Towards High Quality PCB Defect Detection Leveraging State-of-the-Art Hybrid Models

Tuan Anh Nguyen, Hoanh Nguyen*
Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

***Abstract*—The automatic detection of defects in printed circuit boards (PCBs) is a critical step in ensuring the reliability of electronic devices. This paper introduces a novel approach for PCB defect detection. It incorporates a state-of-the-art hybrid architecture that leverages both convolutional neural networks (CNNs) and transformer-based models. Our model comprises three main components: a Backbone for feature extraction, a Neck for feature map refinement, and a Head for defect prediction. The Backbone utilizes ResNet and Bottleneck Transformer blocks, which are proficient at highlighting small defect features and overcoming the shortcomings of previous models. The Neck module, designed with Ghost Convolution, refines feature maps. It reduces computational demands while preserving the quality of feature representation. This module also facilitates the integration of multi-scale features, essential for accurately detecting a wide range of defect sizes. The Head employs a Fully Convolutional One-stage detection approach, allowing for the prediction process to proceed without reliance on predefined anchors. Within the Head, we incorporate the Wise-IoU loss to refine bounding box regression. This optimizes the model's focus on high-overlap regions and mitigates the influence of outlier samples. Comprehensive experiments on standard PCB datasets validate the effectiveness of our proposed method. The results show significant improvements over existing techniques, particularly in the detection of small and subtle defects.**

***Keywords*—PCB defect detection; hybrid neural network; bottleneck transformer; ghost convolution; wise-IoU loss**

I. Introduction

PCBs are the cornerstone of modern electronics, providing a critical framework for the interconnection of electronic components. They consist of a complex network of conductive pathways, tracks, and traces etched onto a non-conductive substrate, enabling the integration of various components such as resistors, capacitors, and integrated circuits to form functional electronic devices. The integrity of these boards is paramount, as any defects can lead to malfunctioning or failure of the electronic equipment. PCB defect detection, therefore, is a vital process in the manufacturing industry, aimed at identifying and rectifying flaws such as short circuits, open circuits, missing components, or misalignments. Traditionally performed by human inspectors, this process has increasingly been entrusted to automated systems that leverage advanced imaging technologies and machine learning algorithms. These systems offer greater accuracy, consistency, and efficiency in detecting a wide array of subtle and overt flaws that might be overlooked by the human eye, ensuring high-quality outputs in the fast-paced production environments that define today's electronic manufacturing sector.

Traditional image processing techniques for PCB defect detection typically involve a sequence of algorithmic steps such as noise reduction, thresholding, edge detection, and pattern recognition to analyze images of PCBs for anomalies. These methods often start with pre-processing to enhance image quality, followed by segmentation to isolate regions of interest. Techniques like morphological operations may be used to highlight features of defects, and template matching could be employed to compare segments against known good patterns. While these techniques are deterministic and relatively straightforward to implement, they come with significant shortcomings. They tend to be highly sensitive to variations in lighting, alignment, and image quality, leading to false positives or negatives. Additionally, traditional methods may struggle with the complexity of modern PCBs, which can have intricate designs and high component densities. These methods can also be computationally intensive and inflexible, requiring manual tuning and adjustments when dealing with different types of PCBs or new defect profiles, limiting their scalability and adaptability in fast-evolving manufacturing environments.

In recent years, deep learning has revolutionized the field of artificial intelligence, leading to significant advancements in various domains such as computer vision, natural language processing, autonomous vehicles, and medical diagnostics [1, 2]. At its core, deep learning utilizes neural networks with multiple layers to learn representations of data with multiple levels of abstraction, enabling the discovery of intricate structures in large datasets. As a result, applications that were once thought to be challenging, like image and speech recognition, have seen substantial improvements in accuracy and efficiency. Leveraging these developments, deep learning has also been proposed for PCB defect detection, representing a paradigm shift from traditional image processing techniques. Unlike conventional methods, which rely on hand-engineered features and are prone to performance degradation under variations in lighting and complex patterns, deep learning models can automatically learn to identify defects from data. These models, particularly convolutional neural networks (CNNs), have shown remarkable success in detecting intricate and subtle anomalies on PCBs by learning features directly from the raw pixels. However, despite their success, current deep learning methods for PCB defect detection still face challenges. They require large annotated datasets to learn effectively, which can be expensive and time-consuming to create. Moreover, they may not generalize well across different

PCB designs or manufacturing processes without extensive retraining or fine-tuning. To address these shortcomings, the method proposed in this paper integrates advanced neural network architectures that enhance feature extraction and defect localization capabilities, while also employing data augmentation and specialized loss functions to improve model robustness and generalizability. This approach aims to overcome the limitations of both traditional image processing and current deep learning techniques, providing a more reliable and adaptable solution for PCB defect detection.

The rest of the paper is organized as follows: Section II presents related studies; Section III details the proposed model; Section IV describes the experiments and results; Section V provides the conclusions.

