Instance segmentation for PCB defect detection with Detectron2

Aravalli Sainath Chaithanya, Lavadya Nirmala Devi, Putty Srividya
Department of Electronics and Communication Engineering, University College of Engineering, Osmania University, Hyderabad, Telangana, India

Article Info

Article history:

Received Sep 2, 2024
Revised Mar 17, 2025
Accepted May 23, 2025

Keywords:

Detectron2
Instance segmentation
Masks
Object detection
RoI align

ABSTRACT

Printed circuit boards (PCBs) are essential in modern electronics, where even minor defects can lead to failures. Traditional inspection methods struggle with complex PCB designs, necessitating automated deep learning techniques. Object detection models like Faster R-CNN and YOLO rely on bounding boxes for defect localization but face overlap issues, limiting precise defect isolation. This paper presents a segmentation-based PCB defect detection model using Detectron2's Mask R-CNN. By leveraging instance segmentation, the model enables pixel-level defect localization and classification, addressing challenges such as shape variations, complex structures, and occlusions. Trained on a dataset of 690 COCO-annotated images, the model underwent rigorous experimentation and parameter tuning. Evaluation metrics, including loss functions and mean average precision (mAP), assessed performance. Results showed a steady decline in loss values and high precision for defects like mouse bites and missing holes. However, performance was lower for complex defects like spurs and spurious copper. This study highlights the effectiveness of instance segmentation in PCB defect detection, contributing to improved quality control and manufacturing automation.

Corresponding Author:

Aravalli Sainath Chaithanya
Department of Electronics and Communication Engineering, University College of Engineering, Osmania University
Hyderabad - 500007, Telangana, India
Email: chaitanya.aravalli@gmail.com

1. INTRODUCTION

The electronics industry relies on printed circuit boards (PCBs) for applications ranging from consumer devices to defense systems. During manufacturing [1], factors such as dust, over-etching, and spurious metals can cause dimensional changes in PCB insulators and conductors, leading to defects that affect performance and reliability. Traditionally, human operators inspected PCBs, but with the increasing complexity of modern designs and ultra-large-scale integration, manual methods have become inefficient and subjective, struggling to meet demands for accuracy and efficiency. To address these challenges, automated defect detection methods utilizing deep learning have gained prominence. Advanced object detection models, including region-based convolutional neural networks (R-CNN) [2], Fast R-CNN [3], Faster R-CNN [4], and you only look once (YOLO) [5], have been extensively utilized for PCB defect detection. These methods rely on bounding boxes for defect localization, ensuring computational efficiency but with limitations in precision. During localization, bounding boxes may overlap, making it challenging to accurately isolate defects and capture their intricate morphologies, as illustrated in Figure 1.

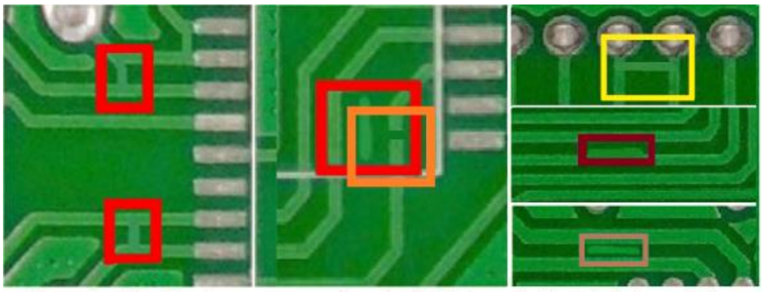

Figure 1. PCB defect detection using bounding boxes

Instance segmentation [6] has emerged as a promising solution to overcome these limitations. Through pixel-level feature extraction, instance segmentation enables precise defect localization and detailed morphological characterization. It effectively manages overlapping bounding boxes, reduces false positives, and ultimately provides a more accurate representation of defect patterns, aiding engineers in root cause analysis and process optimization.

