IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 May 2024, accepted 9 July 2024, date of publication 18 July 2024, date of current version 29 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3430329

RESEARCH ARTICLE

Utilizing YOLO Models for Real-World Scenarios: Assessing Novel Mixed Defect Detection Dataset in PCBs

VINOD KUMAR ANCHA[1], FADI N. SIBAI[2], VENKATESWARLU GONUGUNTLA[3], (Member, IEEE), AND RAMESH VADDI[1]

[1]Department of Electronics and Communication Engineering, SRM University AP, Amaravati, Andhra Pradesh 522502, India
[2]GUST Engineering and Applied Innovation Research Center (GEAR), Gulf University for Science and Technology, Mishref 32093, Kuwait
[3]Symbiosis Centre for Medical Image Analysis, Symbiosis International (Deemed University), Pune 412115, India

Corresponding authors: Venkateswarlu Gonuguntla (venkateswarlu.phd@gmail.com) and Ramesh Vaddi (ramesh.v@srmap.edu.in)

This work was supported in part by the Symbiosis Centre for Medical Image Analysis, Symbiosis International (Deemed University); and in part by the Gulf University for Science and Technology (GUST) Engineering and Applied Innovation Research Center (GEAR), GUST, Mishref, Kuwait.

ABSTRACT In the domain of printed circuit board (PCB) defect detection and classification, the availability of diverse and comprehensive datasets is fundamental for developing effective detection models. However, existing datasets often lack comprehensive labeling and focus on specific defect types, limiting their applicability to real-world scenarios. To address this gap, we introduce a new dataset named 'dataset for Mixed Defect Detection in PCB' (MDD_PCB), which includes intentionally induced mixed PCB defects to provide a more realistic representation of practical scenarios. We evaluate the MDD_PCB dataset using YOLO models and implement it successfully for real-time inference on Jetson Nano, achieving enhanced detection capabilities. Our optimized YOLOv5n model trained on the MDD_PCB dataset achieves impressive metrics (accuracy 93%, precision 95%, recall 96%, mAP 95%, F1-score 94%) with a detection speed of 120.69 frames per second (FPS). Real-time deployment on the Jetson Nano demonstrates practical usability with a detection speed of 30 frames per second (FPS). These results underscore the significance of the diverse dataset proposed, which contributes to robust detection solutions and advances in PCB defect detection methodologies.

INDEX TERMS YOLO models, PCB defect detection, classification, printed circuit boards (PCBs), PCB dataset, NVIDIA Jetson nano, deep learning.

I. INTRODUCTION

Printed circuit boards (PCBs) are indispensable components for electronic devices, ensuring connectivity and structural integrity. They are widely utilized in various electronic devices, extending beyond simple models. Valued at $82 billion in 2022, the global PCB market is predicted to expand to approximately $140.73 billion by 2032 [1]. Yet, as PCBs decrease in size, identifying flaws becomes increasingly challenging. Therefore, implementing accurate defect detection techniques during manufacturing is crucial for enhancing product quality and reducing company expenditures.

Fig. 1 illustrates common PCB defects found in manufacturing industries. The methods for detecting PCB defects include manual visual inspection, electrical testing, and optical inspection [2]. Manual inspection is inefficient due to low accuracy and productivity. Electrical testing requires complex circuits and expensive equipment, with limitations in detecting issues in multi-layer PCBs and potential secondary damage. Automated Optical Inspection (AOI) utilizes cameras and image processing to identify defects such as missing components and soldering issues in PCBs during manufacturing. AOI systems, widely used in electronics production, offer a non-contact method using machine vision

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

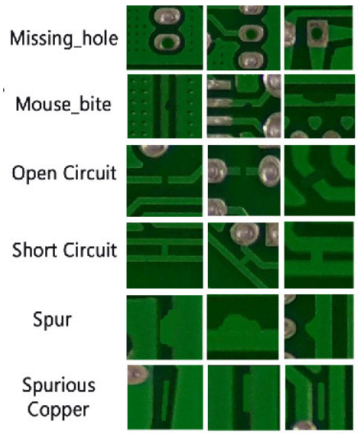

FIGURE 1. Common PCB defects found in manufacturing industries.

technology, ensuring more precise and faster defect detection than alternative approaches [3], [4]. However, they are slow, sensitive to environmental conditions, and are less effective for complex defects [5].

Deep learning, particularly convolutional neural networks (CNNs), has gained popularity for object detection in computer vision. Object detection techniques in deep learning are categorized into two main approaches: two-stage and one-stage methods. Two-stage detection involves generating potential candidate boxes and then classifying them using CNNs, with common algorithms including RCNN and Faster R-CNN. However, Faster R-CNN [6], while accurate, suffers from slow processing speed due to the region proposal network (RPN) component, posing challenges for real-time applications.

