analysis utilizes mAP0.5:0.95 and mAP@0.5 (AP0.5) as the primary metrics for evaluation. The formula is illustrated in Equation (6).

$AP=\int 10P(R)dr, mAP=1N\sum(i=1)NAPi(6)$

6) mAPSmall, mAPmedium, and mAPlarge

These three metrics represent the average accuracies for objects of varying sizes. Specifically, mAPSmall is associated with objects that have an area smaller than 32×32 pixels, mAPmedium applies to objects with an area between 32×32 pixels and 96×96 pixels, and mAPlarge pertains to objects exceeding 96×96 pixels in area. These metrics offer a comprehensive understanding of the algorithm's detection accuracy across small, medium, and large objects.

7) FPS

FPS is a metric used to evaluate inference speed, indicating the number of images that can be processed per second on particular hardware. It is an important measure for assessing model performance and its suitability for real-time applications. By calculating the model's ability to process images within a second, one can evaluate its real-time performance. Higher FPS values indicate the model's capability for faster image processing, thus improving real-time inference efficiency. This characteristic is vital for various applications that require quick response times, such as real-time video analysis, autonomous driving systems, and real-time object recognition. As a result, researchers and developers aim to improve the FPS value of their models.

B. PCB Defect Datasets

In the existing literature on PCB defect detection, numerous studies rely on proprietary datasets. Consequently, this subsection aims to present several publicly accessible datasets, which offer distinct advantages over proprietary alternatives. Public datasets not only enhance credibility but also establish more precise baselines. Furthermore, they promote reproducibility of experiments among researchers. These publicly available datasets provide researchers with dependable standards and benchmarks, enabling effective performance comparisons and advancements in methodologies. Additionally, they create a wider platform for innovation and collaboration within the research community. The currently available datasets include PCB Defect, PCB Defect-Augmented [8], DEEP PCB [115], HRIPCB [116], and Micro-PCB, among others. These datasets feature diverse characteristics, including various defect types, quantities of images, and environmental conditions. Models trained on different datasets demonstrate varying levels of accuracy and are tailored for specific scenarios. Comprehensive details regarding the publicly accessible PCB defect detection datasets can be found in Table 5.

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.

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]

C. Summary

This section delineates the evaluation metrics relevant to PCB defect detection, examines publicly available datasets, and contrasts these metrics with the algorithms presented in section II. Evaluation metrics enable an analysis of a model's advantages and limitations, where a higher frames per second (FPS) indicates quicker model inference, an increased mean Average Precision (mAP) and mAP@0.5 denote enhanced accuracy, and a lower parameter count signifies a more efficient model. In the context of PCB defect detection datasets, our focus is on several public collections that vary in size from over a thousand to more than 10,000 images. These publicly available datasets establish a benchmark, facilitate performance comparisons among models, and contribute to advancements in PCB defect detection through their superior quality. In our comparative analysis, we detail the backbone, FPS, mAP, mAP@0.5, parameters, recall, and the dataset utilized in each study for every model. By leveraging this information, we generate a series of statistical charts that offer a clear depiction of each model's strengths and weaknesses. However, it is important to note that the use of different datasets means these comparative results do not provide a definitive judgment on the models' performance

4. 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.

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.

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, pp. 2766, Mar. 2023.

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

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.

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.
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.
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.
9. X. Wu, Y. Ge, Q. Zhang and D. Zhang, "PCB defect detection using deep learning methods", *Proc. IEEE 24th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, pp. 873-876, May 2021.
10. F. Raihan and W. Ce, "PCB defect detection USING OPENCV with image subtraction method", *Proc. Int. Conf. Inf. Manage. Technol. (ICIMTech)*, pp. 204-209, Nov. 2017.
11. A. Raj and A. Sajeena, "Defects detection in PCB using image processing for industrial applications", *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, pp. 1077-1079, Apr. 2018.
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.
13. Y. Tang, M. Chen, C. Wang, L. Luo, J. Li, G. Lian, et al., "Recognition and localization methods for vision-based fruit picking robots: A review", *Frontiers Plant Sci.*, vol. 11, pp. 510, May 2020.
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.
15. W. Ming, F. Shen, X. Li, Z. Zhang, J. Du, Z. Chen, et al., "A comprehensive review of defect detection in 3C glass components", *Measurement*, vol. 158, Jul. 2020.
16. W. Ming, F. Shen, H. Zhang, X. Li, J. Ma, J. Du, et al., "Defect detection of LGP based on combined classifier with dynamic weights", *Measurement*, vol. 143, pp. 211-225, Sep. 2019.
17. W. Ming, C. Cao, G. Zhang, H. Zhang, F. Zhang, Z. Jiang, et al., "Review: Application of convolutional neural network in defect detection of 3C products", *IEEE Access*, vol. 9, pp. 135657-135674, 2021.
18. W. He, Z. Shi, Y. Liu, T. Liu, J. Du, J. Ma, et al., "Feature fusion classifier with dynamic weights for abnormality detection of amniotic fluid cell chromosome", *IEEE Access*, vol. 11, pp. 31755-31766, 2023.
19. W. He, Y. Han, W. Ming, J. Du, Y. Liu, Y. Yang, et al., "Progress of machine vision in the detection of cancer cells in histopathology", *IEEE Access*, vol. 10, pp. 46753-46771, 2022.
20. A. Kamilaris and F. X. Prenafeta-Boldu, "Deep learning in agriculture: A survey", *Comput. Electron. Agricult.*, vol. 147, pp. 70-90, Apr. 2018.
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.
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, pp. 1058, Mar. 2019.
23. V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, K. Kavita, M. F. Ijaz, et al., "A survey of deep convolutional neural networks applied for prediction of plant leaf diseases", *Sensors*, vol. 21, no. 14, pp. 4749, Jul. 2021.
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.
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.
26. H. Tian, X. Fang, Y. Lan, C. Ma, H. Huang, X. Lu, et al., "Extraction of citrus trees from UAV remote sensing imagery using YOLOv5s and coordinate transformation", *Remote Sens.*, vol. 14, no. 17, pp. 4208, Aug. 2022.
27. W. Ming, P. Sun, Z. Zhang, W. Qiu, J. Du, X. Li, et al., "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.
28. W. He, Z. Li, T. Liu, Z. Liu, X. Guo, J. Du, et al., "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.
29. J. Schmidhuber, "Deep learning in neural networks: An overview", *Neural Netw.*, vol. 61, pp. 85-117, Jan. 2015.
30. Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436-444, May 2015.
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.
32. A. Radford, L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", *arXiv:1511.06434*, 2015.
33. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks" in Adv. Neural Inf. Process. Syst, pp. 1-9, 2012, [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", *Proc. IEEE IWAENC*, vol. 15, pp. 315-323, Jun. 2011.
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.
36. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv:1409.1556*, 2014.
37. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions", *arXiv:1409.4842*, 2014.
38. K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", *arXiv:1512.03385*, 2015.

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.
40. V. Mnih, N. Heess, A. Graves and K. Kavukcuoglu, "Recurrent models of visual attention", *arXiv:1406.6247*, 2014.
41. D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate", *arXiv:1409.0473*, 2014.