Numerous studies have explored both traditional and deep learning approaches for PCB defect detection. One such study, conducted by Lu *et al*. [7], proposed a framework in which HOG and LBP features were obtained from PCB images for further processing. These extracted features were used to train separate support vector machine (SVM) models, which were later integrated using Bayes fusion theory for defect classification. Their approach showed higher accuracy compared to individual features, though its practical applicability is limited. Likewise, Wei *et al*. [8] developed a CNN model trained on a dataset of 1,818 PCB images, outperforming traditional models like VGG16 and ResNet50 but failing to capture finer defect details. Adibhatla *et al*. [9] worked with a large dataset of 47,428 images to reduce misclassifications, achieving over 85% accuracy; however, their focus was not specifically on defect detection. Other researchers, such as Hu and Wang [10], modified Faster R-CNN for enhanced feature extraction, achieving a mean Average Precision (mAP) of 94.2%. Despite these advancements, they still encountered issues with overlapping bounding boxes during localization, an ongoing challenge in PCB defect detection. Another notable approach, developed by Chaithanya and Devi [11], uses a template-based inspection system for granular defect detection via segmentation instead of bounding boxes. While this method allows for precise localization and analysis, it requires extensive preprocessing and lacks robustness against environmental influences. In contrast, Calabrese *et al*. [12] employed Mask R-CNN for defect detection, eliminating the need for complex preprocessing steps. They achieved satisfactory results on a smaller dataset of missing holes and shorts. However, their study emphasized the necessity for enhancements, particularly in identifying diverse defect categories and improving detection robustness.

Despite advancements in the field, supervised classifiers like support vector machines (SVMs) [13] are effective for simpler tasks but struggle with complex, nonlinear data. Deep learning models like CNNs and R-CNNs often fail to capture intricate defect morphologies, including fine details and varying defect sizes. Although Mask R-CNN has been employed in PCB defect detection, its application has primarily been restricted to specific defect types. Additionally, template-based methods still face practical challenges, such as the need for precise image alignment. This research aims to mitigate these limitations by utilizing Mask R-CNN [14] within the Detectron2 framework [15] for precise defect localization and detailed characterization. By addressing challenges like overlapping bounding boxes, it seeks to enhance the detection of intricate defect morphologies.

The subsequent sections outline the approach adopted in this study, covering the dataset, model design, and training strategy. We then present the experimental findings and analyze their significance in improving PCB defect detection. This work advances defect detection techniques in PCB manufacturing and opens avenues for future research in automated inspection systems.

2. METHOD

Figure 2 illustrates the workflow for PCB defect segmentation, outlining the sequential steps required for accurate detection and segmentation of defects. These steps employ the Mask R-CNN model integrated within the Detectron2 framework, which is renowned for its precision in instance segmentation tasks. This workflow highlights the systematic approach used in this study to address intricate PCB defect localization.

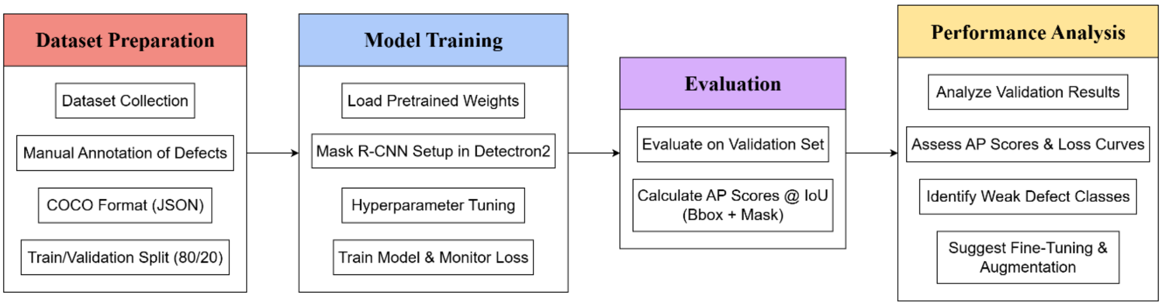

Figure 2. Process flow for instance segmentation of defects

2.1. Dataset preparation and splitting

This study utilizes a dataset of 690 PCB images, systematically categorized into six distinct defect types: missing hole, mouse bite, open circuit, short, spur, and spurious copper, as described in [16]. To enhance model training and reduce classification ambiguity, each image contains a single defect type positioned variably across the PCB, enabling accurate detection and classification.

Instance segmentation requires pixel-level annotations, a process significantly more labor-intensive than traditional bounding box methods. Each defect was manually annotated using Label Studio [17], generating binary masks that delineate defective regions. As outlined in a related study [18], these annotations were converted into the COCO JSON format [19], ensuring compatibility with frameworks like Detectron2 and preserving essential metadata such as segmentation masks, bounding boxes, and class labels. This meticulous preparation facilitates seamless integration into instance segmentation pipelines.

