Multidisciplinary Reviews

REVIEW ARTICLE
Published Online: September 14, 2024
https://doi.org/10.31893/multirev.2025011

A deep learning approach for automated PCB defect detection: A comprehensive review

Tay Sui Jian[a] | Muhammad Hafidz Fazli Md Fauadi[a] ✉ | Saifudin Hafiz Yahaya[a] | Ahamad Zaki Mohamed Noor[b] | Adi Saptari[c]

[a]Fakulti Teknologi dan Kejuruteraan Industri dan Pembuatan, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Malaysia.
[b]Manufacturing Section, Universiti Kuala Lumpur, Malaysian Spanish Institute Kulim Hi-Tech Park, 09000 Kulim, Kedah, Malaysia.
[c]Department of Industrial Engineering, President University, Kota Jababeka, Cikarang Baru, Bekasi 17550, Indonesia.

Abstract The ever-expanding market for electronic devices has significantly heightened the demand for high-quality printed circuit boards (PCBs). Even minor defects in PCBs can pose substantial safety risks for end-users. This article provides a comprehensive review on deep learning-based approaches for PCB defect detection. Our exploration covers various critical aspects, including the classification of PCB defects, automated vision inspection (AVI) techniques, object detection methodologies, and the widespread adoption of deep learning models. Specifically, we focus on the state-of-the-art approach known as region-based Fully Convolutional Network with Feature Pyramid Networks (FPN-RFCN). Additionally, we discuss effective data augmentation techniques and commonly used evaluation metrics in this domain. This review provides valuable insights for researchers, practitioners, and industry professionals engaged in PCB quality assurance.

Keywords: deep learning, feature pyramid network, PCB, defect detection, image processing, smart manufacturing

1. Introduction

The widespread adoption of consumer electronics, including smartphones, laptops, smart televisions, and tablets, has driven the demand for high-quality printed circuit boards (PCBs) produced in large quantities. However, maintaining high PCB quality at such scales presents substantial challenges (Chung et al., 2023; Fung et al., 2024). As fundamental components in electronic devices, PCBs must exhibit robust stability, high resistance to interference, and advanced features such as enhanced integration, compactness, and high-speed transmission capabilities (Chen et al., 2023).

Effective PCB quality control is crucial. Even minor defects can lead to significant issues, including reduced manufacturing quality due to undetected flaws, increased production costs from reworking or scrapping faulty products, delays in time to market while allocating extra time for quality assurance, and potential safety hazards such as short circuits, component failure, or overheating that pose risks to users (Fung et al., 2024; Łyczek & Skarka, 2024; Reichenstein et al., 2022; Rosli et al., 2018).

The advent of Industry 4.0 (IR 4.0) and the rapid growth of AI have prompted the integration of deep learning techniques into automated optical inspection (AOI) systems across various industries. These advancements have significantly improved the accuracy and speed of detecting various PCB defects (Tai et al., 2021). However, achieving zero-defect production remains a formidable challenge for modern electronics manufacturers (Saadallah et al., 2022; Rosli et al., 2016).

This study focuses on identifying the most critical defects in PCBs. Additionally, this article reviews the deep learning approaches used for defect detection in printed circuit boards (PCBs). Consequently, a generic deep learning-based approach for detecting critical defects is proposed.

2. Types of PCB Deficits

Any anomalies or irregularities in the PCB structure or the assembled components are considered PCB defects. These imperfections can cause electronic devices to malfunction, shorten their lifespan, or reduce their reliability. Schubeck et al. (2021) classified PCB defects into four types: trace, component, solder, and via and pad. Each type of defect is further classified into subclasses, with each subclass consisting of several related defects. In addition, a study by Sankar et al. (2022) examined various defects found in PCBs, collecting data from the PCB fabrication industry from April 2017 until July 2020.

Currently, PCB defects can be classified as bare PCB defects or component-mounted PCB defects, which include through-hole defects and surface mount device (SMD) defects. A bare PCB is a circuit board without any electronic components mounted on its surface, whereas a through-hole PCB is full of holes where components are inserted and soldered. The SMD refers to components mounted on the surface of a PCB. In their papers, both Schubeck et al. (2021) and Sankar et al. (2022) focused only on component-mounted PCB defects. On the basis of an exhaustive review, the types of PCB defects classified are comprehensively illustrated in Figure 1.

 Received: July 8, 2024 | Accepted: August 11, 2024

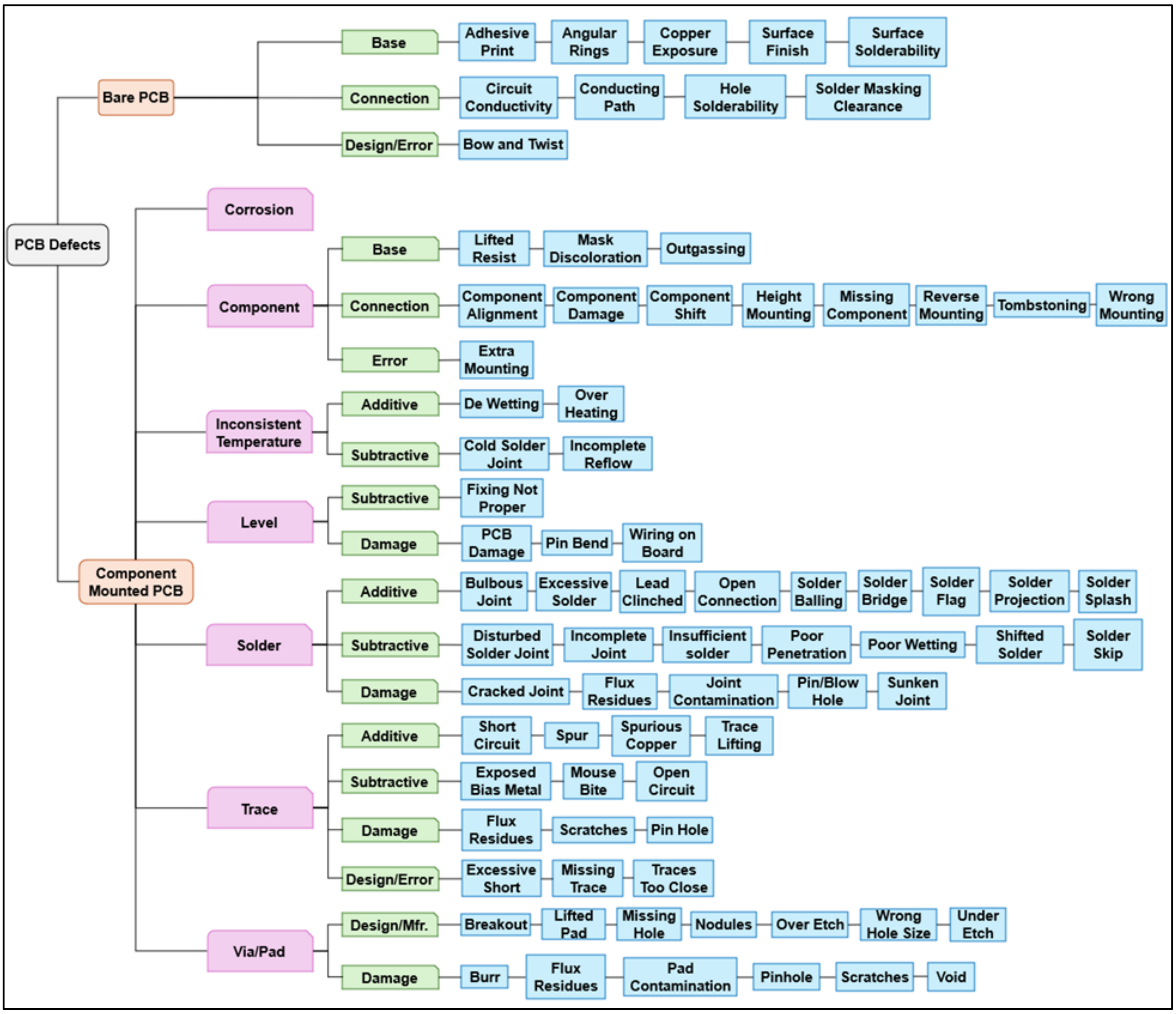


Figure 1 Taxonomy for PCB defect classification.