One-stage detection methods such as SSD [7] and YOLO [8] directly convert object localization into a regression problem without using candidate box sampling. They offer higher accuracy and faster training compared to multi-stage methods by capturing essential image features in a single step. However, SSD can struggle with detecting small objects due to its reliance on low-resolution features. As a result, YOLO models, particularly YOLOv5 [9], are preferred for object detection tasks [10], [11] because of their balanced performance in terms of speed, accuracy, and robustness.

Having access to varied and comprehensive datasets is essential for developing effective models to detect PCB defects. Nonetheless, current datasets tend to concentrate on specific defect types and have incomplete labeling. To address this limitation, we have introduced a 'dataset for Mixed Defect Detection in PCB' (MDD_PCB), which includes mixed PCB defects to provide a more comprehensive depiction of real-world defect scenarios. The dataset 'MDD_PCB' contains intentionally induced mixed PCB defects in 640 × 640 pixel regions of interest (ROI), aimed at improving training efficiency and practical applicability for real-world scenarios. The dataset was evaluated with YOLO models and successfully implemented for real-time inference on Jetson Nano, demonstrating enhanced detection capabilities.

II. MATERIALS AND METHODS

A. EXISTING DATASET

The dataset TDD_PCB, derived from [12] and abbreviated as Tiny Defect Detection Printed Circuit Board, comprises 693 annotated images of PCB defects with a high resolution of 2777 × 2138 pixels. It covers various defect categories like missing holes, mouse bites, open circuits, shorts, spurs, and spurious copper [13]. This dataset is valuable for studying and developing detection and classification algorithms tailored for PCB defect analysis. The dataset's high resolution and narrow focus on single defect types could slow down training and limit diversity in PCB defect types due to insufficient labeling, leading to increased computational demands and longer training times, potentially affecting testing and training accuracy.

B. PROPOSED DATASET

The current dataset, facing the aforementioned issues, was considered inappropriate for real-world applications. The proposed solution involves creating a specialized dataset for PCB defect detection and classification, focusing on ROI by intentionally inducing multiple mixed defects. Intentionally introduced defects in the dataset improve algorithm training by providing diverse and realistic scenarios, essential for precise detection and classification of PCB defects in manufacturing. The dataset, consisting of images sized at 640 × 640 pixels, went through multiple processing stages including image labeling, preprocessing, defect detection, and classification. This was done to improve training efficiency and reduce memory usage. Afterward, we utilized the Roboflow annotation software for image labeling due to its user-friendly interface and cost-effectiveness [14]. To illustrate, Fig. 2 displays each annotated point generated with Roboflow, which contributed to enhancing the overall quality of the dataset. These annotations were saved in.txt format for subsequent use in training the YOLO algorithm for evaluation purposes. The 'Bounding box annotations tool' feature improved accuracy and expanded the number of annotations for the target object.

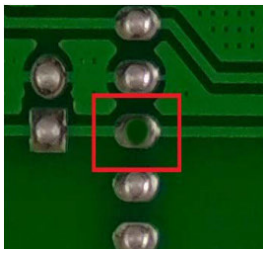

FIGURE 2. Labeling image in Roboflow bounding box tool.

The resulting dataset comprises 1741 PCB images, each featuring two to three defects, offering a diverse training set with multiple defect combinations. In total, there are 3704 distinct defects, with precise labeling for each image reflecting the specific defect types. These classifications represent common defects that arise in PCB manufacturing.

TABLE 1. Comparison of the existing and proposed datasets.

Dataset / Defect	TDD_PCB: Total of 2953 defects over 693 images (2777 x 2138 pixels)		MDD_PCB: Total of 3704 defects over 1741 images (640 x 640 pixels)	
PCB Defect type	Number of images	Number of defects	Number of images	Number of defects
Missing hole	115	497	600	600
Mouse bite	115	492	674	674
Open Circuit	116	482	847	847
Short circuit	116	491	483	483
Spur	115	488	442	442
Spurious copper	116	503	658	658
Total	693	2953	1741	3704

The proposed dataset can be openly accessed from 'Mendeley Data', link: https://data.mendeley.com/datasets/fj4krvmrr5/2.