To prepare the dataset for model training and evaluation, it was divided into 80% for training and 20% for validation, with both subsets annotated in the COCO JSON format. This division enabled the model to learn from diverse defect variations while retaining a validation subset for hyperparameter tuning and performance assessment. Following the training process, the model's ability to generalize was assessed on an unseen test dataset, ensuring its robustness in detecting and segmenting a wide range of defect types.

2.2. Detectron2

Detectron2, built on PyTorch, is a versatile deep learning framework widely used for object detection and segmentation tasks. It offers pre-trained models such as Faster R-CNN and Mask R-CNN, known for their robustness in handling complex detection scenarios. Mask R-CNN, in particular, excels in pixel-level localization, making it essential for applications like PCB defect detection, where intricate and overlapping defects require precise segmentation. The modular architecture of Detectron2 allows researchers to customize its functionality for specific use cases. This flexibility facilitates the adaptation of backbone networks, training pipelines, and datasets, making it ideal for industrial applications. Its built-in tools for dataset management, visualization, evaluation, and checkpoint handling further enhance workflow efficiency, ensuring seamless integration into various domains.

In this research, Detectron2 is employed for PCB defect detection using Mask R-CNN, which efficiently handles overlapping defects while capturing fine structural details. The framework supports dataset registration in COCO JSON format, enabling compatibility with other deep learning tools. These features collectively make Detectron2 a powerful choice for tackling the complexities of PCB defect analysis.

2.2.1. Mask R-CNN

Mask R-CNN is a widely adopted instance segmentation model capable of detecting objects using bounding boxes while also predicting pixel-level segmentation masks. Since it operates in a supervised learning paradigm, it requires labeled segmentation masks for training. Built upon Faster R-CNN, Mask R-CNN introduces an additional "mask head" branch, enabling it to generate detailed segmentation masks for each detected object. The architecture is highly adaptable, with components such as the backbone network, region proposal network (RPN), classifier, and mask head configurable for performance optimization. Additionally, hyperparameter tuning allows the model to be tailored for different datasets or specific application needs. The structure of Mask R-CNN, depicted in Figure 3, consists of four key components essential for precise instance segmentation.