3. Defect Inspection Via the Vision System

3.1. Automated vision inspection

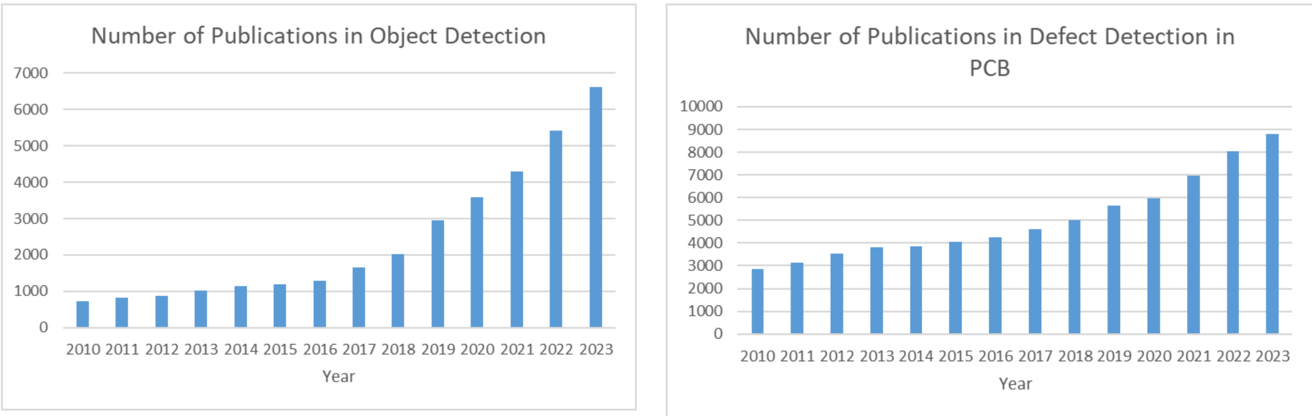
Automated vision inspection (AVI), also known as automatic optical inspection, represents a significant advancement over traditional human vision inspection methods (Silva et al., 2019). Techniques that are utilized for AVI vary depending on the specific requirements and characteristics of the inspection task. These include:

- Object detection: Identifying all objects in an image and pinpointing their locations and scales through bounding boxes (Ge et al., 2020).
- Image classification: The core of image classification research always revolves around feature extraction, the process of categorizing entire images into predefined classes or categories on the basis of their visual characteristics. This process proves valuable in scenarios where the goal is to determine whether an entire image meets certain criteria or belongs to a particular class (Chen et al., 2021).
- Image segmentation: A fundamental and pivotal step. This technique divides an image into distinct regions on the basis of specific characteristics and extracts the target of interest (Wang et al., 2020).
- Feature extraction: This method simplifies data complexity by creating a straightforward representation for each variable within a feature space, which is often achieved through linear combinations of the original input variables. Feature extraction typically involves transforming the original data into features with enhanced pattern recognition capabilities, in contrast to the original data, which may possess weaker recognition capabilities as features (Arif et al., 2022).

- Template matching: Scans the image across a dense array of subwindows and compares each one with a template representing the object (Mercier et al., 2021). This method proves valuable for detecting objects or patterns that closely resemble the template.
- Optical character recognition (OCR): Additionally, text detection and recognition involve extracting information from input images and translating it into digital text. This process involves transmitting information from the image domain to the text domain (Wang et al., 2021).
- Anomaly detection: Additionally, outlier detection or novel detection involves identifying data instances that deviate significantly from the majority of data points (Pang et al., 2021).
- Motion analysis: Relies on data from motion capture systems. There are three types of motion capture systems: marker-based systems, vision-based systems, and volumetric capture systems. Marker-based systems can generate highly accurate motion tracking data, but attaching markers to human bodies is inconvenient and time-consuming. Vision-based motion capture systems use computer vision to identify motion from images or videos. Volumetric motion capture systems go a step further than vision-based systems by generating textured and animated 3D character models instead of skeleton models (Zhu et al., 2022).
- 3D vision involves capturing and analysing three-dimensional information from images or depth sensors. These techniques are beneficial for tasks such as measuring object dimensions, detecting surface defects, or performing volumetric inspections (Linton et al., 2023).

3.2. Object detection technique

A core task in computer vision involves simultaneously classifying and pinpointing potential objects within an image (Li et al., 2023). The utilization of object detection techniques by the public has shown a consistent upwards trend from 1998--2021, as illustrated in Figure 2 (Zou et al., 2023).



(a) (b)

Figure 2 Number of publications in (a) object detection from 2010--2023. (Data from Google scholar advanced search: allintitle: "object detection"); and (b) defect detection in PCB from 2010--2023. (Data from Google scholar advanced search: allintitle: "defect detection PCB").

Several methods have been employed for object detection:

- Traditional computer vision approach: Two fundamental techniques within this domain are edge detection and the Hough transform. Edge detection plays a critical role in image processing by identifying discontinuities in intensity, which often correspond to object boundaries. In edge detection, the input image is passed through a filter to detect lines or boundaries of shapes and then extract and highlight the edge lines within the image (Dong et al., 2022). The Hough transform (HT) is a representative technique for line detection in digital images. Owing to its simplicity and efficiency, the HT can be extended to identify other regular shapes, such as circles and rectangles. The main principle of the HT involves voting for evidence from the image domain to the parameter domain, followed by the detection of shapes in the parametric domain by identifying local-maximal responses. This technique is sensitive to lighting variations and occlusions, resulting in noisy outputs containing irrelevant lines (Zhao et al., 2022).
- Deep learning-based approaches: There are two powerful deep learning approaches for detecting objects in images: convolutional neural networks (CNNs) and transformer models, as illustrated in Figure 3 (Sun et al., 2024).
- Traditional machine learning approaches: Support vector machines (SVMs), decision trees, random forests, and gradient boosting machines (GBMs) are several machine learning algorithms that can be used for object detection. SVM is a powerful machine learning technique used for classification tasks. It is designed to identify the optimal hyperplane within the feature space that maximizes the separation between two sets of training samples (Mohan et al., 2020). A random forest classifier consists of numerous trees, each grown via a form of randomization. The terminal nodes, which

are the leaf nodes, are assigned labels representing estimations of the posterior probability distribution across the image classes. Every internal node within the decision tree structure incorporates a test function. This test function serves to optimally partition the data space on the basis of the feature characteristics of the data points. To classify an image, it performs classification for each tree, and the resulting leaf distributions are aggregated (Boateng et al., 2020). The gradient boosting machine (GBM) constitutes a powerful technique in machine learning. It works by combining multiple weaker learners to create a powerful ensemble with superior predictive ability. GBMs are successful in various prediction tasks, including spam filtering, online advertising, fraud detection, anomaly detection, and computational physics (Lu & Mazumder, 2020).

- Feature-based approach: A traditional approach in computer vision that relies on extracting informative features from images to identify and locate objects. Two common feature-based object detection techniques include the histogram of oriented gradients (HOG) and the Haar cascade. The core principle of the HOG algorithm lies in its computation of gradients as local descriptors within an image. These gradients are subsequently normalized locally, resulting in features that are invariant to the object location and robust to illumination changes within the image (Ghaffari et al., 2020). The core principle of Haar cascade revolves around the utilization of Haar-like features, which are simple contrast-based descriptors employed for object identification within an image. These features are subsequently leveraged to train a cascaded structure of classifiers. The Haar cascade approach involves training a cascade of classifiers, which are progressively more complex. This cascaded structure allows for the efficient discarding of nonrelevant image regions during the detection process. While this approach offers significant advantages in terms of computational efficiency, it can be limited in its ability to handle variations in object appearance, such as orientation, scale, and lighting conditions (Georgiev et al., 2023).
- Hybrid approach: This approach involves combining multiple techniques to leverage their respective strengths. These approaches are characterized by the strategic integration of diverse methodologies, encompassing both established CV techniques and deep learning methods. This combination offers a distinct advantage in capitalizing on the heterogeneous computing capabilities available on edge devices. Heterogeneous computing refers to the ability to utilize a variety of processing units, such as CPUs and GPUs, within a single device (O'Mahony et al., 2020).

