Design of Deep Learning Techniques for PCBs Defect Detecting System based on YOLOv10

Sumarin Ruengrote
Engineering Technology, Graduate School, Thai-Nichi Institute of Technology, Thailand
ru.sumarin_st@tni.ac.th

Kittikun Kasetravetin
Engineering Director, Madesco Intelligence Co. Ltd., Thailand
kittikun@madesco-int.com

Phanuphop Srisom
Robotics and Lean Automation Engineering, Faculty of Engineering, Thai-Nichi Institute of Technology, Thailand
phanuphop.sri@tni.ac.th

Theeraphan Sukchok
Analytic Solution, Tspacedigital Company, Thailand
theeraphan.s@tspacedigital.com

Don Keawdook
Robotics and Lean Automation Engineering, Faculty of Engineering, Thai-Nichi Institute of Technology, Thailand
don@tni.ac.th (corresponding author)

Received: 18 September 2024 | Revised: 14 October 2024 | Accepted: 21 October 2024

 DOI: https://doi.org/10.48084/etasr.9028

ABSTRACT

As Printed Circuit Boards (PCBs) are critical components in electronic products, their quality inspection is crucial. This study focuses on quality inspection to detect PCB defects using deep learning techniques. Traditional widely used quality control methods are time-consuming, labor-intensive, and prone to human errors, making the manufacturing process inefficient. This study proposes a deep-learning approach using YOLOv10. Through the incorporation of architectural improvements such as CSPNet and PANet that improve feature extraction and fusion, as well as a dual assignments mechanism that increases localization accuracy, YOLOv10 offers significant improvements over earlier versions, such as YOLOv5 and YOLOv8, and Faster R-CNN models. These innovations allow YOLOv10 to deliver superior performance in terms of both speed and precision. The experiments used a custom dataset consisting of 1,260 PCB samples collected from the industry. The dataset was partitioned into 80% for model training and 20% for testing. The model was trained for 100 epochs with a batch size of 32 to evaluate its performance in identifying various PCB defects. YOLOv10, with its optimized architecture, fully utilized its capabilities while requiring less computational power than YOLOv5 and YOLOv8, especially in resource-constrained environments. Despite resource constraints, YOLOv10 achieved high accuracy, with a precision of at least 96% and a recall of 97%, surpassing earlier YOLO models and Faster R-CNN. It also achieved 99% mAP and more than 96% F1 score. These improvements in speed and accuracy make YOLOv10 a highly efficient solution for automated PCB inspection, reducing manual effort and offering fast and accurate classification adaptable to various applications.

Keywords-Printed Circuit Boards (PCBs); quality inspection; YOLOv10; deep learning

I. INTRODUCTION

Printed Circuit Boards (PCBs) comprise multiple layers of substrates embedded with electrical circuits to enable efficient interconnection of electronic components. The integrity of a PCB is critical to the overall performance of electronic products, making the detection and identification of defects such as misaligned, damaged, reversed, or missing parts an essential process. [1-2]

Faster R-CNN's great accuracy and capacity to handle intricate details have made it a popular choice for PCB defect detection. By splitting item detection into two phases, region proposal generation and additional analysis, its architecture makes accurate detection possible even in difficult-to-reach contexts. However, Faster R-CNN is less suitable for real-time applications, since its two-step procedure may restrict its speed. On the other hand, You Only Look Once (YOLO) provides real-time object detection in a single image pass without the need for region suggestions. Although YOLO had accuracy problems at first, especially when dealing with many or different-sized objects, later versions, such as YOLOv2 and YOLOv3, greatly increased detection accuracy by implementing improvements such as batch normalization, Feature Pyramid Networks (FPN), and anchor boxes. These revisions brought improved multi-scale object identification and accuracy but at a slightly slower pace. By combining cutting-edge methods such as Bag of Freebies (BoF) and Bag of Specials (BoS) with Cross Stage Partial Networks (CSPNet), YOLOv4 improved speed and accuracy even further. The model's complexity increased training time and resource needs despite these advances. The release of YOLOv5 in 2020 was well-liked by developers due to its emphasis on simplicity of use, lightweight deployment, and increased speed. With advances such as the Extended Efficient Layer Aggregation Networks (E-ELAN) to improve complex object detection, later versions such as YOLOv6 and YOLOv7 continued to optimize the trade-off between speed and accuracy. However, when it came to identifying small or complex objects, YOLOv7 was still limited. YOLOv8 includes transformer layers to further improve detection in intricate scenes, but the larger model size requires greater processing power [3-5].

YOLOv10, launched in 2024, leverages CSPNet, Feature Pyramid Networks (FPN), and Path Aggregation Networks (PANet) to facilitate effective multi-scale object detection, boosting speed and accuracy over earlier iterations. To provide better real-time detection performance, it also combined sophisticated post-processing techniques with transformers. This makes it ideal for industrial applications that need both speed and precision. YOLO is a widely popular deep-learning technique for defect detection, particularly in PCB inspection, due to its accuracy and processing speed. YOLO has continuously developed from YOLOv1 in 2016 to YOLOv10 in 2024, focusing on improving object detection performance in complex environments and supporting devices with limited resources. This study selected YOLOv10 [6] to detect PCB defections.