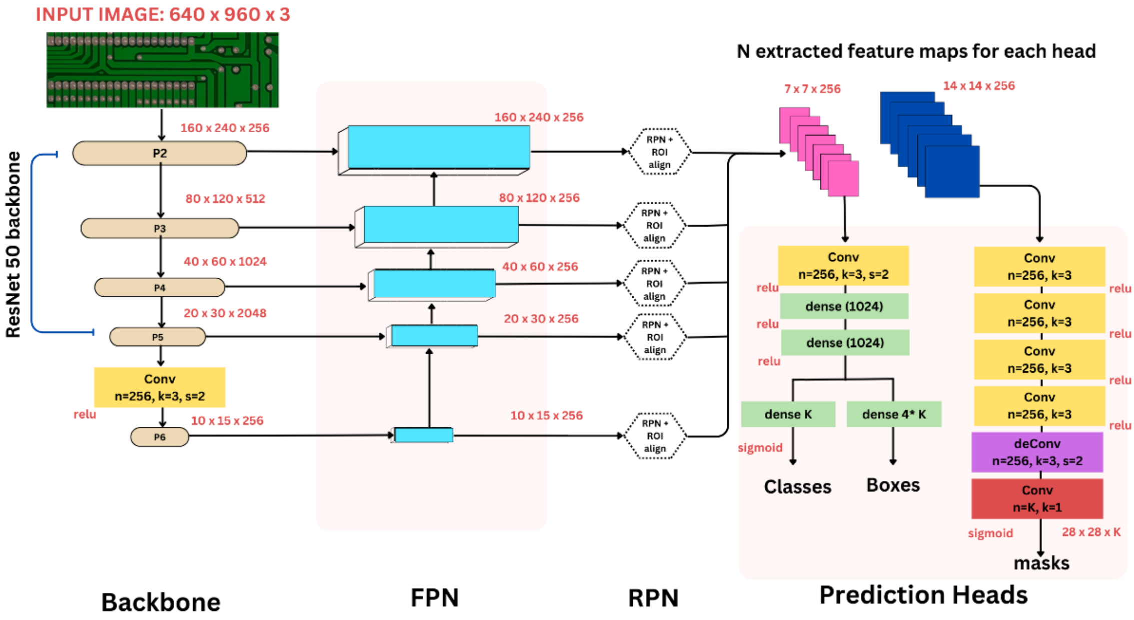

Figure 3. Architectural breakdown of mask R-CNN

a. Backbone Network: Mask R-CNN commonly utilizes ResNet-50 [20] as its backbone due to its strong feature extraction capability and effectiveness in instance segmentation. With 50 layers, it captures intricate image details crucial for precise object detection and segmentation at the pixel level. Residual connections mitigate training challenges, improving learning efficiency. Pretrained on ImageNet, ResNet-50's weights provide a solid foundation for fine-tuning [21], boosting model performance and convergence.
b. Neck: The feature pyramid network (FPN) [22] serves as the "neck" of Mask R-CNN, refining high-level features extracted by the ResNet-50 backbone into a hierarchical multi-scale representation. By integrating both low-resolution and high-resolution features, FPN enhances object detection and segmentation across varying object sizes. It assists the (RPN) in generating high-quality proposals while also enabling precise instance segmentation via the ROI Align layer. This multi-scale feature learning significantly improves Mask R-CNN's performance in complex vision tasks.
c. RPN+RoI Align: The region proposal network (RPN) identifies potential object regions by evaluating anchor boxes of different scales and aspect ratios, refining their coordinates based on objectness scores. It classifies anchors as foreground or background, generating refined proposals. The region of interest (RoI) align layer [23] then processes these proposals using bilinear interpolation to preserve spatial alignment, effectively addressing the misalignment issues caused by quantization in traditional RoI pooling. This ensures precise object localization and pixel-wise segmentation, particularly for small objects, thereby enhancing Mask R-CNN's performance in fine-grained detection tasks.
d. Bounding Box Head: The bounding box head in Mask R-CNN processes the feature maps extracted by the RoI Align layer to perform two tasks: classification of object categories and regression of bounding box coordinates. These are jointly optimized using a multi-task loss function, improving detection precision by refining region proposals and enhancing spatial localization.
e. Mask Head: Mask R-CNN extends Faster R-CNN by incorporating a dedicated "mask head" branch for precise instance segmentation [24]. This branch processes ROI Align features, applying convolutional and upsampling layers to produce dense spatial representations. It outputs a binary mask for each detected object using a fully convolutional network (FCN) [25]. During training, the model optimizes segmentation accuracy by minimizing the difference between predicted and ground-truth masks through a loss function such as binary cross-entropy [26]. This enables accurate pixel-wise segmentation, particularly useful for capturing irregular object boundaries.

2.3. Experimental setup

The Detectron2-based PCB defect segmentation model was executed in a well-configured computational environment.

a. Environment configuration – The model was implemented using Detectron2 0.6, built on PyTorch 2.2, with Python 3.10.12 and CUDA 12.2 for GPU acceleration. Additional dependencies included pycocotools 2.0.2.

b. Computational resources – Training was conducted on Google Colab, utilizing its cloud-based infrastructure with CPU and T4 GPU access. The hardware setup consisted of NVIDIA-SMI 535.104.05, 4 GB RAM, 500 GB storage, and an Intel Core i5-6200U CPU.
c. Training approach – The study employed Mask R-CNN for PCB defect segmentation, with hyperparameter tuning to optimize performance. The next section details training configurations, hyperparameter tuning, and evaluation metrics.

3. RESULTS AND DISCUSSION

3.1. Training configuration and loss function

Training and validation were conducted using a dataset comprising 552 images for training and 138 for validation, allowing the model's performance to be assessed on unseen data. The model was trained using four different configurations, as presented in Table 1. Across all experiments, the NUMBER_OF_CLASSES was set to 6, the PATIENCE parameter to 500, and the MAX_ITER to 1500.

Table 1. Model parameter tuning

Parameter	Exp1	Exp2	Exp3	Exp4
IMS_PER BATCH	2	4	2	4
BASE_LR	0.001	0.00025	0.00025	0.001
ROI_HEADS_BATCH_SIZE_PER_IMAGE	512	256	512	256

Detectron2 offers pre-trained models with optimized weights for standard datasets, making them adaptable for custom datasets. To evaluate their effectiveness on new data, a multi-component loss function is utilized during training.
The total loss function consists of:
a. Classification loss (loss_cls): Measures the classification error.
b. Bounding box regression loss (loss_box_reg): Measures localization error.
c. Segmentation mask loss (loss_mask): Measures segmentation accuracy.