C. EXISTING VS PROPOSED DATASET COMPARISONS

Table 1 provides the dataset's distribution of various defects along with the number of images with the specific defect type. In the existing dataset, each image contains multiple instances of a single type of defect, resulting in 693 images with a total of 2953 defects. Furthermore, the dataset has a high resolution. These factors make model learning challenging and complex, making it unsuitable for real-world scenarios. In the proposed dataset, there are multiple unique defects in each image, but each defect occurs just once in each image. As a result, the dataset has 1741 images with 3704 defects in total. Fig. 3 provides a visual comparison between the existing and proposed datasets, showcasing one sample image from each dataset for illustration purposes.

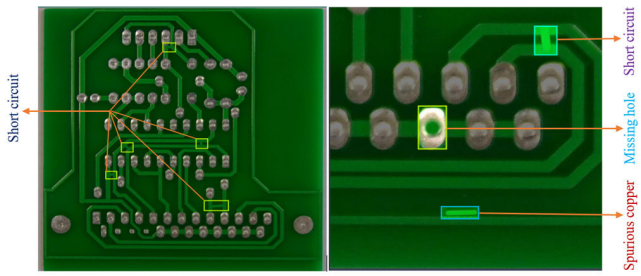

FIGURE 3. (a) TDD_PCB Dataset (b) MDD_PCB dataset.

D. PREPROCESSING

To address the scarcity of large-scale PCB defect datasets due to confidentiality and cost constraints, we used data augmentation techniques to enrich and diversify our dataset. Six traditional augmentation methods were applied, including adding Gaussian noise, adjusting lighting, rotating images, flipping, random cropping, and shifting. Fig. 4 demonstrates the sample representation of the augmentation methods utilized in this dataset preparation.

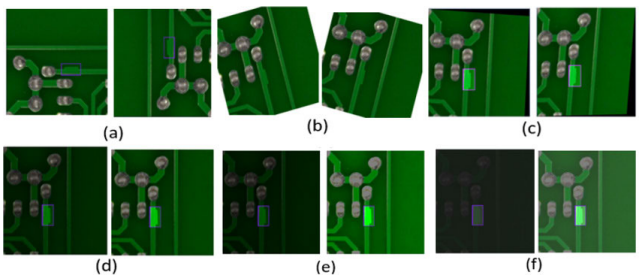

FIGURE 4. Example of the augmentation methods used in the proposed dataset (a) Flip horizontal/vertical (b) Rotation - range [−14° 14°] (c) Shear ±15° (d) Saturation - range [−73 73] (%) (e) Brightness - range [-51 51] (%) (f) Exposure - range [-25 25] (%).

Basic augmentation techniques like Gaussian noise addition and lighting adjustment do not require annotation changes for bounding boxes. However, advanced augmentation techniques like image rotation, cropping, and shifting necessitate modifications to the corresponding bounding box annotations [15]. Overall, after preprocessing the dataset resulted in 8705 images with 18520 defects.

The augmented dataset with precise annotations improves practical applicability by incorporating realistic defects found in real-world scenarios, enabling efficient development and evaluation of PCB defect detection and classification models. Following this, the MDD_PCB dataset underwent evaluation

using YOLO models and was implemented on Jetson Nano to validate enhanced capabilities for detecting PCB defects.

III. METHODOLOGY

A. DATA

During experiments, to ensure consistency and comparability between the TDD_PCB and MDD_PCB datasets, we cropped each image from TDD_PCB to a central area of 1280 × 1280 pixels and resized it to 640 × 640 pixels, matching the resolution of MDD_PCB. This preprocessing step ensures almost complete defect coverage and standardizes image sizes. Post-preprocessing, MDD_PCB contains 8705 images, and TDD_PCB contains 3465 images, all at a resolution of 640 × 640 pixels. The dataset has been split into a ratio of 8:1:1 for training, validation, and testing purposes.

B. MODELS

YOLO models are popular for object detection because of their speed, accuracy, and reliability. YOLOv5 offers enhanced performance over earlier versions due to its improved architecture and optimizations, making it an ideal choice for detecting PCB defects in our study. While simulations were conducted across various versions from YOLOv5 to YOLOv8, this paper focuses specifically on the architecture of YOLOv5 due to simulation results indicating that YOLOv5 yielded the highest performance.

The YOLOv5 architecture, depicted in Fig. 5, consists of three main components: backbone, neck, and detection heads. The backbone, utilizing CSPDarknet53, extracts fine features at different scales using pre-trained networks and improves feature quality with focus structures and the Cross-Stage Partial Network (CSPNet) [16]. The neck networks, incorporating Feature Pyramid Network (FPN) [17] and Path Aggregation Network (PANet) techniques, facilitate efficient spatial function sharing. The prediction head handles confidence estimation and bounding box regression based on anchor priors.