II. Related Work

The emergence of end-to-end deep learning technology [3, 4] has introduced new opportunities for PCB fault detection. Currently, extensive research is being carried out on PCB defect detection methods that leverage deep learning. Mingu et al. [5] presented a novel contactless inspection method that utilizes deep learning to analyze thermal images for the detection of PCBA defects. The authors explore the efficacy of combining a rule-based object detection approach, employing a structural similarity index map, with advanced deep learning techniques including CNNs, region with CNN features, and autoencoders, thereby enhancing the accuracy and reliability of contactless PCBA inspection methods. Sik-Ho et al. [6] introduced PCBMTL, multitask learning model designed to concurrently tackle classification and segmentation tasks, specifically tailored for scenarios with limited data availability. This model leverages the intrinsic correlation between segmentation knowledge and classification tasks, significantly enhancing the classification accuracy even when only a sparse dataset is available. Gor et al. [7] proposed an Automated Visual Inspection (AVI) methodology for detecting hardware trojans (HTs) on PCBs, utilizing imagery from a low-cost digital optical camera. This method combines traditional computer vision techniques with a dual-tower Siamese Neural Network (SNN), structured within a three-stage pipeline for effective HT detection. To address the issues of inadequate accuracy and speed in visual matching systems, the study in [8] introduced a deep learning-based alignment system utilizing YOLOv5. This system enhances production efficiency by preprocessing images captured by an industrial camera, delineating sensitive areas rich in feature points for improved alignment accuracy. Naifu et al. [9] employed techniques such as relative position estimation, spatially adjacent similarity, and k-means clustering of patches to discern finely classified semantic features, followed by a local image patch completion network that learns the feature consistency between these local patches and the background, using the disparities between the estimated and original image patches to effectively identify anomaly areas in PCBs.

To enhance the efficiency of current defect detection algorithms, [10] introduced RAR-SSD, a novel method combining multiscale PCB defect target detection with an attention mechanism. This approach integrates a lightweight receptive field block module (RFB-s) with an attention mechanism, effectively focusing on crucial features across various channels without escalating computational demands, and incorporates a feature fusion module that synergizes low-level and high-level feature information, resulting in a comprehensive feature map that significantly boosts fault recognition accuracy. JiaYou et al. [11] introduce an advanced deep learning network specifically designed to tackle the challenge of detecting small or variable defects on PCBs in real-time. The proposed improvements include a unique multi-scale feature pyramid network that boosts tiny defect detection by incorporating context information and a refined complete intersection over union loss function that accurately targets and identifies these minuscule defects. CS-ResNet [12] introduced a new model, which innovates upon the standard ResNet by incorporating a cost-sensitive adjustment layer. This model specifically addresses class imbalance by assigning greater weights to minority real defects based on their degree of imbalance, and optimizes performance through the minimization of a weighted cross-entropy loss function. Boyuan et al. [13] presented a cutting-edge PCB defect detection method utilizing YOLOv7. Additionally, the integration of the CBAM attention mechanism with a feature fusion module enables the model to selectively focus on pertinent feature channels and spatial locations, significantly boosting the discriminative power of the feature representation and thereby increasing overall accuracy. KD-LightNet [14] introduced an efficient and lightweight defect detection network optimized for edge computing scenarios. The network architecture, LightNet, is crafted using structure reparameterization to boost feature extraction capabilities while reducing model complexity.

III. Method

In this section, we provide details of our approach for PCB defect detection. Fig. 1 illustrates the overall structure of the proposed model, which includes three modules: Backbone for extracting features from the input image, Neck for enhancing the feature maps, and Head for making predictions. Initially, the input image is processed by the Backbone, consisting of multiple layers that perform feature extraction. Subsequently, the extracted features are refined by the Neck module, which is designed to enhance and integrate the feature maps at different scales. Finally, the Head module takes over, comprising three key components: Classification, Center-ness, and Regression, which work collectively to output the final defect detection results. Details of each module will be explained in the following subsections.