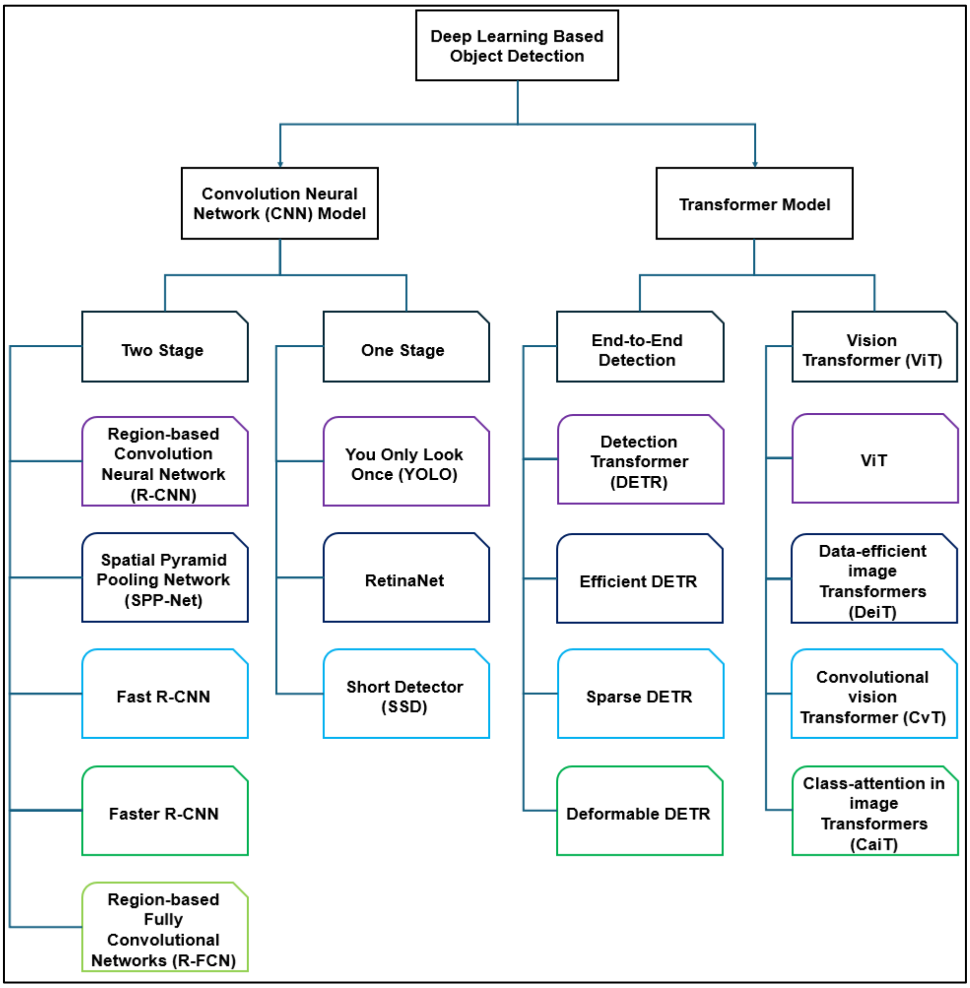


Figure 3 Classification of deep learning-based object detection. *Source:* Sun et al. (2024).

3.3. Application of object detection in PCBs

Zhou et al. (2021) proposed a multiview template matching method for PCB component detection, as opposed to manual inspection. The experiment compared multiview and single-view approaches and demonstrated a significant improvement in performance with the multiview approach. Bahrebar et al. (2022) proposed the use of machine learning to predict PCB corrosion conditions. Predictive models are employed to forecast the levels of electrochemical migration (ECM) and leakage current (LC) within electronic systems subjected to corrosive environments. Supervised machine learning approaches are utilized, with predictions from the ECM and LC models guiding the training process. Commonly used algorithms in this domain include k-nearest neighbors (k-NNs), decision trees (DTs), random forests (RFs), and support vector machines (SVMs). As a result, the SVM and RF algorithms achieve the highest scores in terms of the F1 score, AUC, accuracy, sensitivity, and precision for PCB corrosion prediction under diverse critical factor combinations.

Piliposyan and Khursheed (2022) proposed a hybrid object detection method that combines traditional computer vision with the Siamese neural network (SNN) for detecting hardware Trojan components on PCBs. An SNN is a combination of a CNN and a feedforward fully connected neural network. A hardware Trojan is an extra component implanted on a PCB. A private PCB dataset was used for training the model. The dataset was categorized into three types: small (surface area within 4 to 9 mm^2), medium (surface area within 15 to 50 mm^2), and large (surface area 280 mm^2 and above). The proposed method achieved an overall accuracy of 95.6%.

Lou et al. (2023) proposed a modification of the Vision Transformer (ViT) for multiscale feature extraction. The method uses a multistage ViT architecture, with each stage focusing on capturing features at a specific scale. Subsequently, 1x1 convolution layers are employed between adjacent stages. These convolution layers act as connectors, effectively increasing the dimensionality of the extracted features by combining information from different scales. For the final object detection algorithm, the detection head of YOLOx was adopted. The model was tested on three datasets and achieved a mAP of 69.3% for the MSCOCO 2017 dataset, 86.0% for the PASCAL VOC 2007 dataset, and 99.9% for a private PCB ultrasonic image dataset.

Zhang et al. (2024) proposed LDD-Net, a lightweight network designed for detecting defects on PCBs. LDD-Net comprises three components: lightweight feature extraction (LFE), a multiscale aggregation network (MAN), and LD-Head. The LFE is used for extracting features from images, the MAN is used for fusing multiscale feature maps, and the LD-Head acts as the detector, performing object classification and regression for defect detection. The dataset used is the Test PCB dataset, which is an enhanced version of the public synthesis PCB dataset. The model achieved a mAP of 95.9%.

Fung et al. (2024) proposed the SF-PSPyramid for the Faster R-CNN for PCB defect detection. The structure includes a bottom-up pathway, composed of C1 to C5, and a top-down pathway, composed of P1 to P6. The model achieved 98.6% AP50 for the Deep PCB dataset, whereas for the public synthesis PCB dataset, the model achieved 99.3% AP50. The proposed method, object detection approach, and contribution of the study are summarized in Table 1.

Table 1 Summary of object detection applications in PCB defect detection.

Authors	Proposed Method	Object Detection Approach	Contribution
Zhou et al. (2021)	Multiview template matching	Template matching approach	Has great detection on PCB component
Bahrebar et al. (2022)	Machine Learning	Machine learning approach	SVM and RF algorithms achieved highest score for corrosive condition prediction
Piliposyan & Khursheed. (2022)	Combination of traditional CV and SNN	Hybrid approach	High accuracy of 95.6% for detecting three type of surface area for hardware trojan
Lou et al. (2023)	Vision Transformer (ViT) for multiscale	Deep learning approach	A high mAP of 99.9% for private PCB ultrasonic image dataset
Zhang et al. (2024)	LDD-Net	Deep learning approach	Perform well in detecting PCB defect which achieved 95.9% of mAP
Fung et al. (2024)	SF-PSPyramid	Deep learning approach	Perform well on both public synthesis dataset (99.3% AP@50) and Deep PCB dataset (98.6% AP @50).

3.4. Deep learning technique for PCB vision inspection

Deep learning is a technique rooted in neural networks and inspired by the structure of the human brain. A neural network typically comprises three layers: an input layer, a hidden layer, and an output layer. In the input layer, artificial neurons, also known as nodes or units, function as feature detectors, representing specific data. This information is then transmitted to the hidden layer where computations occur. Each neuron in the hidden layer receives weighted inputs from the previous layer and applies an activation function to compute an output. The final layer, the output layer, generates the desired output. If the output does not match expectations, the weights are adjusted iteratively until the desired output is achieved (Robert, 2024).

Deep learning involves two primary stages: training and testing. During the training stage, a substantial amount of data undergoes a labelling process, which defines identifiable features. Through a comparison of these features, the model learns to identify patterns, enabling accurate predictions and decisions when encountering similar data in the future. In the testing stage, the model applies its acquired knowledge to analyse new data and make predictions (Patel and Thakkar, 2020).

A convolutional neural network (CNN) is a specialized type of neural network designed for analysing visual data, specifically images. Compared with traditional neural networks, CNNs are more intricate and incorporate filters, kernels, and pooling layers. In the CNN architecture, the input is an image, and the second layer involves a convolutional layer where filters and kernels are applied to extract important features from the input image. The convolutional layer produces outputs known as feature maps, which are then passed to the pooling layer. In the pooling layer, feature maps undergo downsampling to reduce computational complexity and extract crucial features. Typically, a CNN includes multiple convolutional layers and a pooling layer. Fully connected layers serve as activation layers for prediction and generating outputs, as illustrated in Figure 4. The adoption of CNNs in PCB vision inspection has significantly enhanced performance, offering benefits such as reduced computational time, robust handling of variable and complex visual data, and improved detection accuracy (Robert, 2024).