All YOLOv5 models share common components, including CSP-Darknet53 for the backbone, SPP (Spatial Pyramid Pooling), PANet in the neck, and the detection head from YOLOv4. YOLOv5 models are categorized based on size: YOLOv5n (nano), YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large), and YOLOv5x (extra large), which are adjusted using parameters like depth_multiple and width_multiple to vary network depth and width. These adjustments affect the number of Bottleneck Cross-Stage Partials (BCSPs) and convolution kernels, to suit different application needs.

Among all the versions, choosing YOLOv5n was essential for maximizing efficiency within platform constraints and maintaining real-time processing capabilities on Jetson Nano. This focused approach provided valuable insights into practicality and performance. As a result, all nano models ranging from YOLOv5n to YOLOv8n were evaluated for further analysis and comparisons.

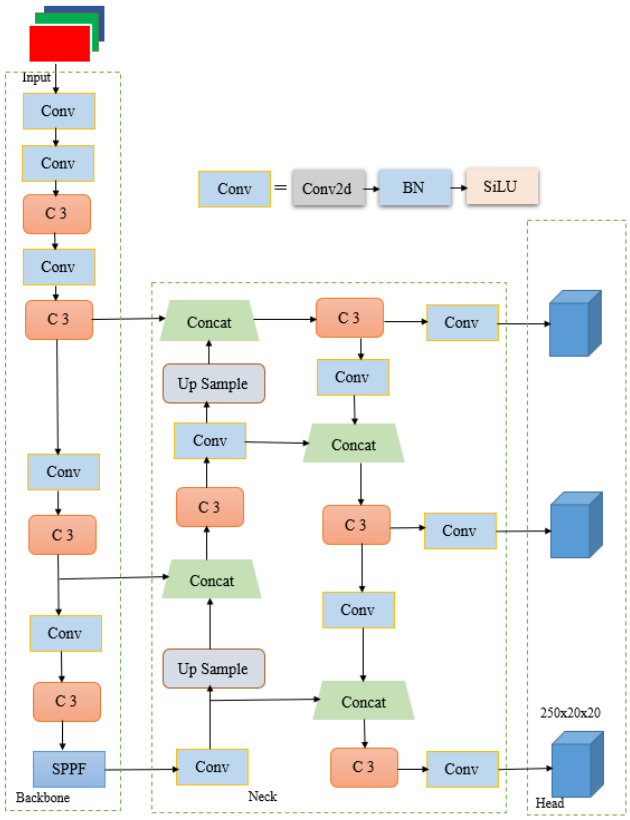

FIGURE 5. Architecture of Yolov5 model.

IV. SIMULATION RESULTS

All YOLO models from YOLOv5n to YOLOv8n [18], [19], [20] were examined using the stochastic gradient descent optimizer, with specific hyperparameters including an initial learning rate of 0.001, momentum of 0.9, and weight decay of 0.0005 for both TDD_PCB (existing) and MDD_PCB (proposed) datasets. The datasets were split into training, validation, and testing sets at a ratio of 8:1:1 for evaluation. To ensure effective learning, we trained the models for 300 epochs in each trial, maintaining consistency and optimizing performance on the datasets.

A. TRAINING

Transfer learning is used to accelerate training and improve object detection accuracy by leveraging pre-trained models. In this study, a model initially trained on the COCO dataset is adapted for PCB detection [21], [22], [23], [24], [25], [26] (depicted in Fig. 6), enhancing its ability to recognize and classify PCB defects in real-world scenarios.

B. EVALUATION METRICS

The essential metrics for assessing network applicability in real-world defect detection include precision (P), recall (R), speed (measured by frames per second (FPS)), and network complexity (evaluated by the total number of parameters). These metrics reflect the model's efficiency and accuracy in practical deployment scenarios. P, R, and mean average

TABLE 2. Performance of YOLO models using TDD_PCB and MDD_PCB.

Model	Data	Precision	Recall	mAP 0.5	Accuracy	F1-Score	Parameters (M)	Flops (G)	Speed (FPS)	Memory Usage (MB)
Yolov 8n	TDD_PCB	0.88	0.94	0.90	0.90	0.90	3.0	8.1	112.13	6.23
	MDD_PCB	0.85	0.94	0.93	0.90	0.92				
Yolov 7n	TDD_PCB	0.93	**0.97**	0.89	0.91	0.86	2.34	6.9	115.73	71.3
	MDD_PCB	0.90	0.98	0.94	0.91	0.89				
Yolov 6n	TDD_PCB	0.85	0.54	0.85	0.89	0.64	4.7	11.4	118.69	39.3
	MDD_PCB	0.95	0.80	0.95	0.92	0.86				
Yolov 5n	TDD_PCB	0.92	0.95	0.92	0.91	0.91	1.7	4.2	120.69	3.87
	MDD_PCB	**0.95**	0.96	**0.95**	**0.93**	**0.94**				