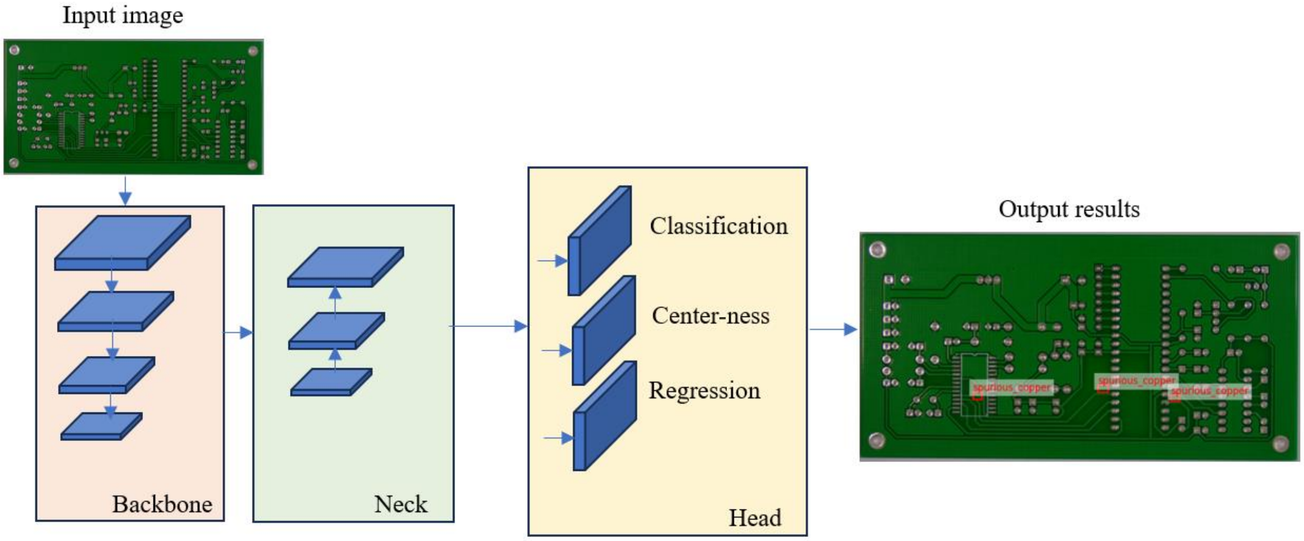

Fig. 1. Overall structure of the proposed model.

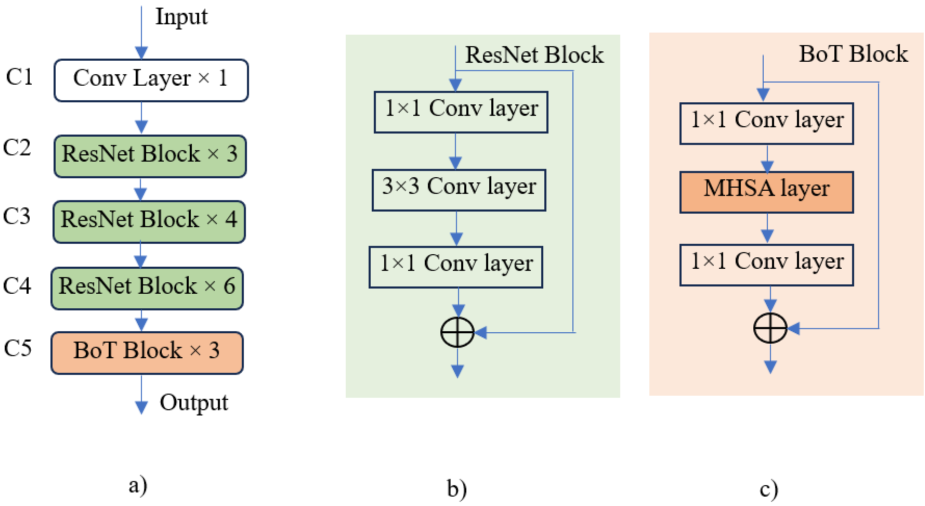

Fig. 2. The structure of the feature extraction network (a) which includes input (b) ResNet Block (c) BoT block.

A. Feature Extraction with Self-Attention Mechanism

1) The backbone network proposed for PCB defect detection is a critical component of the object detection system, designed to process input images and extract relevant features that are essential for identifying defects. The advanced architecture of this backbone is built upon a combination of convolutional layers and ResNet blocks [15], further enhanced with Bottleneck Transformer (BoT) blocks [16]. The structure is outlined in Fig. 2, which depicts the sequential layers and blocks within the network. In detail, the backbone begins with a single convolutional layer (C1) that performs initial feature extraction. This is followed by a series of ResNet blocks (C2, C3, C4) that apply residual learning to prevent the vanishing gradient problem and allow deeper networks to learn effectively. Each ResNet block consists of a bottleneck design with three convolutional layers: a 1×1 convolution that reduces the dimensionality, a 3×3 convolution that processes features, and another 1×1 convolution that restores dimensionality. These blocks are equipped with skip connections that add the input of the block to its output, facilitating the training of deep networks by allowing gradients to flow through.

2) The novelty of this architecture lies in the integration of BoT blocks (C5), which introduce a multi-head self-attention (MHSA) mechanism within the transformer architecture [17]. Each BoT block is comprised of a 1×1 convolution layer followed by an MHSA layer and another 1×1 convolution layer. The MHSA layer enables the network to focus on different parts of the image when extracting features, which is particularly beneficial for detecting small objects-a common challenge in PCB defect detection. This capability is contrasted with the DETR (Detection Transformer) model [3], which shows improvements in detecting larger objects but not smaller ones. The use of BoT blocks in the backbone could potentially address this shortfall, enhancing the model's ability to recognize smaller defects on PCBs that are often difficult to detect. In the BoT block, the MHSA mechanism efficiently captures long-range dependencies across the input feature map. By utilizing multiple attention heads, MHSA is able to concurrently process and focus on various aspects of the semantic space within the feature map. This allows the model to consider information from different representation subspaces at reduced computational costs. The operation of MHSA is as follows:

$$MHSA(Q,K,V) = Concatenation(H^0, H^1, H^2, H^3) \quad (1)$$

where, Q, K, and V represent three linear layers used for computing queries, keys, and values in a standard self-attention task. H^i represent the head of the self-attention mechanism as follows:

$$H^i = Softmax(Q_i K_i^T + qr^T) \times V_i, i \in [0,3] \qquad (2)$$

$$qr = (R_h + R_w) \times Q \qquad (3)$$

where, R_h ans R_w represent height and width relative position, respectively.