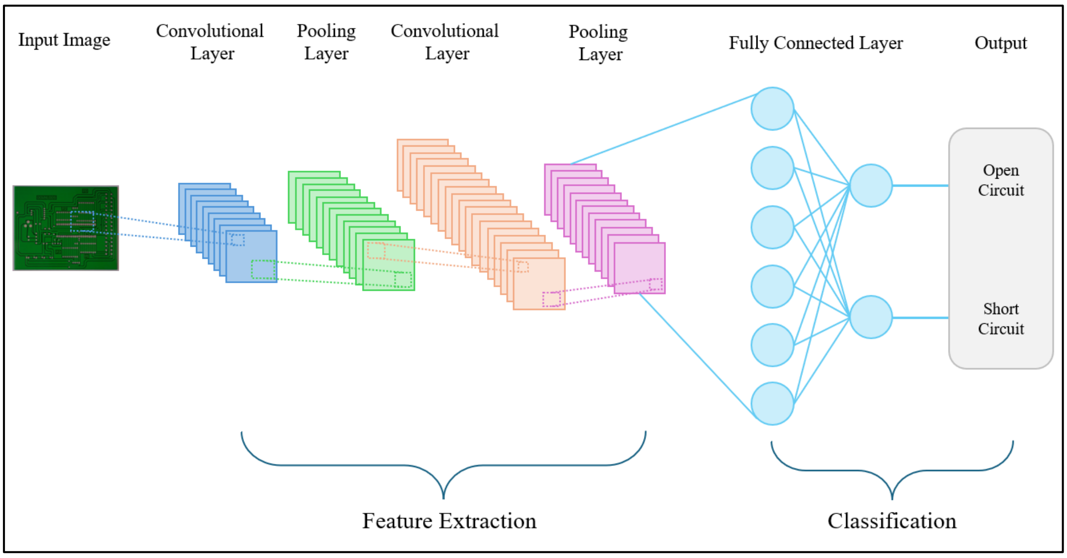


Figure 4 Convolutional neural network architecture. *Source:* Li et al. (2022)

Ding et al. (2019) proposed a tiny defect detection network (TDD-Net) specifically designed to identify tiny defects on PCB surfaces. While it builds upon the Faster R-CNN architecture, TDD-Net incorporates three significant modifications to Faster R-CNN by implementing a data augmentation technique, designing reasonable anchors, and a multiscale feature fusion strategy for the model. The experimental results indicate a 98.80% mAP. Hu & Wang proposed a combination of a feature pyramid network (FPN) with Faster R-CNN and a guided anchor-based region proposal network (GARPN). FPN with Faster R-CNN as the backbone for defect detection. GARPN uses a multiscale RPN rather than a standard RPN. The experimental results indicate a 95.6% mAP.

Xia et al. (2021) proposed a combination of parallel high-definition feature extraction (PHFE), focal loss (FL) and a region-based fully convolutional network (RFCN) for inspecting PCB minor defects. The experimental results indicate a 97.3% mAP, which is 12.3 higher and 6.7 higher than those of YOLOv3 and Faster R-CNN, respectively. Lan et al. proposed an improvement to YOLOv3. There are three improvements to YOLOv3. First, a batch normalization (BN) layer is combined with a convolutional layer. Second, the GIoU loss function is implemented rather than the standard loss function, which is the mean square error (MSE). Finally, the K-means++ clustering algorithm is used. The experimental results indicate a precision of 92.98% mAP compared with SSD (84.67% MAP) but slightly less precision compared with Faster R-CNN (94.72% mAP).

Li et al. (2022) proposed an extended FPN model for hard sample PCB detection. Moreover, the focal loss function is implemented in the model to reduce the loss of samples and increase the weight of hard samples. This approach achieved a 96.2% mAP. Liu et al. proposed Gaussian intersection of union (GsIoU) loss with YOLOv4. The GSIoU addresses the problem of having multiple bounding boxes predicted for a single object. It addresses this by using a Gaussian function to correlate these predictions. Boxes predicted at identical positions are treated collectively. The experimental results revealed an 86.9% AP.

Moreover, Zeng et al. (2022) proposed ABFPN for defect detection. An atrous spatial pyramid pooling (ASPP)-balanced FPN (ABFPN), a novel approach for detecting small objects, combines the strengths of three different techniques for small object detection. ABFPN is flexible and can be utilized in existing object detection systems. Two main parts are used to combine image features: the skip-ASPP module and the balanced module. The proposed improved PCB defect detection (IPDD) framework consists of four parts: an input layer, an enhanced ResNeXt-152 as the backbone, ABFPN as the neck part and a cascaded RCNN as the detection head. The experimental results indicate an 85.59% mAP compared with Faster R-CNN (73.2% mAP) and a slight improvement compared with RefineDet (83.8% mAP).

Moreover, Yu et al. (2022) proposed ES-Net for tiny defect detection. ES-Net consists of three parts: the backbone, neck, and head. The system leverages the CSPDarknet53 network as its backbone for feature extraction, while the aggregated feature guidance module (AFGM) is the neck. The DSH (dynamic scale-aware head) serves as the final module of the network. It utilizes the fused features from the ESP module, which incorporates information from all scales. This allows DSH to perform multiscale object prediction, considering details from both coarse- and fine-grained feature maps. DSH acts as the prediction head of the network. According to the experimental results on the PCB defect dataset, ES-Net with CSPDarknet53 achieves a 97.5% mAP.

Chen et al. (2023) proposed a simple CNN model to classify normal and component defects. The model mainly adopts the method of ensemble learning. This model combines the strengths of multiple mini models. Instead of using one model, it leverages the diverse features extracted by two submodules. The probabilities are used for these initial outputs to generate the fourth output. The experimental results for the proposed CNN model achieve an accuracy of 96.96%.

Ling et al. (2023) proposed an enhancement to the YOLOv8 object detection model. The primary improvements focus on the backbone architecture, which incorporates ghost convolution and C2 focal modules. C2 focal modules address the challenge of detecting densely packed objects across various scales. Traditionally, this is achieved by generating and merging feature maps of different sizes. The proposed approach implements FocalNeXt blocks as a substitute for the original C2f module within YOLOv8. By introducing the C2Focal module, the model achieves competitive performance while maintaining minimal computational costs. In addition, a new Sig-IoU loss function for bounding box regression has been introduced to replace the original loss function for YOLOv8. The experimental results indicate that 87.7% mAP is achieved. The deep learning technique that has been proposed for detecting PCB defects is summarized in Table 2.

Table 2 Summary of deep learning techniques for PCB defect detection.

Authors	Proposed Approach	Contribution	Limitations
Ding et al. (2019)	TDD-Net	Achieved high mAP of 98.8% on detecting six types of defects	• Higher training time • Expensive computational cost
Hu & Wang (2020)	Faster R-CNN with FPN and GARPN	The GARPN has perform well compared to standard RPN for Faster R-CNN	• Unable have a great performance when detecting different PCB dataset, even after adding the training dataset to model
Lan et al. (2021)	Improvement YOLOv3	Success to detect and achieve high precision and improve in detection rate for one stage object detection	• Lower mAP compared to Faster R-CNN
Xia et al. (2021)	PHFE and FL-RFCN	The data enhancement has improved the accuracy of the detection and focal loss function has perform well in deal with imbalance dataset	• The model It is not clear how the model arrives at its defect classifications, which could be a concern in safety-critical applications. • No real-world applications
Yu et al. (2022)	ES-Net	Has successful detect for tiny defect detection	• Even the mAP achieved high accuracy of 76.2% but still not suitable for real industry application
Li et al. (2022)	Extended FPN	Has a better performance compared to original FPN and achieved high precision for hard samples	• Not suitable for real-world industrial environments • Model performance may affect by light and noise
Liu et al. (2022)	Gaussian-IoU loss	Has the ability to solve the problem deal with multiple bounding boxes for single object condition	• Not real-time yet
Zeng et al., 2022	ABFPN	Has achieved high 98.8% AP at 50 for PCB defect detection	• Model exhibits low performance on COCO 2017 dataset, and VisDrone 2019 dataset
Chen et al. (2023)	CNN model based on depth wise separable convolution	Success in achieved high accuracy detection and able to replace the traditional CNN	• Only detected small region rather than whole image
Ling et al. (2023)	Improvement YOLOv8	High performance for 87.7% of mAP at threshold 0.5 and high inference speed has the possibility for real application	• Unable to deal with imbalanced PCB components

4. FPN-RFCN: A State-of-Art Approach

FPN generates feature pyramids with robust semantic information at various scales to obtain useful details on small objects without increasing the amount of calculation and occupied memory. It uses a CNN to generate a set of hierarchical features that encode semantic information at different scales in the pyramid. The different levels of features in the hierarchical pyramid represent the objects in the image and their contextual information from different perspectives (Hu and Wang, 2020).