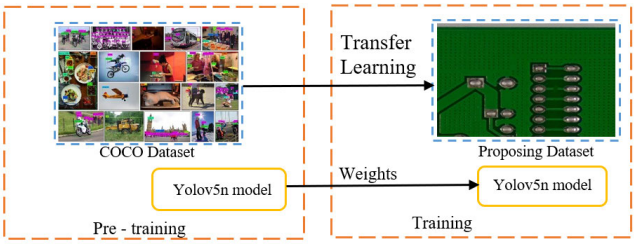

FIGURE 6. PCB defect detection training process (Transfer learning utilization).

precision (mAP) are calculated as follows:

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

$$mAP = \frac{\sum_{n=1}^{M} AP_n}{M} \tag{3}$$

where M is the total number of classes and AP_n is the average precision of the class 'n'. TP, FN, and FP are the true positive, false negative, and false positive, respectively. The detection speed is measured in FPS and for mAP, higher mAP values indicate improved detection performance.

C. PERFORMANCE OF YOLO MODELS

All simulations were conducted using Ubuntu 22.04.2 LTS, Python 3.8, PyTorch 1.10.0-GPU, CUDA 11.3, and CUDNN 8.2.2. The experiments with the TDD_PCB and MDD_PCB datasets were carried out, and the results were tabulated in Table 2. This table provides a comprehensive overview of each dataset's performance with the YOLO models. The proposed datasets demonstrated significantly enhanced results compared to the existing dataset, highlighting their effectiveness in improving the accuracy and reliability of PCB defect detection. Furthermore, accuracy and mAP comparisons using both datasets are illustrated in Fig. 7, underscoring the promising potential of the proposed dataset for enhancing defect detection systems in the PCB domain. Careful labeling and organization of defect classes in the proposed datasets played a critical role in achieving these improvements, making them suitable for resource-constrained applications.

The YOLOv5 model was employed to demonstrate PCB defect detection using test images from TDD_PCB and MDD_PCB datasets, and is as shown in Fig. 8. The results demonstrate enhanced detection accuracy with consistently higher confidence scores for MDD_PCB compared to TDD_PCB, achieving a detection speed of 120.69 FPS for MDD_PCB. This enhancement in precision and reliability is attributed to MDD_PCB's intentional inclusion of mixed defects, which offers a more diverse learning environment for the model and demonstrates significant progress in PCB defect detection. Therefore, the proposed MDD_PCB dataset offers a more accurate and comprehensive benchmark for evaluating PCB defect detection approaches.

Also, we tested the performance and memory utilization of all YOLO models, and the results are tabulated in Table 2. The table shows that YOLOv6n, YOLOv7n, and YOLOv8n exhibit increasingly lower frames per second (FPS) with larger inference times. Among these, YOLOv7n has the highest memory utilization at 71.3 MB. Despite this, all versions maintain a high degree of processing speed, with YOLOv5n achieving the highest total efficiency, making it ideal for real-time applications. These results indicate that YOLOv5n has the fastest FPS at 120.69 and the lowest memory consumption at 3.87 MB, making it the most efficient in terms of both speed and resource utilization. This is attributed to YOLOv5n's optimized design, which effectively balances computational efficiency with resource management.

Of all the models, YOLOv5n exhibits the best performance with the fewest parameters and flops, making it ideal for real-time deployment. Next, we demonstrate the implementation of a real-time PCB defect detection system using the trained model's deployment on the Jetson Nano board by refining and adapting the model with Python scripts optimized for the board's CUDA capabilities.

V. EXPERIMENTAL RESULTS

A. IMPLEMENTATION DETAILS OF A REAL-TIME PCB DEFECT DETECTION SYSTEM

We utilized the Jetson Nano board to deploy the selected model. The Nvidia Jetson Nano, known for its compact size and energy efficiency, features a quad-core processor, 128-core Max GPU, and ARM Cortex-A57 CPU, with a