B. *Improving Multi-scale Feature with Ghost Convolution*

The neck network for PCB defect detection is designed based on Ghost Convolution [18], as depicted in Fig. 3. This network serves as an intermediary between the feature-rich output from the backbone and the predictive head of the model, enhancing the feature maps for more accurate defect localization. Starting from the deepest layer (C5), the network utilizes Ghost Convolution layers, which are designed to generate more feature maps from fewer intrinsic maps, thereby reducing computational requirements while maintaining effective representation capacity. This is followed by an upsampling step, which increases the resolution of the feature maps to match the scale of the subsequent layer. The upscaled features are then concatenated with the features from the previous layer (C4), integrating multi-level semantic information. This process is repeated as the network proceeds to shallower layers (C4 to C3). Each time, the Ghost Convolution layers generate rich feature representations that are then upsampled and concatenated with features from earlier in the network. This concatenation ensures that the final feature maps encompass both high-level semantic information and finer, low-level details, which is crucial for detecting the often-minute anomalies present in PCBs. The repeated pattern of Ghost Convolution, upsampling, and concatenation progressively enriches the feature maps, culminating in a comprehensive composite that feeds into the detection head. The head then uses these refined features to make precise predictions about the presence, location, and types of defects on the PCB. This neck architecture, with its efficient and hierarchical processing, is particularly well-suited for the demands of PCB defect detection, where the ability to discern subtle and small-scale imperfections is key.

1) Ghost convolution: Ghost Convolution is an innovative approach to convolutional neural network design that aims to reduce computational workload and model complexity without sacrificing performance. The core idea behind Ghost Convolution is to generate additional feature maps, known as "ghost" features, from inexpensive operations on the original convolutional features. This is achieved by applying a series of linear transformations, such as simple arithmetic operations or small-kernel convolutions, to the output of standard convolutional layers. The original set of feature maps is obtained through regular convolution operations, which can be computationally intensive. Then, for each of these original maps, several ghost feature maps are derived using the lightweight transformations. These ghost maps are capable of capturing variations and fine details by reusing the information present in the original feature maps, effectively augmenting the feature space with minimal extra computation. This process substantially reduces the number of direct convolutions that the network needs to perform, thus decreasing the number of parameters and the computational cost. Despite this reduction, Ghost Convolution preserves the network's capacity to encode rich and complex representations of the input data, making it particularly useful for resource-constrained environments or applications where efficiency is paramount, such as mobile devices, embedded systems, or real-time applications.

For a standard convolution, the number of FLOPs is calculated as follows:

$$\text{FLOPs}_{Standard} = H_{out} \times W_{out} \times N_{out} \times (C_{in} \times K_h \times K_w + 1) \qquad (4)$$

where, H_{out} and W_{out} are the height and width of the output feature map; N_{out} is the number of output channels; C_{in} is the number of input channels; K_h and K_w are the height and width of the kernel.

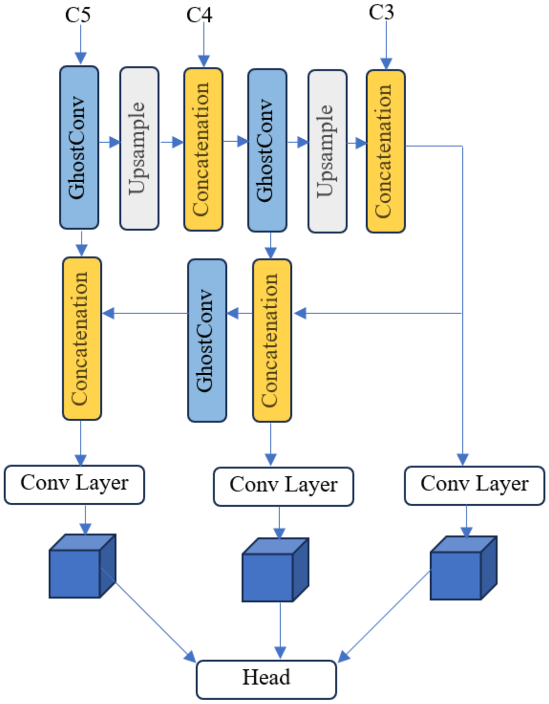

Fig. 3. The neck network with ghost convolution layers.