The region-based fully convolutional network (R-FCN) was proposed by Dai et al. (2016) and is composed of 4 main parts, including the CNN backbone, region proposal network (RPN), classification, and regression, as illustrated in Figure 5. The main contribution of this network is position-sensitive score maps. The R-FCN architecture employs a modified version of the ResNet-101 architecture as its backbone. Any CNN can be used as the backbone for the RFCN. This modification entails the removal of the average pooling layer and fully connected layer and uses only the convolutional layer of ResNet101 to generate the feature maps. Since the last convolutional block in ResNet101 is 2048-dimensional, a randomly initialized 1x1 convolutional layer with 1024 dimensions was attached to reduce the dimension. To capture spatial information within each region of interest (ROI), the RFCN architecture divides each ROI into a k × k grid, creating k^2 individual bins. Each bin in this grid corresponds to a specific position-sensitive score map generated by the final convolutional layer. This layer uses a $k^2(C+1)$-channel convolution to account for both the object category (C) and background (+1) (Du & Wang, 2021).

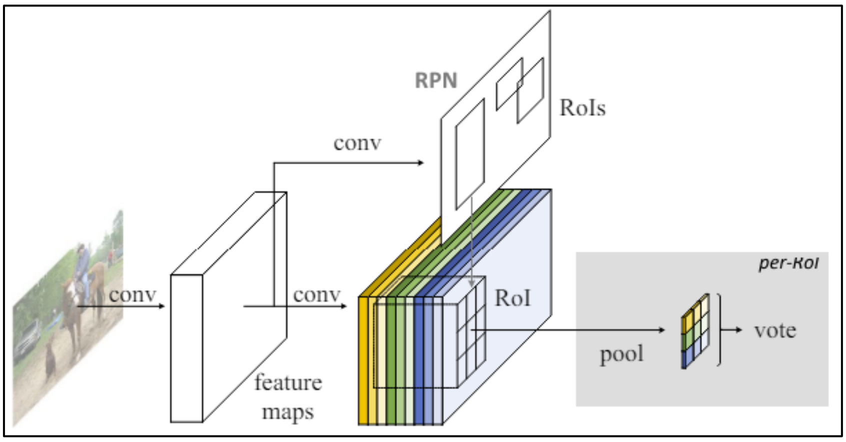


Figure 5 Architecture of the RFCN. *Source:* Dai et al. (2016).

The region proposal network (RPN) is a fully convolutional network that operates on a feature map derived from a CNN. It employs a set of predefined boxes called anchors to generate k anchor boxes of varying sizes and aspect ratios positioned at different locations across the feature map. These anchor boxes are then processed through two fully connected layers, a classification layer and a box regression layer, as depicted in Figure 6 (Zhang et al., 2023). In the RFCN architecture, the RPN identifies candidate regions of interest (RoIs) that potentially correspond to object locations. This efficiency stems from the use of shared feature representations extracted from the image by both the RPN and R-FCN. The R-FCN architecture subsequently utilizes these proposed RoIs in conjunction with the shared features to generate position-sensitive score maps for classification.

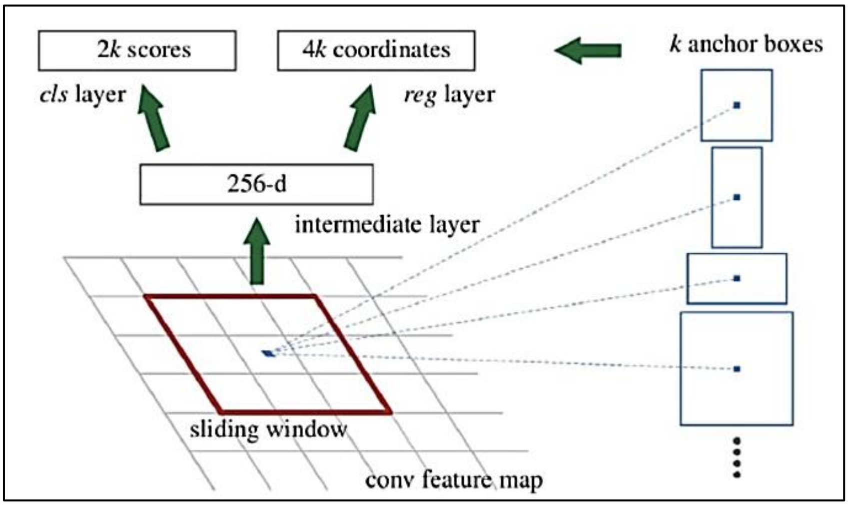


Figure 6 RPN representation. *Source:* Zhang et al. (2023)

4.1. Data augmentation techniques

Recent advancements in artificial intelligence research have led to significant developments in novel network architectures and the exploitation of increasingly powerful graphics processing unit (GPU) computational capabilities. However, the exploration of data augmentation techniques remains an area that warrants more attention. Data augmentation techniques enhance the diversity and adequacy of training datasets by generating synthetic data. This augmented data can be seen as being sampled from a distribution closely resembling real-world data, thereby enabling the dataset to capture a more comprehensive range of characteristics. These methods are versatile and applicable across various computer vision tasks, including semantic segmentation, object detection, and image classification. Image data augmentation broadly encompasses two approaches: basic and advanced techniques. The methodologies for both basic and advanced approaches are illustrated in Figure 7, as outlined by Yang et al. (2022).

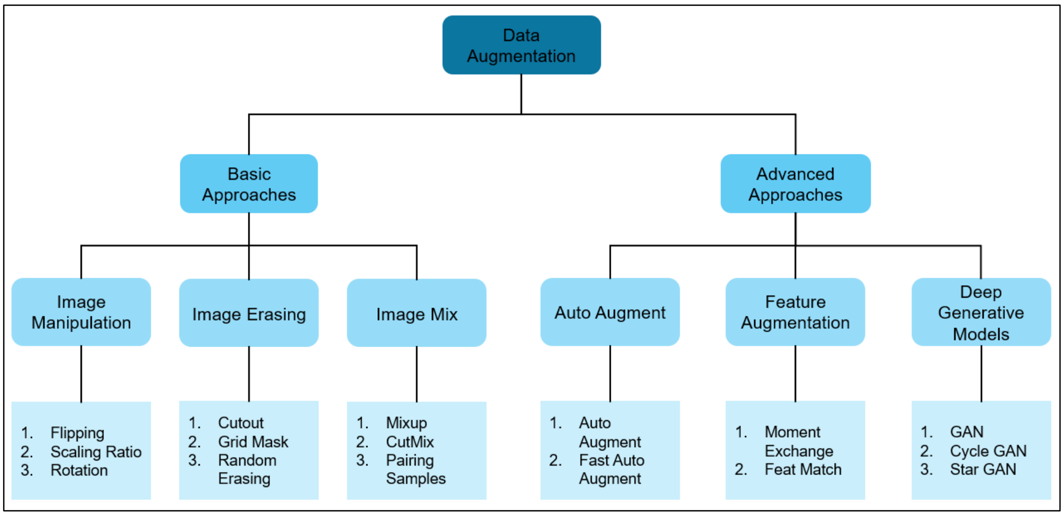


Figure 7 Taxonomy of data augmentation methods. *Source:* Yang et al. (2022)

Furthermore, deep transfer learning (DTL) presents a solution to the problem of limited data, which is a significant hurdle for many deep learning applications. DTL uses knowledge gained from source tasks and datasets to accelerate learning on different but related target tasks. Unlike standard deep learning models initialized with random weights, DTL utilizes pretrained weights from models trained on the source task. This pretraining serves as a foundation, enabling the model to learn the target task more efficiently and achieve superior performance. DTL allows source and target data to have different distributions. On the other hand, DTL involves the transfer of preacquired knowledge from the source task to the target task, and these tasks are not necessarily interrelated or learned simultaneously (Iman et al., 2023). A recent study by Zhuang et al. (2021) proposed four main categories for DTL approaches:

i. Innovation-based approach: This approach focuses on feature selection within source data instances. It applies differential weighting to these features, optimizing the relevance for use with the target data.
ii. Feature-based/mapping-based: This approach aims to make the source and target data more similar by transforming the features into a more homogeneous format. The study has further divided this category into two subcategories:
   a. Asymmetric: This method focuses on aligning the source features with the target features.
   b. Symmetric: This method focuses on identifying a fundamental set of features that are relevant to both the source and target data and transforms both sets of features into new spaces.
iii. Parameter-based/Model-based: This approach prioritizes the reuse of knowledge encoded within a pretrained model. This is achieved by implementing various combinations of freezing, fine-tuning, or adding new layers to the pretrained network architecture.
iv. Relational-based/adversarial-based: This approach focuses on the extraction of transferable features that are applicable to both the source and target data. This is achieved through two primary methods: leveraging the established logical relationships or rules gleaned from the source domain and employing techniques inspired by the architecture and functionality of generative adversarial networks (GANs).