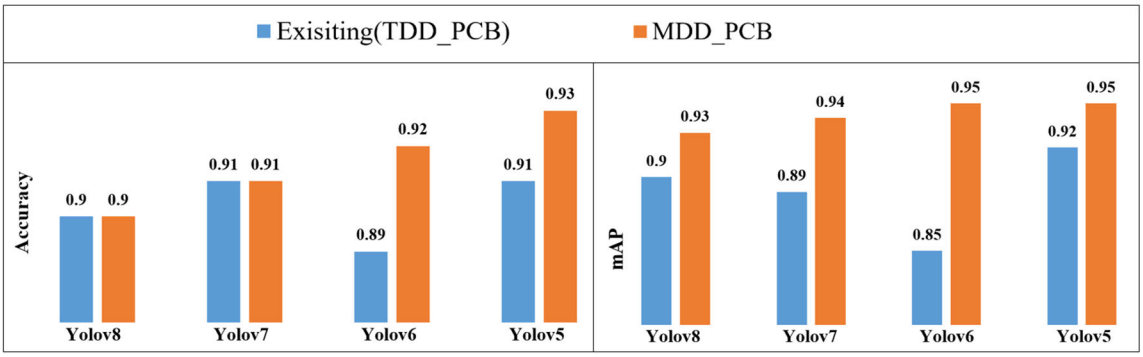

FIGURE 7. Accuracy and mAP of using TDD_PCB and MDD_PCB datasets.

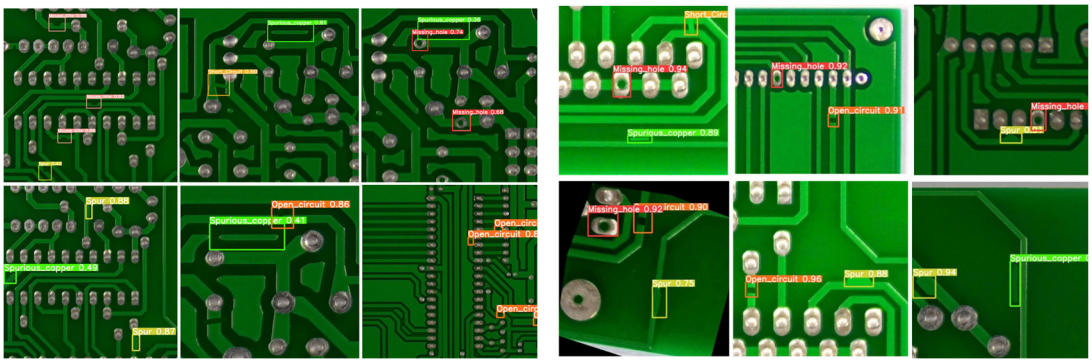

FIGURE 8. Illustration of PCB defect detection in (a) TDD_PCB and (b) MDD_PCB using YOLOv5.

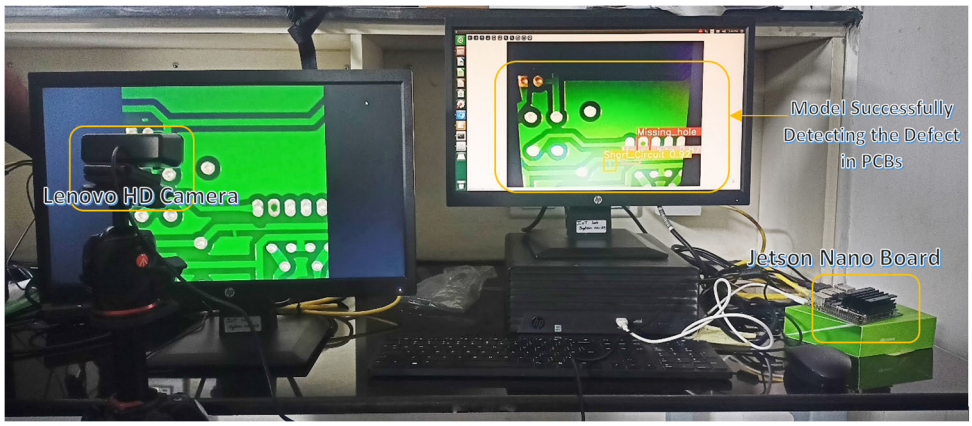

FIGURE 9. Implementation of real-time PCB defect detection system.

memory bandwidth of 25.6 GB/s facilitating fast CPU-GPU communication [27]. Its low power consumption and high memory bandwidth make it well-suited for AI system deployments.

To implement a real-time PCB defect detection system, we opted for the YOLOv5n model due to its superior performance with minimal parameters and flops, ideal for real-time applications. We utilized a Lenovo FHD webcam to capture live stream images of PCBs, which were displayed on a monitor. The Nvidia Jetson Nano served as the processing unit, with an additional monitor used for displaying the effective detection and classification of PCB defects. The complete implementation of the real-time PCB defect detection system, incorporating the described details, is illustrated in Fig. 9.