Many artificial intelligence techniques have been studied in electronics manufacturing [7-11]. In [12], a comparison of YOLOv3 and the Faster Region-based CNN showed that the defect detection model based on YOLOv3 identified defects with 95% accuracy. Thus, the Incoming Quality Control (IQC) random sampling inspection could be replaced by a full inspection, and the Surface Mount Technology (SMT) full inspection stations could be eliminated to reduce the need for inspection personnel. In [13], YOLOv5l and YOLOv8l were compared in tomato disease detection, showing that YOLOv8l was slightly more accurate than YOLOv5l. In [14], a YOLO-based deep learning algorithm was proposed to evaluate the quality of PCBs. This study recorded and labeled a dataset of 11,000 images of defective PCBs, and the proposed model achieved a defect detection accuracy of 98.79% for a batch size of 32. In [15], transfer learning with six pre-trained CNN models (DenseNet121, DenseNet169, MobileNetV2, ResNet50V2, VGG16, and VGG19) was used for automatic cotton plant disease detection, where DenseNet169 and ResNet50V2 achieved the highest accuracy at 96%, while MobileNetV2 had the lowest accuracy at 52%. In [16], an uncoupled-state multimodel approach was presented to identify nonlinear PCB soldering systems, achieving high accuracy (MSE=0.0727, VAF=99.998%), and demonstrating its efficiency in identifying and modeling complex systems into manageable linear subsystems. In [17], a semi-supervised learning model was developed for PCB defect detection, trained using both labeled and unlabeled data with data augmentation techniques. The semi-supervised model demonstrated higher accuracy and robustness to mislabeled data compared to a fully supervised model, with an error rate increase of less than 0.5% even when up to 9% of the data was incorrectly labeled. In [18], the LW-YOLO model, a lightweight deep learning model, was employed for defect detection in PCBs, using a bidirectional feature pyramid network, partial convolution module, and minimum point distance intersection over union loss function. This model achieved 96.4% mAP, 97.1% precision, and a detection speed of 141.5 fps, outperforming YOLOv8 in accuracy and speed. In [19], YOLO-MBBi, an improvement over YOLOv5 for PCB surface defect detection, integrated MBConv, CBAM attention, BiFPN, and depth-wise convolution, replacing the CIoU loss function with SIoU. Experimental results on two public datasets showed that YOLO-MBBi achieved a mAP50 of 95.3% and a recall of 94.6%, surpassing YOLOv5s and outperforming YOLOv7 with lower FLOPs and 48.9 fps. This makes YOLO-MBBi well-suited for industrial production needs.

In [20], an image quality assessment model was developed, using lightweight deep learning (EfficientNetV2) and implementing it on FPGA hardware for real-time processing. The model demonstrated high accuracy and efficiency, which makes it suitable for mobile devices such as smartphones and smart cameras. In [21], a comprehensive review and comparison of YOLOv5, YOLOv8, and YOLOv10 was presented. This review provides insights into the trade-offs between model complexity and detection accuracy, providing guidance for selecting the most appropriate YOLO version for specific edge computing applications. In [22], SolDef_AI, an open dataset was presented for defect identification in the soldering process of PCBs. To create a thorough database containing component location and solder volume that captures

flaws occurring during the soldering process, this study collected 1,150 photos of SMT component soldering from three different viewpoints. Testing with Mask R-CNN showed that the SolDef_AI dataset can be used to improve PCB defect detection systems in soldering operations, thus increasing the accuracy of defect inspections. In [23], an end-to-end deep learning system was proposed for fault detection and classification in PCB production. This study presented a single-stage object detection model that combined transformers and CNNs. Using images of PCBs with six fault types, including missing holes and mouse bites, the model was tested on the HRIPCB dataset and its performance was evaluated against other models including Faster R-CNN and YOLOv5. With low-resolution photos, the proposed model performed well with 98.1% mAP. Furthermore, the model was perfect for defect detection in manufacturing processes that require high speed and accuracy, since it contained almost three times fewer parameters than other models.

This study examined distinct variants of the YOLOv10 model. To improve the accuracy and efficiency in detecting five types of PCB defects, each version was fine-tuned with distinct parameters. This approach involved gradually adjusting the models by increasing the number of parameters to enhance their ability to identify defects across a variety of defect categories.

II. METHOD

A. Dataset

The PCB defect detection dataset utilized PCBs from a stepper motor driver type A4988 DRV8825 model SKR V1.3 1.4GTR V1.0 of a 3D printer machine. The dataset contains two types of PCB images, as shown in Figure 1:

- Fully functional PCBs: PCB images that lack any defect indications.
- Defective PCBs: PCBs that have imperfections.

Approximately 1,260 PCB images were collected, consisting of 590 images of intact boards and 670 of defective boards. The defective images were divided into five classes based on the defect type to ensure balanced representation. This study focuses only on defective PCB images, which were labeled using Roboflow and then preprocessed in Google Colab.

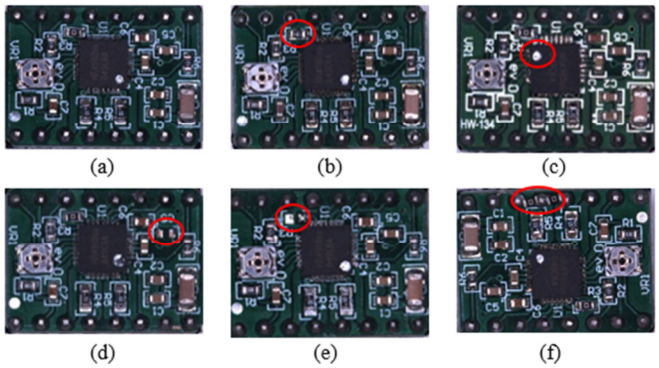

Fig. 1. PCBs: (a) Normal PCBs, (b) wrong component, (c) misdirection, (d) component crack, (e) missing component, (f) abnormal.