Deep transfer learning (DTL) primarily uses model-based approaches to address the challenge of domain adaptation. Since source and target data often have different distributions, these approaches adjust the network architecture, also known as the model, to overcome this disparity and achieve better performance. In simpler terms, DTL focuses primarily on leveraging model-based techniques for knowledge transfer. There are many ways to adapt models for deep transfer learning (DTL). These techniques often involve a combination of pretraining, freezing, fine-tuning, and/or adding new layers. A deep learning network trained on a source dataset is referred to as a pretrained model, which consists of pretrained layers that encapsulate learned weights. Freezing and fine-tuning are techniques that utilize some or all layers of pretrained models to train the model on target data. Freezing refers to a technique where specific layers within a pretrained model have their own weights or parameters locked, whereas fine-tuning refers to the parameters or weights that utilize pretrained values initialized with pretrained values instead of random initialization for the whole network or specific chosen layers. Another recent DTL technique is based on freezing a pretrained model and adding a new layer to that model for training on target data.

A loss function and a performance metric are both instrumental in evaluating the effectiveness of a deep learning model. However, both serve distinct purposes within the training process. The loss function continuously evaluates the discrepancy between the model's predictions and the actual targets. It plays a crucial role during the training phase. A table summarized by Terven et al. (2023) provides a comparative overview of four loss functions commonly utilized in object detection tasks: smooth L1, IoU loss, focal loss, and YOLO loss. This research focuses on the implementation of a loss function specifically designed for imbalanced datasets. Focal loss (FL) is effective for handling highly imbalanced class problems because it addresses the imbalance between difficult and easy-to-classify examples. Even easy samples contribute minimal loss to the overall training objective, but a larger volume of easy examples will lead to a cumulative loss that may outweigh the loss from hard examples. Thus, FL focuses on learning more on hard examples by downweighting the loss contribution from correctly classified or confidently predicted examples (Lin et al., 2021). The mathematical formula for FL is defined as follows:

$$FL\left(p_t\right) = -\alpha_t(1 - p_t)^{\gamma} \log(p_t) \quad (1)$$

where:

- p_t refer to the predicted probability
- α_t refers to the weighting factor
- γ refers to the focusing parameter
- where p_t represents the model confidence in predicting the correct class for a specific sample.

A higher value close to 1 indicates greater confidence. α_t is the hyperparameter controlling the importance of each sample within the loss calculation. It is often set to the inverse class frequency to balance the loss across all classes. γ is another hyperparameter that determines the degree to which easy examples are downweighted. Usually, the value is set between 2 and 4. A higher γ value leads to a stronger focus on hard-to-classify examples during training.

In addition to focal loss, another approach for handling imbalanced datasets is focal Tversky loss (FTL). This loss function forces the model to focus on harder training examples, which include minority class segments (Sinha & Senapati, 2023). The formula is defined as follows (Das and Zhang, 2020):

$$T(\alpha,\beta) = \frac{\sum_{i=1}^{N} p_{0i} g_{0i}}{\sum_i p_{0i} g_{0i} + \alpha \sum_i p_{0i} g_{0i} + \beta \sum_i p_{0i} g_{0i}} \quad (2)$$

$$FTL(\alpha,\beta,\gamma) = \left(1 - T(\alpha,\beta)\right)^{\gamma} \quad (3)$$

where:

- $T(\alpha, \beta)$ refers to the Tversky index (TI).
- P is the predicted label.
- G is the ground truth label.

p_{0i} indicates the probability that detected object i belongs to an actual object, whereas p_{1i} indicates the probability that detected object i does not belong to an actual object. g_{0i} is 1 for an actual object and 0 for a nonobject; similarly, g_{1i} is 1 for a nonobject and 0 for an actual object. Additionally, $\alpha\ and\ \beta$ are the hyperparameters used to determine the relative importance of penalizing false positives and false negatives. The α and β in the FTL equation act as the balancing factors, whereas the γ controls the difficulty level assigned to the misclassified ROIs.

On the other hand, performance metrics are a critical component of model evaluation following the training process. These metrics assess the model's ability to generalize novel data and generate accurate predictions. Performance metrics enable the comparative analysis of different models or configurations to choose the optimal model (Terven et al., 2023). The standard evaluation criteria for performance metrics include accuracy (Acc), precision (Precision), recall (Recall), the F1 score, and the mean average precision (mAP). The four mathematical metrics were defined as follows and can be computed using various software (Awang et al., 2015):

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \qquad (5)$$

$$Recall = \frac{TP}{TP+FN} \qquad (6)$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \qquad (7)$$

where:

- True positive (TP) represents the number of correct predictions for positive samples.
- True negative (TN) indicates the number of correct predictions for negative samples.
- False negative (FN) refers to the number of incorrect predictions for positive samples.
- False positive (FP) represents the number of incorrect predictions for negative samples.

4.2. Dataset

Zhang and Liu (2021) introduced a Faster R-CNN with an FPN for PCB defect detection. The experiment uses two publicly available datasets, the public synthesis PCB dataset and the deep PCB dataset, and one private dataset, the PCB dataset with components. In this research, two publicly available PCB datasets are combined into one dataset and utilized for the model. The public synthetic PCB dataset contains single-defect images in the RGB color space, whereas the deep PCB dataset offers images of PCBs with multiple defects in grayscale format, as illustrated in Figure 8 and Figure 9. These combinations of datasets were able to enhance model robustness and generalizability. The model is expected to learn spatial patterns from the grayscale images in the Deep PCB dataset and leverage the rich color information from the Public Synthetic PCB dataset. This combined training strategy aims to improve the model's ability to detect a wider variety of defects, including those that may not exhibit distinct color variations. The datasets are divided into three sets: a training set, a validation set, and a test set. Seventy percent of the dataset is used for training the model. The 20% of the dataset is allocated as the validation dataset, which is used for fine-tuning the model, and the remaining 10% of the dataset serves as the test set, which acts as new data to evaluate the model performance.

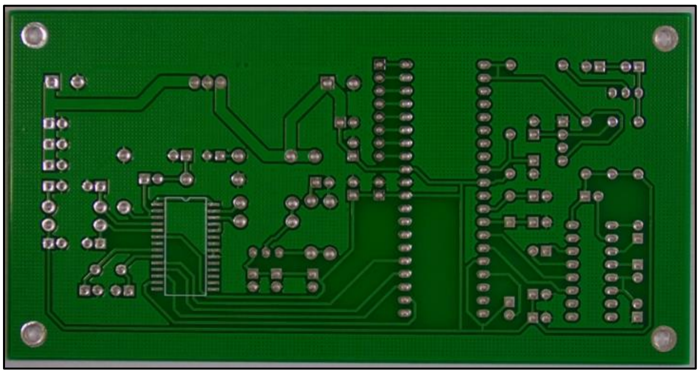


Figure 8 Public synthesis PCB defect image. *Source:* Zhang and Liu (2021)

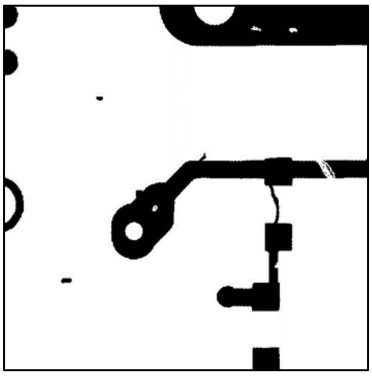


Figure 9 Deep PCB defect image. *Source:* Zhang and Liu (2021)

4.3. Proposed model for PCB defect detection

The architecture for the proposed method is the combination of an FPN with an RFCN. The PCB is defected as the input image and fed into the CNN. The CNN performs feature extraction and generates feature maps as outputs. The feature maps are fed into both the RPN and FPN in parallel. The RPN generates the objectness score and bounding box regression. The bounding box regression is used to refine the bounding box. The refined bounding box acts as a reference for the position-sensitive score map stage. For the FPN stage, the FPN generates multiple levels of feature maps whose low level contains higher resolution to capture fine details, and higher levels contain lower resolution but capture a larger field of view. These multiple levels of feature maps are passed to the position-sensitive score map stage. In the position-sensitive score map stage, the refined bounding boxes help select the most appropriate level of detail from the multiscale feature maps extracted by the FPN. This chosen feature map is then used to generate position-sensitive score maps. Each element in the score maps represents a location within the candidate region and predicts the likelihood of a defect being present. A threshold is applied to these scores, and regions exceeding it are considered potential defects. These position-sensitive score maps are used to vote by using nonmax suppression (NMS), which removes redundant bounding boxes, and the remaining ones with the highest scores are classified to identify defects. These classified defects are then displayed visually on the image. Figure 10 illustrates the proposed model architecture.