B. PERFORMANCE OF THE REAL-TIME PCB DEFECT DETECTION SYSTEM

The deployment process involves fine-tuning and adapting the selected YOLOv5n model for the Jetson Nano board using Python scripts optimized for CUDA. PCB images captured by the Lenovo FHD webcam were sourced from the live display on the monitor. We aimed to assess the model's performance when deployed on the Jetson Nano. To achieve optimized inference, we converted our model from PyTorch training to

FIGURE 10. Performance of the real-time PCB defect detection system using the MDD_PCB dataset.

TensorRT. This successful transition significantly reduced the memory footprint and computational demands and resulted in a compact 3.87 MB model with an inference time of 33.32 ms.

Real-time PCB defect detection using the Jetson Nano board and the optimized YOLOv5 model (implemented as a TensorRT model) demonstrated significant improvements in detection accuracy and confidence scores, as depicted in Fig. 10. Jetson Nano's computational capabilities helped to achieve quick inference in defect identification, making it suitable for industrial settings requiring rapid and accurate defect detection. Results also highlight the cost-effective and efficient solutions for PCB defect detection, addressing critical needs in manufacturing quality control and demonstrating the feasibility of real-time object detection on edge computing systems like the Jetson Nano.

VI. CONCLUSION

In this work, we introduced the MDD_PCB dataset, a dataset for mixed defect detection in PCB, to address the limitations observed in current datasets for PCB defect detection. Existing datasets primarily emphasize single defect types and often lack detailed labeling. In contrast, the MDD_PCB dataset overcomes these challenges by intentionally introducing mixed defects into the images. By incorporating mixed defects, the proposed MDD_PCB dataset offers a more comprehensive representation of real-world defect scenarios encountered in manufacturing industries, thus addressing the shortcomings of existing PCB defect datasets. The dataset was assessed using YOLO models, particularly YOLOv5 when compared with other models, which demonstrated high accuracy in defect detection as evidenced by metrics such as precision, recall, mAP, and F1-score. The increased diversity of defects in the dataset facilitated the learning of more robust and generalized features by the models, while the standardized resolution contributed to improved model training and generalization capabilities.

The model's successful deployment on the Jetson Nano underscores its practical usability in real-world deployment scenarios. The advancements introduced by the proposed dataset significantly advance PCB defect detection methodologies by offering a curated dataset featuring intentionally induced mixed defects, which is critical in robust and reliable PCB defect identification systems. These methodologies find applications in electronics manufacturing, quality control, and automated inspection processes, enhancing defect detection precision and reliability. Future works will focus on expanding the dataset's defect diversity and variants to refine the model by reducing parameters and floating-point calculations, resulting in more efficient and lightweight models with high performance. Additionally, future work will explore integration with emerging technologies such as artificial intelligence and edge computing. Collaboration with industry partners will be critical for testing these advancements in real-world production environments, fostering acceptance, and maximizing impact.

ACKNOWLEDGMENT

The authors would like to thank Weibo Huang and Peng Wei for making the existing PCB dataset open source. This dataset has been crucial to their study and the field of defect detection in PCBs.