For Ghost convolution, we first calculate the FLOPs for the initial standard convolution that generates the intrinsic feature maps, and then add the FLOPs for the linear operations used to generate the ghost feature maps. The equation for Ghost convolution is:

$$\text{FLOPs}_{Ghost} = H_{out} \times W_{out} \times N_{int} \times (C_{in} \times K_h \times K_w + 1) + H_{out} \times W_{out} \times N_{ghost} \times \left(N_{int} \times K_{hghost} \times K_{wghost} + 1\right) \quad (5)$$

where, N_{int} is the number of intrinsic output channels produced by the initial standard convolution; N_{ghost} is the number of ghost channels generated per intrinsic channel; K_{hghost} and K_{wghost} are the height and width of the kernel for generating the ghost feature maps, which are typically much smaller than the original convolution kernel size.

The term $N_{int} \times (C_{in} \times K_h \times K_w + 1)$ calculates the FLOPs for the initial convolution, and the term $N_{ghost} \times \left(N_{int} \times K_{hghost} \times K_{wghost} + 1\right)$ calculates the FLOPs for generating the ghost feature maps. Typically, N_{int} is much less than N_{out} and the kernel size for ghost operations (K_{hghost}, K_{wghost}) is smaller, leading to a significant reduction in FLOPs compared to standard convolution.

C. Detection Head with Wise-IoU Loss

1) We employ FCOS head [19] on each output feature layer. FCOS divides its detection head into three branches: the classification branch, the bounding box regression branch, and the centerness branch. In classification branch, a Focal Loss [20] is used to address class imbalance by reducing the weight of easy negatives. The centerness branch uses a binary cross-entropy loss that guides the model to predict higher centerness values for locations closer to the center of an object. For the bounding box regression branch, Wise-IoU loss [21] is employed. This is a novel loss function that modulates the geometric penalty based on the overlap between the predicted bounding box and the ground truth. If the overlap is high, the penalty is reduced, which helps the model to better refine boxes that are already largely accurate. The Wise-IoU loss also includes an outlier penalty term that increases the loss for poor predictions, preventing the model from being overly influenced by difficult or mislabeled examples. The formula for the Wise-IoU loss function is shown as follows:

$$L_{Wise-IoU} = r \times L_{IoU} \times exp(\frac{\left(x-x_{gt}\right)^2+\left(y-y_{gt}\right)^2}{(W_g^2+H_g^2)}) \quad (6)$$

where, r represents the gradient gain.

2) The Wise-IoU loss specifically enhances the bounding box regression branch by incorporating a distance attention mechanism that scales the loss based on the distance metric between the anchor and the target frame. This scaling ensures that when the predicted box is already close to the ground truth (high IoU), the model is encouraged to make finer adjustments rather than over-penalizing small discrepancies. Moreover, by introducing a gradient attenuation factor for outliers, the Wise-IoU loss ensures that samples with poor quality predictions do not dominate the gradient update during backpropagation, thus stabilizing training and steering the model away from local optima that are not generalizable. This thoughtful design of the loss function supports more precise localization in PCB defect detection, which is critical for ensuring the accurate identification of defects.

IV. Experiments

A. Dataset and Experimental Setup

The dataset utilized in this study is derived from the PCB defect dataset released by the Intelligent Robotics Open Laboratory at Peking University. It encompasses various types of defects, such as shorts, open circuits, spurs, spurious copper, mouse bites, and missing holes. To mitigate the risk of network overfitting, we augmented the original 693 samples using techniques like random rotations, random cropping, brightness adjustments, and noise injection, resulting in a substantial increase to 5,814 samples. The distribution of different defect types is depicted in Fig. 4. We partitioned the expanded dataset into training, validation, and test sets in a ratio of 6:2:2, respectively.

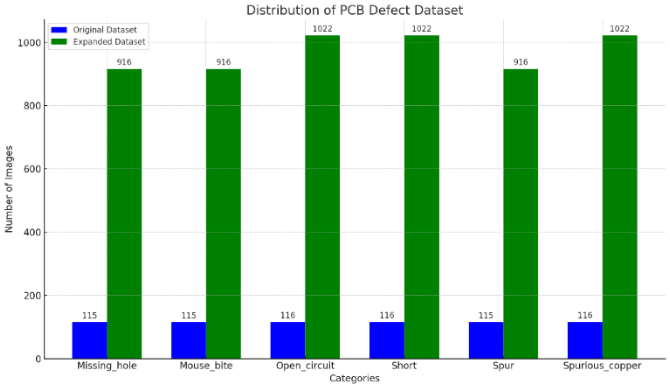

Fig. 4. Distribution of PCB defect dataset.

We conducted our training and evaluation on a high-performance Windows PC outfitted with an Intel Core i7-10400 CPU, an NVIDIA GeForce RTX 4080 GPU, and 32GB of RAM, ensuring efficient processing capabilities for deep learning tasks. Our software stack consisted of Python 3.8, leveraging libraries such as OpenCV for image processing and PyTorch for model development and training. The models were trained over 300 epochs with a batch size of 2, and we standardized the input image size to 640×640 pixels to maintain consistency in training and testing.