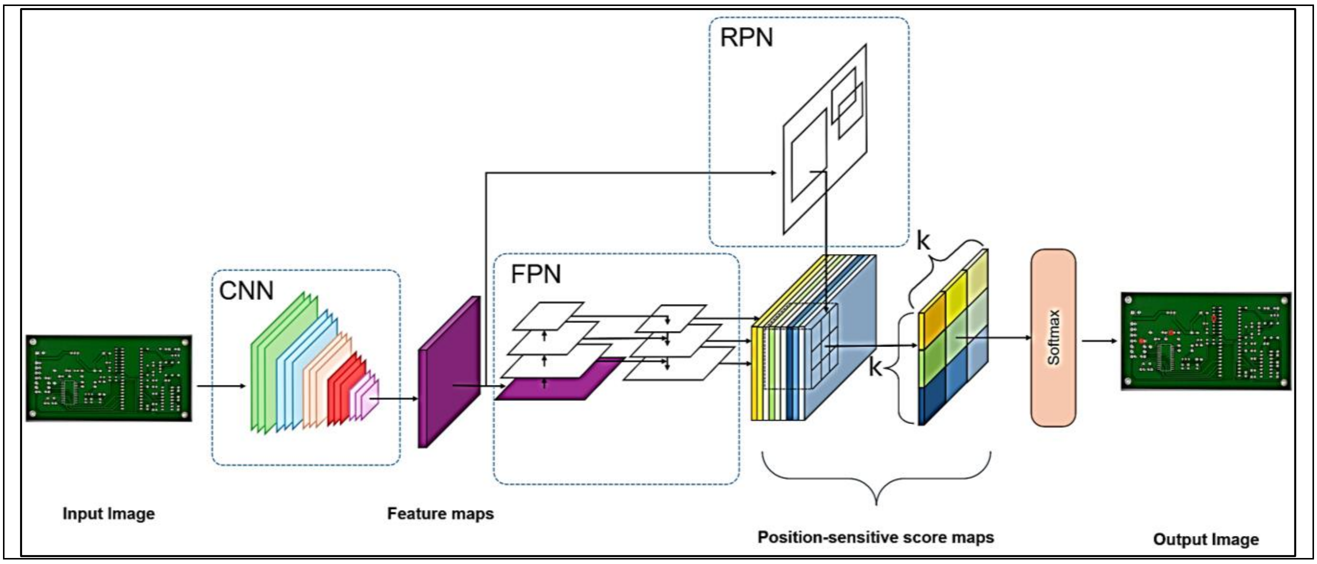


Figure 10 Proposed model for PCB defect detection based on an improved FPN-RFCN method.

5. Final Considerations

In conclusion, the integration of the FPN-RFCN in automated PCB defect detection through deep learning represents a significant improvement in ensuring the reliability and quality of electronic devices. This review highlights the evolution from traditional methods to sophisticated deep learning architectures, emphasizing their ability to detect diverse PCB defects with high precision and efficiency. While challenges such as dataset diversity and model optimization persist, ongoing advancements in deep transfer learning and loss function adaptation promise further enhancements in robustness and real-world applicability. As Industry 4.0 continues to evolve, leveraging these technologies not only enhances manufacturing efficiency but also fosters safer and more reliable consumer electronics, paving the way for a future where zero-defect production becomes a tangible reality.

Acknowledgment

The authors would like to express our gratitude and appreciation to Fakulti Teknologi dan Kejuruteraan Industri dan Pembuatan, Universiti Teknikal Malaysia Melaka, for the support provided in enabling the implementation of this project.

Ethical Considerations

Not applicable.

Conflict of Interest

The authors declare no conflict of interests.

Funding

This research did not receive any financial support.

References

Arif, Z. H., Mahmoud, M. A., Abdulkareem, K. H., Mohammed, M. A., Al-Mhiqani, M. N., Mutlag, A. A., & Damaševičius, R. (2022). Comprehensive Review of Machine Learning (ML) in Image Defogging: Taxonomy of Concepts, Scenes, Feature Extraction, and Classification techniques. *IET Image Processing*, 16(2), 289–310. https://doi.org/10.1049/IPR2.12365

Awang, N., Fauadi, M. H. F. M., & Rosli, N. S. (2015). Image Processing of Product Surface Defect Using Scilab. *Applied Mechanics and Materials* vols. 789–790, pp. 1223–1226. https://doi.org/10.4028/www.scientific.net/amm.789-790.1223

Bahrebar, S., Homayoun, S., & Ambat, R. (2022). Using machine learning algorithms to predict failure on the PCB surface under corrosive conditions. *Corrosion Science*, 206, 110500. https://doi.org/10.1016/J.CORSCI.2022.110500

Boateng, E. Y., Otoo, J., Abaye, D. A., Boateng, E. Y., Otoo, J., & Abaye, D. A. (2020). Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review. *Journal of Data Analysis and Information Processing*, 8(4), 341–357. https://doi.org/10.4236/JDAIP.2020.84020

Chen, I. C., Hwang, R. C., & Huang, H. C. (2023). PCB Defect Detection Based on Deep Learning Algorithm. *Processes*, Vol. 11, Page 775, 11(3), 775. https://doi.org/10.3390/PR11030775

Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of Image Classification Algorithms Based on Convolutional Neural Networks. *Remote Sensing*, Vol. 13, Page 4712, 13(22), 4712. https://doi.org/10.3390/RS13224712

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. *Advances in Neural Information Processing Systems*, 379–387. https://arxiv.org/abs/1605.06409v2

Das, K., & Zhang, Q. (2020). Convolutional Recurrent Residual U-Net Embedded with Attention Mechanism and Focal Tversky Loss Function for Cancerous Nuclei Detection. Queen Mary University of London, United Kingdom. https://arxiv.org/abs/2010.04416v1

Ding, R., Dai, L., Li, G., & Liu, H. (2019). TDD-net: a tiny defect detection network for printed circuit boards. *CAAI Transactions on Intelligence Technology*, 4(2), 110–116. https://doi.org/10.1049/TRIT.2019.0019

Dong, Y., Wang, Z., & Guo, W. (2022). Overview of edge detection algorithms based on mathematical morphology. In *Proceedings of Electronic and Automation Control Conference (IAEAC),* 2022-October, 1321–1326. https://doi.org/10.1109/IAEAC54830.2022.9930043

Du, Y., & Wang, Q. (2021). Multi-Angle Face Detection Based on Improved RFCN Algorithm Using Multi-Scale Training. In *Proceedings of 2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing*, ICSP 2021, 319–322. https://doi.org/10.1109/ICSP51882.2021.9408676

Fung, K. C., Xue, K. W., Lai, C. M., Lin, K. H., & Lam, K. M. (2024). Improving PCB defect detection using selective feature attention and pixel shuffle pyramid. *Results in Engineering*, 21, 101992. https://doi.org/10.1016/J.RINENG.2024.101992

Ge, C., Wang, J., Wang, J., Qi, Q., Sun, H., & Liao, J. (2020). Towards automatic visual inspection: A weakly supervised learning method for industrial applicable object detection. *Computers in Industry*, 121, 103232. https://doi.org/10.1016/J.COMPIND.2020.103232

Georgiev, A. G., Hristov, G., Plamen Zahariev, A., & Diyana Kinaneva, A. (2023). Exploring Object Detection Algorithms: A Comprehensive Overview and Comparative Study 15. *Proceedings of University of Ruse*-2023, 62.

Ghaffari, S., Soleimani, P., Li, K. F., & Capson, D. W. (2020). Analysis and Comparison of FPGA-Based Histogram of Oriented Gradients Implementations. *IEEE Access*, 8, 79920–79934. https://doi.org/10.1109/ACCESS.2020.2989267

Hu, B., & Wang, J. (2020a). Detection of PCB Surface Defects with Improved Faster-RCNN and Feature Pyramid Network. *IEEE Access*, 8, 108335–108345. https://doi.org/10.1109/ACCESS.2020.3001349

Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A Review of Deep Transfer Learning and Recent Advancements. *Technologies*, Vol. 11, Page 40, 11(2), 40. https://doi.org/10.3390/TECHNOLOGIES11020040

Lan, Z., Hong, Y., & Li, Y. (2021). An improved YOLOv3 method for PCB surface defect detection. In *Proceedings of 2021 IEEE International Conference on Power Electronics, Computer Applications, ICPECA 2021*, 1009–1015. https://doi.org/10.1109/ICPECA51329.2021.9362675

Li, C. J., Qu, Z., Wang, S. Y., Bao, K. H., & Wang, S. Y. (2022). A Method of Defect Detection for Focal Hard Samples PCB Based on Extended FPN Model. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 12(2), 217–227. https://doi.org/10.1109/TCPMT.2021.3136823

Li, Y., Miao, N., Ma, L., Shuang, F., & Huang, X. (2023). Transformer for object detection: Review and benchmark. *Engineering Applications of Artificial Intelligence*, 126, 107021. https://doi.org/10.1016/J.ENGAPPAI.2023.107021