B. Preparing PCBs Before Taking Sample Image

Several critical stages are required to ensure that the images acquired are as precise and accurate as possible when preparing a PCB for image capture for a testing program. PCBs must be clean and clear of impurities to obtain the highest level of accuracy in sample imaging. In this procedure, ultrasonic cleaning is essential [24]. This study used an ultrasonic cleaner that employs high-frequency sound waves to eliminate debris and residues to prepare the boards for imaging and testing [25].

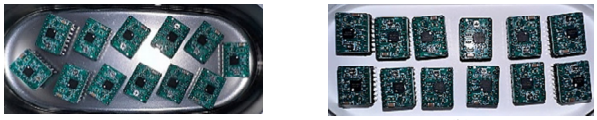

Fig. 2. PCB cleaning: (a) Ultrasonic cleaning (b) Air blow after takeout of ultrasonic machine.

C. Image Acquisition

Installation scenarios were simulated to ensure representative sample images for testing. This method enhances the accuracy of testing and system evaluation by closely mimicking real-world conditions, allowing for verification and improvements before actual deployment [26].

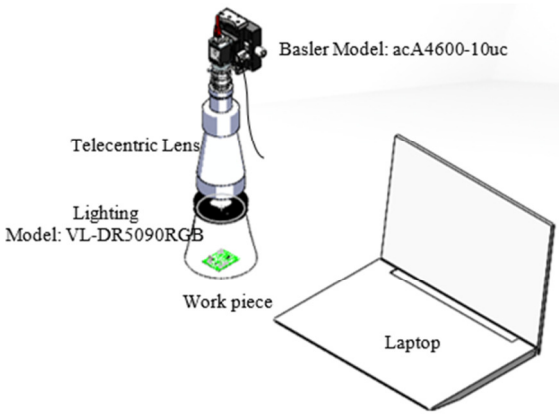

Fig. 3. Image acquisition setup.

This study focused on the significance of correctly establishing a vision system to improve the precision of fault identification throughout the manufacturing process. The Basler model acA4600-10uc camera was used, with a frame rate of 10 Hz and an exposure time of 12,005 ms. It was paired with the VL-DR5090RGB white ring light, which ensures constant and clear lighting. In addition, a Telecentric lens was used to precisely focus on nearby 3D objects. Its characteristics minimize picture distortion, enabling an accurate and reliable examination of item positions. The whole system was powered by the reliable PD3 series (PD3-10024-8-PI) digital power supply, ensuring uninterrupted and seamless operation of the equipment.

D. Data Augmentation

Image augmentation techniques were used to create altered copies of photos, without changing the labels, to artificially increase the size and variety of the image dataset.

- Blurring was applied by setting the kernel size range through the variable *blur_limit* = (3, 7), which randomly selects a kernel size between 3×3 and 7×7. The larger the kernel size, the blurrier the image becomes. Additionally, *blur*(p = 0.01) sets the probability of an image being blurred, where p = 0.01 indicates that each image has a 1% chance of being randomly selected for blurring. This technique enhances the dataset diversity by making some images blurred, helping the model to better learn how to handle unclear or blurred images.
- The MedianBlur method was applied to blur an image by adjusting the target pixel according to the median value of its surrounding pixels. There is a 1% chance that blur will be applied each time the technique is used (p=0.01), and the kernel size is randomly selected between 3×3 and 7×7 (*blur_limit* = (3,7)). This process involves examining the pixels around the target pixel within the chosen kernel size and replacing the target pixel with the median value of the surrounding pixels. This method effectively reduces noise in the image without significantly compromising important edges or details.
- The ToGray method converts color images (RGB) into grayscale images. In each processing instance, there is only a 1% chance (p = 0.01) that the image will be converted to grayscale. Despite this conversion, the image still retains 3 output channels (*num_output_channels*=3) to conform to the RGB standard required by most models. This method employs a weighted average approach, which ensures that the contribution of each color channel (red, green, and blue) is captured accurately. This allows the grayscale image to maintain important data while remaining in an RGB-compatible format, making it suitable for deep learning models that require RGB images for processing.
- The Contrast Limited Adaptive Histogram Equalization (CLAHE) technique was used, which enhances image contrast by dividing the image into smaller sections to adjust the contrast individually for each part. Setting p = 0.01 indicates a 1% chance that CLAHE will be applied, meaning that the adjustment will only occur in a small portion of the image. The *clip_limit* = (1, 4.0) defines the threshold for contrast enhancement to prevent overamplification of light, thereby minimizing the introduction of noise. A higher clip limit allows for greater contrast enhancement. For the tile*_grid_size* = (8, 8) setting, the image is divided into an 8×8 grid, and CLAHE adjusts the contrast separately for each section, ensuring that the contrast improvement is precise and suitable for each part of the image.

E. YOLOv10 Description

YOLOv10, debuted in 2024, is the most recent development in the YOLO series and is intended to greatly increase object identification accuracy and speed in challenging conditions. Several cutting-edge methods, such as PANet and CSPNet, which improve feature extraction and fusion while preserving efficiency, are integrated into the model. YOLOv10 is perfect for resource-constrained applications, such as embedded devices and IoT systems, without sacrificing detection accuracy, since it uses parallel processing and optimized learning models to provide great performance with a smaller model size.