The overall loss is computed as a weighted sum of these components, enabling the model to classify objects, adjust bounding boxes, and generate precise segmentation masks concurrently.

$$loss_{tot} = loss_{cls} + loss_{boxreg} + loss_{mask} \tag{1}$$

3.2. Model evaluation metrics

Model performance is primarily assessed using average precision (AP) metrics, which evaluate both localization and segmentation accuracy. These metrics are based on Intersection over Union (IoU), which quantifies the overlap between predicted and ground truth masks and serves as a scale-invariant measure of detection quality. A commonly used threshold of IoU ≥ 0.50 helps distinguish accurate detections from false positives. Table 2 outlines the evaluation metrics used, including AP across different IoU thresholds and object scales (in pixels).

Table 2. Average precision (AP) and AP across scales

Metric	Description
$AP^{IoU=.50}$	Average precision at intersection over union (IoU) = 0.50
$AP^{IoU=.75}$	Average precision at IoU = 0.75
AP	Average precision across IoU range [0.50:0.95]
AP^{small}	Average precision for small objects (area < 322 px)
AP^{medium}	Average precision for medium objects (322 < area < 962 px)
AP^{large}	Average precision for large objects (area > 962 px)
AP^{all}	Average precision considering all object sizes without regard to size

3.3. Performance analysis

The model's performance was evaluated using loss values from the training dataset and AP metrics from the validation dataset. Among the four experiments, Experiment 4 yielded the best results. Figure 4 presents the loss curves for classification, detection, segmentation, and overall loss, indicating a consistent decline, which signifies effective learning. By iteration 1500, the total loss converged to 1.143 (classification: 0.123, detection: 0.3178, segmentation: 0.3703).

Tables 3 and 4 summarize the model's AP values for bounding box (Bbox) and segmentation (Seg) across different defect categories. The highest precision was observed at IoU=0.50, particularly for *'mouse_bite'* and *'missing_hole*.' In contrast, categories like '*spur*' and *'spurious_copper'* exhibited lower AP values, indicating challenges in detecting smaller or less distinct defects. These findings highlight the need for further fine-tuning and data augmentation to enhance precision for subtle and complex defect type.

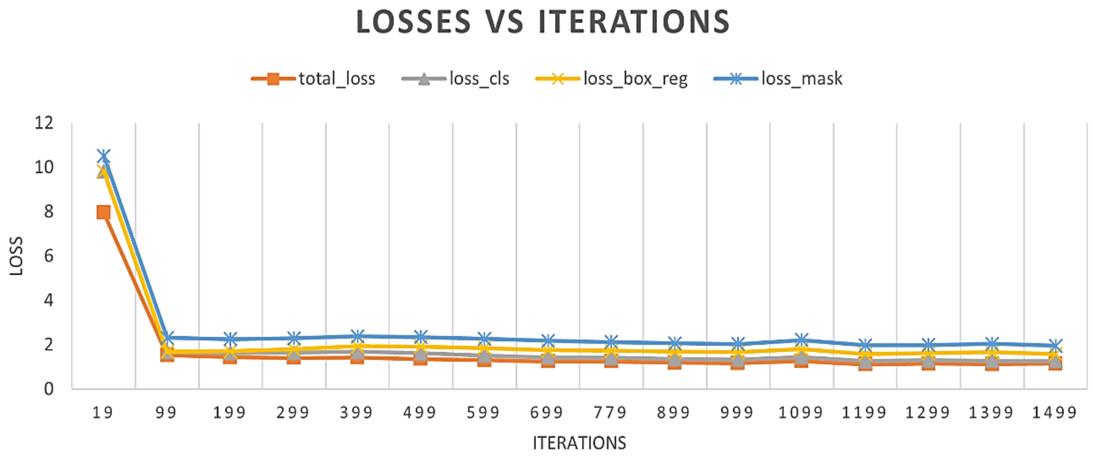

Figure 4. Losses vs iterations curve

Table 3. Evaluation of AP for BBox and mask

Metric	Bbox	Seg
AP	27.07	23.59
AP $^{IoU=.50}$	74.31	69.32
AP $^{IoU=.75}$	11.58	8.08
AP ^{small}	26.64	21.48
AP ^{medium}	31.86	32.20
AP ^{large}	nan	nan