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems,* 33(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Lin, W., Wu, P., & Xiao, X. (2021). Boundary Focal Loss for Class Imbalanced Learning. In *Proceedings of 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*. https://doi.org/10.1109/CISP-BMEI53629.2021.9624339

Ling, Q., Isa, N. A. M., & Asaari, M. S. M. (2023). Precise Detection for Dense PCB Components Based on Modified YOLOv8. *IEEE Access*, 11, 116545–116560. https://doi.org/10.1109/ACCESS.2023.3325885

Linton, P., Morgan, M. J., Read, J. C. A., Vishwanath, D., Creem-Regehr, S. H., & Domini, F. (2023). New Approaches to 3D Vision. *Philosophical Transactions of the Royal Society B*, 378(1869). https://doi.org/10.1098/RSTB.2021.0443

Liu, X., Hu, J., Wang, H., Zhang, Z., Lu, X., Sheng, C., Song, S., & Nie, J. (2022). Gaussian-IoU loss: Better learning for bounding box regression on PCB component detection. *Expert Systems with Applications*, 190, 116178. https://doi.org/10.1016/J.ESWA.2021.116178

Lou, L., Lu, K., & Xue, J. (2023). Multi-Scale Vision Transformer for Defect Object Detection. *Procedia Computer Science*, 222, 397–406. https://doi.org/10.1016/J.PROCS.2023.08.178

LU, H., & MAZUMDER, R. (2020). Randomized Gradient Boosting Machine. *SIAM Journal on Optimization*, 30(4), 2780–2808. https://doi.org/10.1137/18M1223277

Mercier, J. P., Garon, M., Giguere, P., & Lalonde, J. F. (2021). Deep template-based object instance detection. In *Proceedings of 2021 IEEE Winter Conference on Applications of Computer Vision,* WACV 2021, 1506–1515. https://doi.org/10.1109/WACV48630.2021.00155

Mohan, L., Pant, J., Suyal, P., & Kumar, A. (2020). Support Vector Machine Accuracy Improvement with Classification. In *Proceedings of 12th International Conference on Computational Intelligence and Communication Networks,* 477–481. https://doi.org/10.1109/CICN49253.2020.9242572

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., & Walsh, J. (2020). Deep Learning vs. Traditional Computer Vision. *Advances in Intelligent Systems and Computing*, 943, 128–144. https://doi.org/10.1007/978-3-030-17795-9_10

Pang, G., Shen, C., Cao, L., & Hengel, A. Van Den. (2021). Deep Learning for Anomaly Detection. *ACM Computing Surveys (CSUR),* 54(2). https://doi.org/10.1145/3439950

Patel, P., & Thakkar, A. (2020). The upsurge of deep learning for computer vision applications. *International Journal of Electrical and Computer Engineering (IJECE),* 10(1), 538–548. https://doi.org/10.11591/ijece.v10i1.pp538-548

Piliposyan, G., & Khursheed, S. (2022). Computer Vision for Hardware Trojan Detection on a PCB Using Siamese Neural Network. *2022 IEEE Physical Assurance and Inspection of Electronics*, https://doi.org/10.1109/PAINE56030.2022.10014967

Rosli, N.S., Fauadi, M., Awang, N.F., Noor, A.Z.M. (2018). Vision-based defects detection for glass production based on improved image processing method. Journal of Advanced Manufacturing Technology, 12(1(1)), 203-212.

Rosli, N.S., Fauadi, M., Awang, N. (2016). Some technique for an Image of defect in inspection process based on image processing. Journal of Image and Graphics, 4(1), 55-58.

Schubeck, J., Koblah, D., Botero, U. J., & Forte, D. (2021). A Comprehensive Taxonomy of PCB Defects. University of Florida, Gainesville, USA.

Silva, L. H. D. S., Azevedo, G. O. D. A., Fernandes, B. J. T., Bezerra, B. L. D., Lima, E. B., & Oliveira, S. C. (2019). Automatic Optical Inspection for Defective PCB Detection Using Transfer Learning. *2019 IEEE Latin American Conference on Computational Intelligence*. https://doi.org/10.1109/LA-CCI47412.2019.9037036

Sinha, A., & Senapati, K. K. (2023). Analysing Satellite Images using Segmentation with U-net and Focal Tversky Loss. In *Proceedings of 2023 6th International Conference on Information Systems and Computer Networks*. https://doi.org/10.1109/ISCON57294.2023.10112138

Sun, Y., Sun, Z., & Chen, W. (2024). The evolution of object detection methods. *Engineering Applications of Artificial Intelligence*, 133, 108458. https://doi.org/10.1016/J.ENGAPPAI.2024.108458

Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A., & Chavez-Urbiola, E. A. (2023). Loss Functions and Metrics in Deep Learning. https://arxiv.org/abs/2307.02694v2

Udaya Sankar, V., Lakshmi, · Gayathri, & Siva Sankar, · Y. (2022). A Review of Various Defects in PCB. *Journal of Electronic Testing* 1, 3. https://doi.org/10.1007/s10836-022-06026-7

Wang, H., Pan, C., Guo, X., Ji, C., & Deng, K. (2021). From object detection to text detection and recognition: A brief evolution history of optical character recognition. *Wiley Interdisciplinary Reviews: Computational Statistics*, 13(5), e1547. https://doi.org/10.1002/WICS.1547

Wang, Z., Wang, E., & Zhu, Y. (2020). Image segmentation evaluation: a survey of methods. *Artificial Intelligence Review*, 53(8), 5637–5674. https://doi.org/10.1007/S10462-020-09830-9/METRICS

Xia, S., Wang, F., Xie, F., Huang, L., Wang, Q., & Ling, X. (2021). An Efficient and Robust Target Detection Algorithm for Identifying Minor Defects of Printed Circuit Board Based on PHFE and FL-RFCN. *Frontiers in Physics,* 9, 661091. https://doi.org/10.3389/FPHY.2021.661091/BIBTEX

Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., & Shen, F. (2022). Image Data Augmentation for Deep Learning: A Survey. https://arxiv.org/abs/2204.08610v2

Yu, X., Lyu, W., Zhou, D., Wang, C., & Xu, W. (2022). ES-Net: Efficient Scale-Aware Network for Tiny Defect Detection. *IEEE Transactions on Instrumentation and Measurement*, 71. https://doi.org/10.1109/TIM.2022.3168897

Zeng, N., Wu, P., Wang, Z., Li, H., Liu, W., & Liu, X. (2022). A Small-Sized Object Detection Oriented Multi-Scale Feature Fusion Approach with Application to Defect Detection. *IEEE Transactions on Instrumentation and Measurement,* 71. https://doi.org/10.1109/TIM.2022.3153997

Zhang, L., Chen, J., Chen, J., Wen, Z., & Zhou, X. (2024). LDD-Net: Lightweight printed circuit board defect detection network fusing multi-scale features. *Engineering Applications of Artificial Intelligence,* 129, 107628. https://doi.org/10.1016/J.ENGAPPAI.2023.107628

Zhang, Q., & Liu, H. (2021). Multi-scale defect detection of printed circuit board based on feature pyramid network. In *Proceedings of 2021 IEEE International Conference on Artificial Intelligence and Computer Applications*, 911–914. https://doi.org/10.1109/ICAICA52286.2021.9498174

Zhang, W., Zhu, Q., Li, Y., & Li, H. (2023). MAM Faster R-CNN: Improved Faster R-CNN based on Malformed Attention Module for object detection on X-ray security inspection. *Digital Signal Processing*, 139, 104072. https://doi.org/10.1016/J.DSP.2023.104072

Zhao, K., Han, Q., Zhang, C. Bin, Xu, J., & Cheng, M. M. (2022). Deep Hough Transform for Semantic Line Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9), 4793–4806. https://doi.org/10.1109/TPAMI.2021.3077129

Zhou, J., Liu, Y., Zhang, X., & Yang, Z. (2021). Multi-view based template matching method for surface defect detection of circuit board. *Journal of Physics: Conference Series*, 1983(1), 012063. https://doi.org/10.1088/1742-6596/1983/1/012063

Zhu, Y., Detig, C., Kane, S., & Lourie, G. (2022). Kinematic Motion Analysis with Volumetric Motion Capture. In *Proceedings of International Conference on Information Visualisation*, 2022-July, 61–66. https://doi.org/10.1109/IV56949.2022.00019

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. In *Proceedings of the IEEE*, 109(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object Detection in 20 Years: A Survey. In *Proceedings of the IEEE*, 111(3), 257–276. https://doi.org/10.1109/JPROC.2023.3238524