YOLOv10, which builds on the achievements of previous iterations such as YOLOv8 and YOLOv5, utilizes a dual assignment mechanism to improve localization accuracy and shorten inference time. It also adds NMS-free training, which eliminates the necessity for conventional non-maximum suppression during training. Furthermore, robust performance across a range of object scales is ensured by the use of spatial-channel decoupled downsampling, which reduces processing costs while maintaining detection precision. In addition, it uses a partial self-attention module to enhance the recognition of complicated objects and large-kernel convolutions to gather more contextual information for larger items. Due to these advances, YOLOv10 performs more accurately and efficiently than its predecessors, making it an effective tool for real-time object detection in difficult conditions [27].

This study used YOLOv10 to create a model that can identify particular PCB faults. The first step involved marking the PCBs' problem locations, which were subsequently utilized to set up the detecting algorithm. Five categories of flaws were identified and categorized in all collected samples. Figure 4 shows the detection concept.

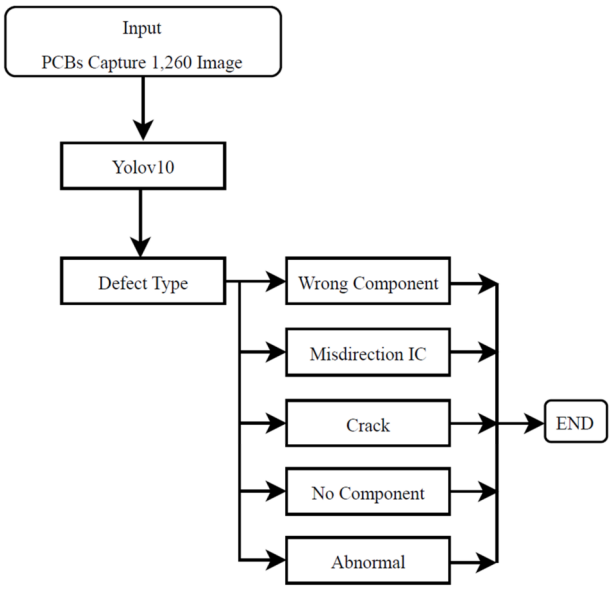

Fig. 4. Detection concept.

III. EXPERIMENTAL RESULTS ANALYSIS

A. Evaluation Metrics

Performance metrics are essential for evaluating the accuracy and efficiency of object detection models, providing

insights into how effectively a model identifies and localizes objects within images. They also clarify the model's ability to manage false positives and false negatives, providing critical information to assess and improve overall performance [28, 29].

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (2)$$

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \quad (3)$$

where TP represents True Positives, TN represents True Negatives, FP represents False Positives, and FN represents False Negatives. The F1 score is a more effective method to evaluate performance on imbalanced data, being the harmonic average of the precision and recall rates [30].

$$\text{F1 score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}} \quad (4)$$

B. Model Training

After labeling and preprocessing the dataset, it was split into an 80:20 ratio for training and testing. Training and testing were carried out on the experimental environment detailed in Table I, using the training parameters listed in Table II [31-32].

TABLE I. EXPERIMENTAL ENVIRONMENT

Category	Configuration
CPU	Intel(R) Core (TM) Ultra 9 185H
GPU	NVIDIA A100 GPU
Operation system	Window 11
Framework	Python3
Programming environment	Google Colab

TABLE II. TRAINING PARAMETERS

Parameter	Value
Image Size	640×640
Batch Size	32
Epochs	100

C. Performance Evaluation and Class-Wise Validation Performance

YOLOv10 was compared with YOLOv5, YOLOv8, and Faster R-CNN in finding defects in PCBs. Five different kinds of flaws were evaluated under identical conditions. Table III shows the general performance and the individual flaw detection capability of each model. Table IV shows comprehensive performance measures for every fault type. The experiments were carried out on an NVIDIA A100 GPU on Google Colab. The A100 GPU's main benefit is its ability to handle tensor cores, facilitating faster model training and testing. The A100 can effectively manage vast datasets with its 40-80 GB VRAM. It also offers mixed precision computation, accelerating processing without sacrificing result quality, and uses Multi-Instance GPU (MIG) technology to let the GPU be split into several instances for more flexibility, making it ideal for sophisticated data science jobs and high-performance computing [33-35].

TABLE III. PERFORMANCE COMPARISON OF DEEP LEARNING MODELS

Model	Precision	Recall	mAP50	F1-score
Faster R-CNN	0.39	0.75	0.61	0.51
YOLOv5	0.99	0.99	0.99	0.86
YOLOv8	0.99	0.99	0.99	0.86
YOLOv10	0.98	0.98	0.99	0.86

The results show that YOLOv5, YOLOv8, and YOLOv10 significantly outperformed Faster R-CNN. With mAP50 scores of 0.99, all three models attained high precision and recall values of 0.98-0.99, proving their accuracy and near-perfect flaw detection capacity. Despite a recall of 0.75, Faster R-CNN displayed much lower precision (0.39) and mAP50 (0.61), indicating a higher error rate in defect detection. Reflecting a well-balanced trade-off between precision and recall, the F1 scores of YOLOv5, YOLOv8, and YOLOv10 were 0.86. Among them, YOLOv10 performed closely to YOLOv5 and YOLOv8 across all measures. Furthermore, unlike YOLOv5, YOLOv8, and Faster R-CNN, which demand more resources, YOLOv10 has the advantage of running in resource-constrained environments. This makes YOLOv10 a perfect fit for situations with hardware restrictions, particularly when processing on less capable GPUs, especially while still producing results equivalent to models that demand more computational resources.