For model evaluation, we adopted two primary metrics: the mean average precision (mAP) and the detection speed, measured in frames per second (FPS). The mAP provides a comprehensive measure of model accuracy across all classes, factoring in both precision and recall, while FPS gauges the model's real-time performance capabilities. These benchmarks allowed us to assess the overall effectiveness and efficiency of our PCB defect detection models in a controlled and quantifiable manner.

B. Comparison with other Models

Table I provides a comparative analysis of various object detection models on the PCB defect dataset, showcasing their performance in terms of mean average precision (mAP), frames per second (FPS), and computational complexity measured in GFLOPs. The proposed model outperforms all other models with an exceptional mAP of 99.2%, indicating its superior accuracy in defect detection. Despite this high precision, it maintains a competitive detection speed of 51 FPS, balancing efficiency with effectiveness. Notably, the proposed model achieves this while having a lower computational cost (41.0 GFLOPs) than YOLOv5 and Faster R-CNN, which have higher GFLOPs of 100 and 170, respectively. The YOLOv7 and the Improved YOLOv5 models also exhibit high mAP scores, suggesting that the latest iterations and enhancements in the YOLO series continue to advance the state-of-the-art in object detection. However, the proposed model's edge in mAP suggests that the integration of novel architectural features or training strategies could be particularly beneficial for the specific challenges presented by PCB defect detection. The Transformer-YOLO and the Improved YOLOv5, while yielding high accuracy, do not report FPS, which leaves a gap in understanding their real-time applicability. On the other end of the spectrum, SSD demonstrates the lowest GFLOPs, indicating a very efficient model, but it lags in mAP, underscoring a trade-off between computational efficiency and detection accuracy. Overall, the results in Table I highlight the proposed model's capability to set a new benchmark for PCB defect detection by achieving a harmonious balance between accuracy, speed, and computational efficiency.

Fig. 5 presents a comprehensive visualization of the detection results achieved by the proposed PCB defect detection model. Across multiple instances, the model successfully identifies and localizes various types of PCB defects. Each type of defect is accurately marked with bounding boxes and labeled, indicating a high level of precision in the model's predictive capability. The clarity of the bounding boxes and the accuracy of the labels suggest that the model is well-tuned to the intricacies of PCB defect detection. The absence of mislabeling or missed detections in the provided visualization underscores the robustness of the model and its potential for practical applications in quality control and automated inspection systems within electronic manufacturing.

TABLE I. COMPARING THE PROPOSED MODEL WITH OTHER MODELS ON THE PCB DEFECT DATASET

Models	mAP (%)	FPS	GFLOPs
Faster R-CNN [22]	74.4	21	170
SSD [23]	82.2	42	2.5
YOLOv3 [24]	87.2	69	65
YOLOv5	91.4	102	100
Transformer-YOLO [25]	97.0	-	-
YOLOv7 [26]	97.8	54	51.2
Improved YOLOv5 [27]	97.9	-	53.5
Proposed Model	99.2	51	41.0