REFERENCES

[1] *Printed Circuit Board (PCB) Market Size*, Precedence Res., Ottawa, ON, Canada, 2023.
[2] L. Deng, "Research on PCB surface assembly defect detection method based on machine vision," M.S. thesis, Dept. Electron. Eng., Wuhan Univ. Technol., Wuhan, China, 2019.
[3] Z. Yun, L. Zhi-Gang, and Z. Yu-Qiang, "Research progress and prospect of machine vision technology," *J. Graph.*, vol. 41, no. 1, p. 871, 2020.
[4] N. K. Khalid, Z. Ibrahim, and M. S. Z. Abidin, "An algorithm to group defects on printed circuit board for automated visual inspection," *Int. J. Simul. Syst. Sci.*, vol. 9, no. 2, pp. 1–10, 2008.
[5] Q. Ling and N. A. M. Isa, "Printed circuit board defect detection methods based on image processing, machine learning and deep learning: A survey," *IEEE Access*, vol. 11, pp. 15921–15944, 2023, doi: 10.1109/ACCESS.2023.3245093.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, and C-Y. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
[9] (2020). *Ultralytics.YOLO V5 Online*. [Online]. Available: https://github.com/ultralytics/yolov5
[10] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Deep convolutional neural network for enhancing traffic sign recognition developed on YOLO v4," *Multimedia Tools Appl.*, vol. 81, no. 26, pp. 37821–37845, Nov. 2022, doi: 10.1007/s11042-022-12962-5.
[11] J. Zhao, C. Li, Z. Xu, L. Jiao, Z. Zhao, and Z. Wang, "Detection of passenger flow on and off buses based on video images and YOLO algorithm," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 4669–4692, Feb. 2022, doi: 10.1007/s11042-021-10747-w.
[12] W. Huang and P. Wei, "A PCB dataset for defects detection and classification," 2019, *arXiv:1901.08204*.
[13] R. Ding, L. Dai, G. Li, and H. Liu, "TDD-Net: A tiny defect detection network for printed circuit boards," *CAAI Trans. Intell. Technol.*, vol. 4, no. 2, pp. 110–116, Jun. 2019, doi: 10.1049/trit.2019.0019.
[14] (2021). *Give Your Software the Sense of Sight*. Accessed: Mar. 1, 2024. [Online]. Available: https://roboflow.com/
[15] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proc.*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.
[16] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.
[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.
[18] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.
[19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.
[20] G. Jocher, A. Chaurasia, and J. Qiu. (2023). *YOLO By Ultralytics*. [Online]. Available: https://github.com/ultralytics/ultralytics
[21] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl.-Based Syst.*, vol. 80, pp. 14–23, May 2015, doi: 10.1016/j.knosys.2015.01.010.
[22] C. K. Man, M. Quddus, and A. Theofilatos, "Transfer learning for spatio-temporal transferability of real-time crash prediction models," *Accident Anal. Prevention*, vol. 165, Feb. 2022, Art. no. 106511, doi: 10.1016/j.aap.2021.106511.
[23] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/tkde.2009.191.
[24] G. Li, Z. Song, and Q. Fu, "A new method of image detection for small datasets under the framework of YOLO network," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Oct. 2018, pp. 1031–1035, doi: 10.1109/IAEAC.2018.8577214.
[25] M. Nie and K. Wang, "Pavement distress detection based on transfer learning," in *Proc. 5th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2018, pp. 435–439, doi: 10.1109/ICSAI.2018.8599473.
[26] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[27] Nvidia Corp. (2024). *Jetson NANO Module*. Accessed: Mar. 1, 2024. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano

VINOD KUMAR ANCHA received the master's degree in VLSI and embedded systems from JNT University Kakinada, Andhra Pradesh, India, in 2021. He is currently a Research Scholar with the School of Engineering and Sciences (SEAS), SRM University AP, Andhra Pradesh, India. His research interests include deep learning, object detection and classification, data preprocessing, and lightweight deep learning architectures.

FADI N. SIBAI received the B.S. degree in electrical (computer) engineering from The University of Texas at Austin and the M.S. and Ph.D. degrees in electrical (computer) engineering from Texas A&M University. He joined GUST as the Associate Dean of the College of Engineering and Architecture, in 2022. Prior to that, he was the acting Dean of the School of Engineering, American International University, as the Dean of the College of Computer Engineering and Science, Prince Mohammad Bin Fahd University, and as the Program Director of the College of Information Technology, UAE University. He also taught engineering with the University of California at San Jose and the University of Akron, USA. At UAE University, he also established and directed an IBM-funded Research and Development Center. He received IBM and nVIDIA equipment grants, an IBM Faculty Award, and research grants from the Emirates Foundation and the University of Akron. He also has extensive industrial experience, having worked for Saudi Aramco and Intel Corporation, and consulted for HPE, SABIC, and KEO. He has authored or co-authored over 250 publications and technical reports, including more than 75 journal publications. He also served in various capacities on program and organizing committees of over 20 international conferences.

VENKATESWARLU GONUGUNTLA (Member, IEEE) is currently a Research Faculty Member of the Symbiosis Centre for Medical Image Analysis (SCMIA), Symbiosis International University (SIU), Lavele, Pune, India. Prior to joining SCMIA, he held a postdoctoral and senior researcher positions with the Samsung Medical Center, Seoul, South Korea, where he was also affiliated with the Sungkyunkwan University Medical Science Research Institute. His current research interests include deep learning for processing and analyzing medical images, brain networks, bio-signal processing and analysis, graph theory, and neural decoding.

RAMESH VADDI received the Ph.D. degree in microelectronics and VLSI from IIT Roorkee, India, in 2011. He was a Senior Research Fellow with the National University of Singapore, from 2018 to 2019, a Postdoctoral Scholar with The Pennsylvania State University, USA, from 2013 to 2014, and a Postdoctoral Research Fellow with Nanyang Technological University's School of Electrical and Electronic Engineering, from 2011 to 2012. He was an Assistant Professor with IIIT Naya Raipur and Shiv Nadar University, India. He is currently an Associate Professor and the Head of the ECE Department, SRM University AP, Amaravati, Andhra Pradesh, India. His research interests include post-CMOS devices for energy-efficient VLSI designs, the IoT hardware security, VLSI accelerators, memory-based computing architecture for deep learning, object detection and classification, and data preprocessing.