TABLE IV. CLASS-WISE PERFORMANCE OF DEEP LEARNING MODELS

Defect	Model	Precision	Recall	mAP50	F1
Abnormal	Faster R-CNN	0.75	0.82	0.91	0.78
	YOLOv5	0.99	1	0.99	0.86
	YOLOv8	0.99	1	0.99	0.86
	YOLOv10	0.99	0.99	0.99	0.85
Misdirection IC	Faster R-CNN	0.89	1	1.00	0.94
	YOLOv5	0.99	1	0.99	0.96
	YOLOv8	0.98	1	0.99	0.96
	YOLOv10	0.97	1	0.99	0.96
No component	Faster R-CNN	0.93	1	1	0.96
	YOLOv5	0.99	1	0.99	0.85
	YOLOv8	0.99	1	0.99	0.85
	YOLOv10	0.98	0.98	0.99	0.85
Crack	Faster R-CNN	1	0.95	0.85	0.97
	YOLOv5	0.99	0.99	0.99	0.82
	YOLOv8	0.99	0.99	0.99	0.82
	YOLOv10	0.97	0.96	0.99	0.81
Wrong Component	Faster R-CNN	1	0.96	0.79	0.96
	YOLOv5	0.99	1	0.99	0.8
	YOLOv8	0.99	1	0.99	0.8
	YOLOv10	0.99	0.99	0.99	0.8

YOLOv5, YOLOv8, and YOLOv10 consistently showed high precision and recall across all defect types. Their mAP50 reached 0.99, suggesting an almost flawless detection accuracy. Their F1 scores, which ranged from 0.81 to 0.97, demonstrated a reasonable mix between recall and precision. Though their performance is outstanding, YOLOv5 and YOLOv8 have restrictions in resource-limited contexts because of their more

complicated designs and greater model sizes, which require more GPU memory and processing capability. YOLOv10 effectively solves these restrictions. YOLOv10 needs only 16-24 GB of GPU RAM even in its more sophisticated forms, far less than Faster R-CNN's 30.5 GB, YOLOv5's 30.4 GB, and YOLOv8's 25.2 GB. Table IV shows that although it used far fewer resources, YOLOv10 performed exactly as YOLOv5 and YOLOv8. This makes YOLOv10 perfect for pragmatic uses in settings with limited resources since it provides exceptional accuracy with fewer resources.

D. Confusion Matrices

The confusion matrices for each subversion of the YOLOv10 model highlight the accuracy and errors of YOLOv10 in classifying data by displaying the number of correct and incorrect predictions for each class. These graphs reflect the model's performance across various categories and pinpoint the areas where errors occur, helping to identify where YOLOv10 might face challenges in detecting certain types of defects. Moreover, the graphs aid in analyzing how the model handles imbalanced data, emphasizing its ability to maintain accuracy even in difficult scenarios. This is particularly crucial for detecting defects on PCBs, where some types of defects may appear less frequently. Figures 5 through 9 demonstrate YOLOv10's performance in detecting defects across five experimental cases, showcasing the model's results and revealing areas that may benefit from further optimization.

By predicting in the background, the YOLO model improves its focus on actual objects, reducing detection errors. Background predictions are necessary to display False Positives (FP) and True Negatives (TN), aiding in assessing the model's accuracy in distinguishing objects from the background. This comprehensive evaluation allows accurate calculations of metrics such as accuracy, precision, recall, and F1 score, reducing FPs caused by background confusion and increasing the model's robustness against irrelevant data.

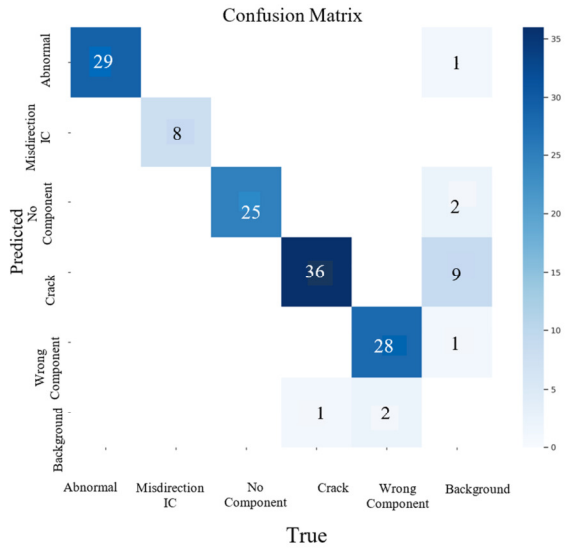

Fig. 5. Confusion matrix from YOLOv10-n.

YOLOv10-n demonstrated high accuracy across several categories, including Abnormal, No Component, Crack, and Wrong Component. However, some misclassifications were observed, such as the confusion between No Component and Crack, possibly due to similarities in the characteristics of these defects. This confusion led the model to incorrectly classify these defects as background. The model's prediction of Background occurs when it struggles to differentiate between defects or when the background in the image shares similar features with certain defects, such as Abnormal or Wrong Component. This similarity causes the model to misclassify these defects as background instead of their actual categories. Frequent Background predictions often happen when the background in the images has textures or patterns similar to the actual defects, indicating that the model still faces challenges in distinguishing complex image features.

The confusion matrix of YOLOv10-s demonstrates the model's classification performance. Diagonal cells represent correct predictions. In contrast, the off-diagonal cells indicate incorrect predictions, such as misclassifying No Component as Wrong Component in 3 instances. The model shows confusion in the Background class, with several misclassifications, such as predicting Background instead of Abnormal in 1 instance and misclassifying Crack as Background in 2 instances. These errors suggest challenges in differentiating the Background class, probably due to similarities between background elements and other categories. This analysis highlights key areas where the model struggles, particularly with background-related misclassifications, providing valuable insights for improving model accuracy in the future.