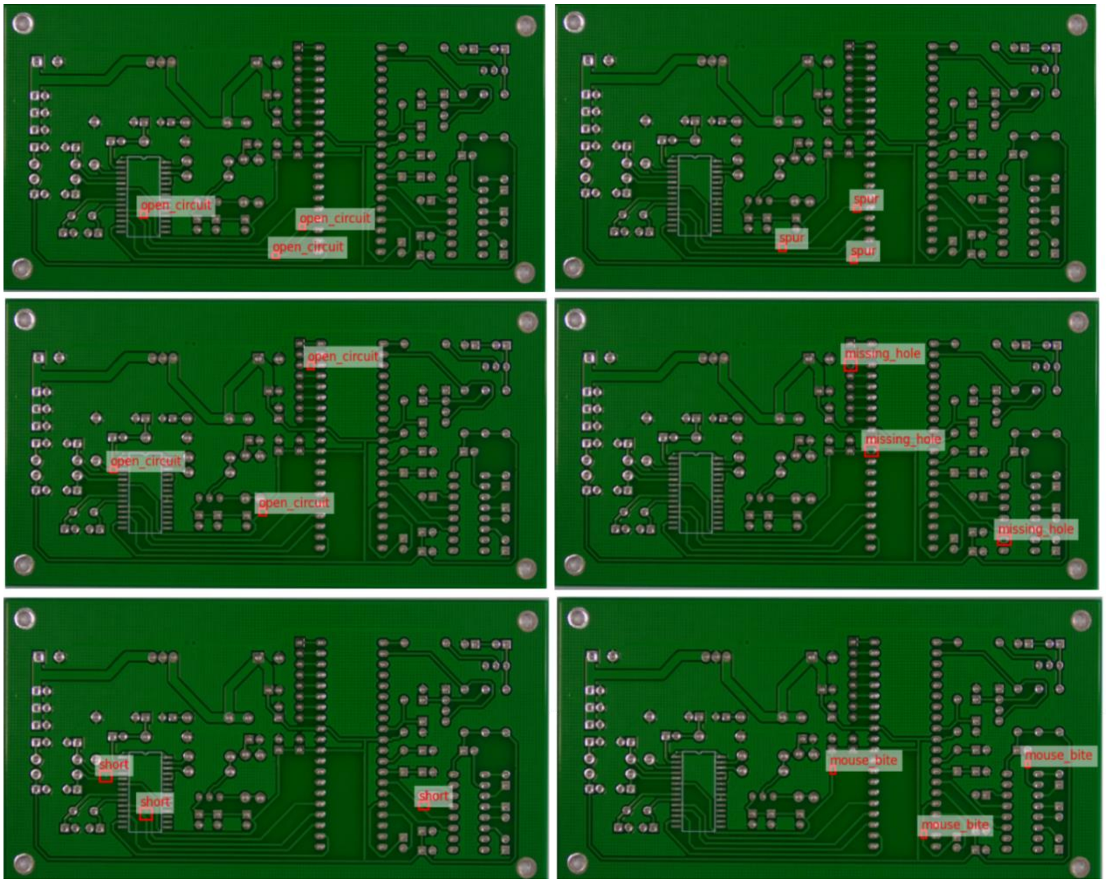

Fig. 5. Visualization of detection results of the proposed model.

C. Effect of Backbone with Self-Attention Mechanism

We also conduct experiments on the PCB defect validation set with various backbone architectures to evaluate the effectiveness of the proposed backbone with self-attention mechanism. Fig. 6 illustrates the performance trade-offs between mean average precision (mAP) and inference speed (FPS) for various backbone architectures on the validation set, including ResNet-50, ResNet-101 [15], EfficientNet [28], SENet-50 [29]. The proposed model achieves the highest mAP of 98.4% with a competitive FPS of 51, showcasing its superior defect detection accuracy without significantly compromising on speed. The ResNet-50 and ResNet-101 architectures offer a good balance between accuracy and speed, with ResNet-101 slightly trailing in FPS at 41 but offering near-top mAP performance at 98.2%. Notably, EfficientNet stands out with the highest FPS of 68, suggesting it is the fastest model; however, this speed comes at the cost of a lower mAP of 95.8%. SENet-50 has the lowest mAP of 94.4% and a modest FPS of 48, indicating it may not be the optimal choice for scenarios where high precision is critical. Overall, the proposed model's impressive mAP, coupled with a substantial FPS, positions it as a compelling choice for real-time PCB defect detection applications.

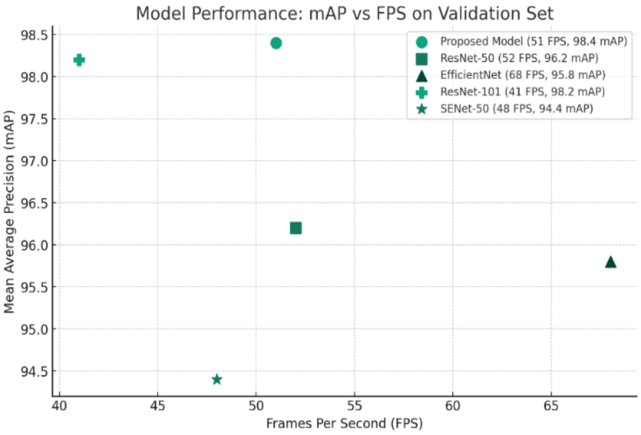

Fig. 6. The performance trade-offs between mean average precision (mAP) and inference speed (FPS) for various backbone architectures on the validation set.

V. Conclusions

In conclusion, this research paper proposes a novel approach to PCB defect detection, leveraging advanced hybrid neural network architecture. Our model integrates a ResNet and Bottleneck Transformer Backbone, a Ghost Convolution Neck, and Fully Convolutional One-stage detection Head, showing superior performance in identifying subtle and small-scale defects on PCBs. The comparative analysis highlights our model's exceptional mean average precision of 99.2%, significantly surpassing that of existing object detection models. Moreover, it achieves this high level of accuracy while maintaining a competitive detection speed of 51 FPS and requiring fewer computational resources compared to other high-performing models. The introduction of extensive augmentation techniques has further enhanced the dataset's diversity, improving the model's robustness and its ability to generalize across various PCB defect types. Future work will focus on optimizing the model to further improve its detection capabilities, particularly for the smallest and most challenging defects. Additionally, we will explore the potential for real-time processing in greater depth, aiming to extend the model's applicability to industrial settings and automated quality control systems. The success of this study marks a significant step forward in the field of automated defect detection, promising to enhance the reliability and efficiency of electronic manufacturing processes through the adoption of advanced neural network architectures.