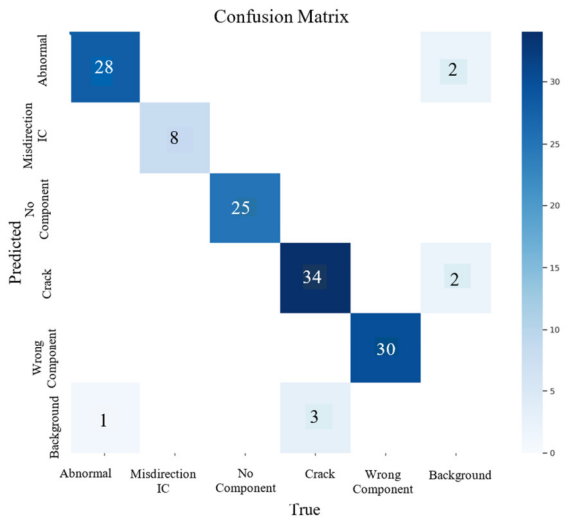

Fig. 6. Confusion matrix from YOLOv10-s.

Figure 8 shows the confusion matrix of YOLOv10-m. Off-diagonal cells show incorrect predictions, such as misclassifying Background as Abnormal once and as Crack once, highlighting noticeable confusion in the Background category. These errors suggest that the Background category shares certain similarities with other categories, making it harder for the model to distinguish. Although the model

performed well in categories such as Crack and Abnormal, the confusion in the Background category clearly requires improvement to enhance the model's accuracy.

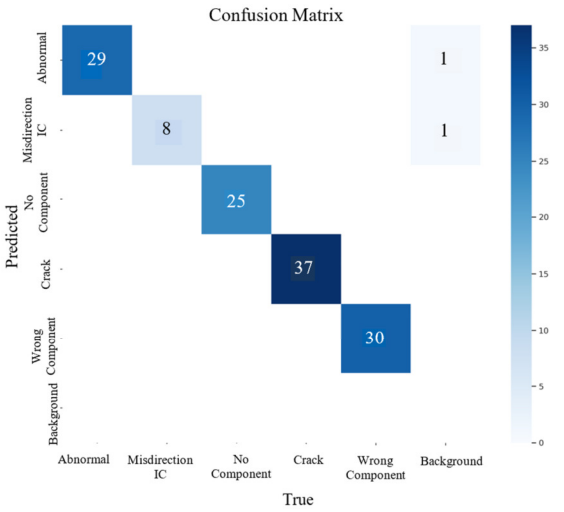

Fig. 7. Confusion matrix from YOLOv10-m.

Figure 8 shows the confusion matrix of YOLOv10-l. The off-diagonal cells indicate misclassifications, such as predicting Background as Abnormal once and Background as crack once. The confusion in the background category is a critical area of concern, as the model struggles to differentiate this class. While the model performs well in certain categories such as Abnormal and Crack, the misclassification in the Background class indicates a weakness that needs improvement to enhance the model's accuracy in future predictions.

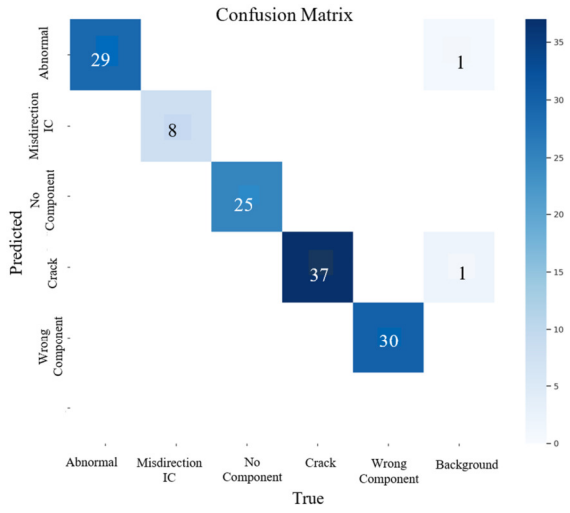

Fig. 8. Confusion Matrix from YOLOv10-l.

Figure 9 shows the confusion matrix of YOLOv10-xl. The off-diagonal cells indicate misclassifications, such as predicting Background as Abnormal twice and Background as Wrong Component once.

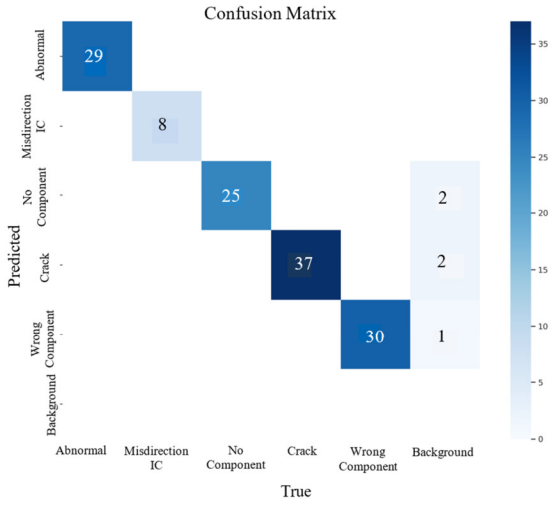

Fig. 9. Confusion matrix from YOLOv10-xl.

The confusion in the Background class is significant, as the model struggles to differentiate this category from others, possibly due to similarities between Background and classes such as Abnormal and Wrong Component. The color shading in the graph provides a clear view of the prediction frequencies, with darker shades indicating higher correct predictions and lighter shades representing errors. Although the model performs well in certain categories, such as Abnormal and Crack, the confusion in the Background category remains a critical weakness. Addressing this issue is crucial to improving the model's accuracy in future classifications.