References

[1] Gao, Chengchong, Fei Hao, Jiatong Song, Ruwen Chen, Fan Wang, and Benxue Liu. "Cylinder Liner Defect Detection and Classification based on Deep Learning." *International Journal of Advanced Computer Science and Applications* 13, no. 8 (2022).

[2] Gollapalli, Mohammed, Sheriff A. Kudos, Mustafa A. Alhamad, Abdullah A. Alshehri, Hamad S. Alyemni, Mustafa O. Alali, Rami M. Mohammad, Mohammad Aftab Alam Khan, Mamoun M. Abdulqader, and Khalid M. Aloup. "Machine Learning Models Towards Prediction of COVID and Non-COVID 19 Patients in the Hospital's Intensive Care Units (ICU)." *Mathematical Modelling of Engineering Problems* 9, no. 6 (2022).

[3] Carion, Nicolas, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end object detection with transformers." In *European conference on computer vision*, pp. 213-229. Cham: Springer International Publishing, 2020.

[4] Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

[5] Jeon, Mingu, Siyun Yoo, and Seong-Woo Kim. "A contactless PCBA defect detection method: Convolutional neural networks with thermographic images." *IEEE Transactions on Components, Packaging and Manufacturing Technology* 12, no. 3 (2022): 489-501.

[6] Tsang, Sik-Ho, Zhaoqing Suo, Tom Tak-Lam Chan, Huu-Thanh Nguyen, and Daniel Pak-Kong Lun. "PCB Soldering Defect Inspection Using Multitask Learning under Low Data Regimes." *Advanced Intelligent Systems* (2023): 2300364.

[7] Piliposyan, Gor, and Saqib Khursheed. "Computer vision for hardware trojan detection on a PCB using siamese neural network." In *2022 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pp. 1-7. IEEE, 2022.

[8] Yaohui, Kang, Gao Yuhang, and Luo Cheng. "Visual alignment system for PCB production based on yolov5." In *2022 34th Chinese Control and Decision Conference (CCDC)*, pp. 445-449. IEEE, 2022.

[9] Yao, Naifu, Yongqiang Zhao, Seong G. Kong, and Yang Guo. "PCB defect detection with self-supervised learning of local image patches." *Measurement* 222 (2023): 113611.

[10] Jiang, Wujin, Taifu Li, Shaolin Zhang, Wenbin Chen, and Jie Yang. "PCB defects target detection combining multi-scale and attention mechanism." *Engineering Applications of Artificial Intelligence* 123 (2023): 106359.

[11] Lim, JiaYou, JunYi Lim, Vishnu Monn Baskaran, and Xin Wang. "A deep context learning based PCB defect detection model with anomalous trend alarming system." *Results in Engineering* 17 (2023): 100968.

[12] Zhang, Huan, Liangxiao Jiang, and Chaoqun Li. "CS-ResNet: Cost-sensitive residual convolutional neural network for PCB cosmetic defect detection." *Expert Systems with Applications* 185 (2021): 115673.

[13] Chen, Boyuan, and Zichen Dang. "Fast PCB defect detection method based on FasterNet backbone network and CBAM attention mechanism integrated with feature fusion module in improved YOLOv7." *IEEE Access* (2023).

[14] Liu, Jinhai, Hengguang Li, Fengyuan Zuo, Zhen Zhao, and Senxiang Lu. "KD-LightNet: A Lightweight Network Based on Knowledge Distillation for Industrial Defect Detection." *IEEE Transactions on Instrumentation and Measurement* (2023).

[15] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[16] Srinivas, Aravind, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. "Bottleneck transformers for visual recognition." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16519-16529. 2021.

[17] Liu, Ze, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. "Swin transformer: Hierarchical vision transformer using shifted windows." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012-10022. 2021.

[18] Han, Kai, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. "Ghostnet: More features from cheap operations." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1580-1589. 2020.

[19] Tian, Zhi, Chunhua Shen, Hao Chen, and Tong He. "Fcos: Fully convolutional one-stage object detection." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627-9636. 2019.

[20] Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal loss for dense object detection." In *Proceedings of the IEEE international conference on computer vision*, pp. 2980-2988. 2017.

[21] Tong, Zanjia, Yuhang Chen, Zewei Xu, and Rong Yu. "Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism." *arXiv preprint arXiv:2301.10051* (2023).

[22] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

[23] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21-37. Springer International Publishing, 2016.

[24] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[25] Chen, Wei, Zhongtian Huang, Qian Mu, and Yi Sun. "PCB Defect Detection Method Based on Transformer-YOLO." *IEEE Access* 10 (2022): 129480-129489.

[26] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7464-7475. 2023.

[27] Chen, Shiqiao, Xiqing Liang, and Wenneng Jiang. "PCB Defect Detection Based on Image Processing and Improved YOLOv5." In *Journal of Physics: Conference Series*, vol. 2562, no. 1, p. 012002. IOP Publishing, 2023.

[28] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In *International conference on machine learning*, pp. 6105-6114. PMLR, 2019.

[29] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132-7141. 2018.