Predictions for the Background category, as shown in the confusion matrix, suggest that the model can differentiate objects from empty space. Reducing detection errors requires this skill. The graph also shows the performance of the model over several classes, where for most of them it has great accuracy. However, there are potential misclassifications in circumstances where some groups exhibit similar traits.

IV. CONCLUSION

This study demonstrated the effectiveness of YOLOv10 in detecting defects on PCBs. PCB integrity is vital for product performance, and accurate, real-time defect detection is essential. YOLOv10 offers significant improvements over previous versions, including YOLOv5 and YOLOv8, as it incorporates advanced techniques such as CSPNet, PANet, and dual assignments that enhance both speed and accuracy, making it particularly suitable for real-time applications. YOLOv10 surpasses both its predecessors by offering a better balance between speed, accuracy, and computational efficiency. The model integrates NMS-free training and spatial-channel decoupled downsampling, reducing the need for computational power while maintaining high detection precision. YOLOv10's partial self-attention module also improves the detection of complex objects by focusing more effectively on key areas of an image. This study used a dataset of 1,260 PCB samples, with 80% allocated for training and 20% for testing. YOLOv10 showed superior performance, with precision reaching 99% and recall above 97%, achieving the

highest mAP and F1 scores among all variants tested. In contrast, YOLOv5 and YOLOv8 require more resources, particularly in constrained environments. YOLOv10's enhancements in computational efficiency enable it to deliver similar accuracy under lower-spec hardware, making it crucial for real-world manufacturing scenarios where high-end GPUs are often unavailable. Although this study offers a comprehensive approach to PCB defect detection, it may not account for all potential variations in actual production settings. Future research should focus on expanding the dataset with diverse PCB types and defect patterns to enhance the model robustness for broader applications.

ACKNOWLEDGMENT

The authors are grateful for the support and guidance provided by the Advanced Design and Manufacturing Technology (ADMT), Faculty of Engineering, Thai-Nichi Institute of Technology, and Madesco Intelligent Co., Ltd. for this research.

REFERENCES

[1] V. Nandini, R. D. Vishal, and C. A. P. and S. Aishwarya, "A Review on Applications of Machine Vision Systems in Industries," *Indian Journal of Science and Technology*, vol. 9, no. 48, pp. 1–5, May 2016, https://doi.org/10.17485/ijst/2016/v9i48/108433.

[2] N. Petkov and M. Ivanova, "Printed circuit board and printed circuit board assembly methods for testing and visual inspection: a review," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 4, pp. 2566–2585, Aug. 2024, https://doi.org/10.11591/eei.v13i4.7601.

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

[4] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448, https://doi.org/10.1109/ICCV.2015.169.

[5] D. Maneetham, The Guide to Machine Vision Control with LabVIEW and Vision Builder. SE Education, 2021.

[6] Ultralytics, "YOLOv10." https://docs.ultralytics.com/models/yolov10.

[7] H. Xu, L. Wang, and F. Chen, "Advancements in Electric Vehicle PCB Inspection: Application of Multi-Scale CBAM, Partial Convolution, and NWD Loss in YOLOv5," *World Electric Vehicle Journal*, vol. 15, no. 1, Jan. 2024, Art. no. 15, https://doi.org/10.3390/wevj15010015.

[8] M. A. Mallaiyan Sathiaseelan, O. P. Paradis, S. Taheri, and N. Asadizanjani, "Why Is Deep Learning Challenging for Printed Circuit Board (PCB) Component Recognition and How Can We Address It?," *Cryptography*, vol. 5, no. 1, Mar. 2021, Art. no. 9, https://doi.org/10.3390/cryptography5010009.

[9] T. Benbarrad, M. Salhaoui, S. B. Kenitar, and M. Arioua, "Intelligent Machine Vision Model for Defective Product Inspection Based on Machine Learning," *Journal of Sensor and Actuator Networks*, vol. 10, no. 1, Jan. 2021, Art. no. 7, https://doi.org/10.3390/jsan10010007.

[10] Q. Ling and N. A. M. Isa, "Printed Circuit Board Defect Detection Methods Based on Image Processing, Machine Learning and Deep Learning: A Survey," *IEEE Access*, vol. 11, pp. 15921–15944, 2023, https://doi.org/10.1109/ACCESS.2023.3245093.

[11] D. B. Anitha and M. Rao, "SMT Component Inspection in PCBA's using Image Processing Techniques," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 541–547, Oct. 2019, https://doi.org/10.35940/ijitee.L3422.1081219.

[12] C. Y. Huang and P. X. Tsai, "Applying Machine Learning to Construct a Printed Circuit Board Gold Finger Defect Detection System," *Electronics*, vol. 13, no. 6, Mar. 2024, Art. no. 1090, https://doi.org/10.3390/electronics13061090.

[13] R. Ahmed and E. H. Abd-Elkawy, "Improved Tomato Disease Detection with YOLOv5 and YOLOv8," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 13922–13928, Jun. 2024, https://doi.org/10.48084/etasr.7262.

[14] V. A. Adibhatla, H. C. Chih, C. C. Hsu, J. Cheng, M. F. Abbod, and J. S. Shieh, "Defect Detection in Printed Circuit Boards Using You-Only-Look-Once Convolutional Neural Networks," *Electronics*, vol. 9, no. 9, Sep. 2020, Art. no. 1547, https://doi.org/10.3390/electronics9091547.

[15] S. Anwar, S. R. Soomro, S. K. Baloch, A. A. Patoli, and A. R. Kolachi, "Performance Analysis of Deep Transfer Learning Models for the Automated Detection of Cotton Plant Diseases," *Engineering, Technology & Applied Science Research*, vol. 13, no. 5, pp. 11561–11567, Oct. 2023, https://doi.org/10.48084/etasr.6187.

[16] S. Khouni and K. E. Hemsas, "Nonlinear System Identification using Uncoupled State Multi-model Approach: Application to the PCB Soldering System," *Engineering, Technology & Applied Science Research*, vol. 10, no. 1, pp. 5221–5227, Feb. 2020, https://doi.org/10.48084/etasr.3247.

[17] T. T. A. Pham, D. K. T. Thoi, H. Choi, and S. Park, "Defect Detection in Printed Circuit Boards Using Semi-Supervised Learning," *Sensors*, vol. 23, no. 6, Mar. 2023, Art. no. 3246, https://doi.org/10.3390/s23063246.

[18] Z. Yuan, X. Tang, H. Ning, and Z. Yang, "LW-YOLO: Lightweight Deep Learning Model for Fast and Precise Defect Detection in Printed Circuit Boards," *Symmetry*, vol. 16, no. 4, Apr. 2024, Art. no. 418, https://doi.org/10.3390/sym16040418.

[19] B. Du, F. Wan, G. Lei, L. Xu, C. Xu, and Y. Xiong, "YOLO-MBBi: PCB Surface Defect Detection Method Based on Enhanced YOLOv5," *Electronics*, vol. 12, no. 13, Jun. 2023, Art. no. 2821, https://doi.org/10.3390/electronics12132821.

[20] Y. Said and Y. A. Alsariera, "Hardware Implementation of a Deep Learning-based Model for Image Quality Assessment," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 13815–13821, Jun. 2024, https://doi.org/10.48084/etasr.7194.

[21] M. Hussain, "YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision." arXiv, Jul. 03, 2024, https://doi.org/10.48550/arXiv.2407.02988.

[22] G. Fontana, M. Calabrese, L. Agnusdei, G. Papadia, and A. Del Prete, "SolDef_AI: An Open Source PCB Dataset for Mask R-CNN Defect Detection in Soldering Processes of Electronic Components," *Journal of Manufacturing and Materials Processing*, vol. 8, no. 3, May 2024, Art. no. 117, https://doi.org/10.3390/jmmp8030117.

[23] A. Bhattacharya and S. G. Cloutier, "End-to-end deep learning framework for printed circuit board manufacturing defect classification," *Scientific Reports*, vol. 12, no. 1, Jul. 2022, Art. no. 12559, https://doi.org/10.1038/s41598-022-16302-3.

[24] "Ultrasonic Cleaning for Electronic Assemblies? - Fact Checks," *ZESTRON*. https://www.zestron.com/en/know-how/factchecks/ultrasonic-cleaning.

[25] "PCB Ultrasonic Cleaning Guide," *ElectronicsHacks*, Sep. 27, 2023. https://electronicshacks.com/pcb-ultrasonic-cleaning-guide/.

[26] B. Sener, "YOLOv10 Custom Object Detection," *Medium*, May 26, 2024. https://medium.com/@batuhansenerr/yolov10-custom-object-detection-bd7298ddbfd3.

[27] W. Kowalke *et al.*, "A New System Supporting the Diagnostics of Electronic Modules Based on an Augmented Reality Solution," *Electronics*, vol. 13, no. 2, Jan. 2024, Art. no. 335, https://doi.org/10.3390/electronics13020335.

[28] N. Promrit and S. Waichanya, *Fundamentals of Deep Learning in Practice*. Hytexts Interactive Limited.

[29] I. C. Chen, R. C. Hwang, and H. C. Huang, "PCB Defect Detection Based on Deep Learning Algorithm," *Processes*, vol. 11, no. 3, Mar. 2023, Art. no. 775, https://doi.org/10.3390/pr11030775.

[30] C. Chousangsuntorn, T. Tongloy, S. Chuwongin, and S. Boonsang, "A Deep Learning System for Recognizing and Recovering Contaminated Slider Serial Numbers in Hard Disk Manufacturing Processes," *Sensors*, vol. 21, no. 18, Sep. 2021, Art. no. 6261, https://doi.org/10.3390/s21186261.

[31] C. Pramerdorfer and M. Kampel, "A dataset for computer-vision-based PCB analysis," in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, Tokyo, Japan, May 2015, pp. 378–381, https://doi.org/10.1109/MVA.2015.7153209.

[32] Ultralytics, "YOLO Performance Metrics." https://docs.ultralytics.com/guides/yolo-performance-metrics.

[33] L. Lou, K. Lu, and J. Xue, "Multi-Scale Vision Transformer for Defect Object Detection," *Procedia Computer Science*, vol. 222, pp. 397–406, 2023, https://doi.org/10.1016/j.procs.2023.08.178.

[34] T. Najib, F. Muntasir, and W. A. W. Wasi, "Benchmark Study on YOLOv8 Variants in Localized Multiclass Fault Detection in PCBs," in *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, Dhaka, Bangladesh, May 2024, pp. 562–567, https://doi.org/10.1109/ICEEICT62016.2024.10534448.

[35] S. Mukherjee, "The Evolution of YOLO: From Single Shot Detection to State-of-the-Art Object Recognition," *Medium*, Oct. 22, 2023. https://medium.com/@shaomukherjee/the-evolution-of-yolo-from-single-shot-detection-to-state-of-the-art-object-recognition-2670f96f730c.