Table 4. AP for each category of defects

Category	Bbox	Seg
missing_hole	28.03	24.72
mouse_bite	35.59	33.41
open_circuit	26.13	23.22
short	27.47	22.39
spur	21.55	15.54
spurious_copper	23.64	20.25

Figure 5 illustrates the model's predictions, including bounding boxes and segmentation masks, for various PCBs in the validation dataset. Each PCB image features a single defect type, selected from six distinct categories: missing hole, mouse bite, open circuit, short, spur, and spurious copper, as described in the dataset section. The consolidated figure offers a clear visualization of the model's performance across these defect types, with the corresponding AP percentages displayed for each defect instance.

3.4. Discussion on key findings

The results demonstrate that Detectron2's Mask R-CNN framework is highly effective for PCB defect detection, particularly for defects like *'mouse_bite*' and *'missing_hole*,' which exhibited high precision. By leveraging instance segmentation, the model improves both object localization and pixel-level segmentation, outperforming traditional image processing and earlier deep learning approaches. Parameter tuning, such as adjusting *IMS_PER_BATCH* and *BASE_LR*, significantly impacted performance, with Experiment 4 yielding the best results. However, subtle defects like '*spur*' and *'spurious_copper*' showed lower precision, highlighting the need for further refinement. Future work should focus on data augmentation and fine-tuning, as increasing dataset diversity—especially for subtle defects—can improve generalization and enhance real-world detection accuracy.

Figure 5. Model predictions of bounding boxes and segmentation masks for various defect instances in PCBs with corresponding AP percentages

4. CONCLUSION

This study demonstrates the efficiency of Detectron2's Mask R-CNN in detecting and segmenting PCB defects, enhancing localization accuracy and minimizing false positives using pixel-level masks. The model achieved high precision for defects like "mouse_bite" and "missing_hole," with bounding box AP of 27.07 and segmentation AP of 23.59. At an IoU threshold of 0.50, it recorded 74.31 (Bbox) and 69.32 (Seg), demonstrating robust performance. However, performance for smaller or more complex defects, such as "spur" and "spurious_copper," was lower, reflecting the need for further refinement, potentially through additional training data or fine-tuning. The consistent decline in classification, detection, and segmentation loss values confirms the model's learning efficacy. This research highlights the potential of segmentation-based models to improve automated PCB inspection, enabling detailed defect analysis to complement traditional methods. Future directions may involve integrating advanced models like YOLO or ensemble approaches to further enhance defect detection for real-time quality control in industrial settings.

ACKNOWLEDGMENTS

The authors would like to specially thank "The Open Lab on Human-Robot Interaction-Peking University" for PCB Defect Dataset.

FUNDING INFORMATION

Author state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Aravalli Sainath Chaithanya	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	
Lavadya Nirmala Devi					✓					✓	✓	✓	✓	
Putty Srividya					✓					✓	✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Author state no conflict of interest.

DATA AVAILABILITY

The data supporting the findings of this study are available from Peking University under license and cannot be publicly shared. Derived segmentation annotations and the code may be granted by the authors upon reasonable request, subject to approval and ethical considerations.

REFERENCES

[1] "PCB manufacturing process: A comprehensive guide," *MKTPCB*, 2023. https://www.mktpcb.com/pcb-manufacturing-process/ (accessed May 11, 2023).

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

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

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[5] A. Sharma, "Introduction to the YOLO family," *PyImageSearch*, 2022. https://pyimagesearch.com/2022/04/04/introduction-to-the-yolo-family/ (accessed Oct. 10, 2023).

[6] H. Bandyopadhyay, "What is instance segmentation, how does it work and what are its real-life applications? Read on to learn how to train your own instance segmentation model on V7 in less than an hour!," *V7*. 2022, Accessed: Nov. 15, 2023. [Online]. Available: https://www.v7labs.com/blog/instance-segmentation-guide.

[7] Z. Lu, Q. He, X. Xiang, and H. Liu, "Defect detection of PCB based on Bayes feature fusion," *Journal of Engineering*, vol. 2018, no. 16, pp. 1741–1745, 2018, doi: 10.1049/joe.2018.8270.

[8] P. Wei, C. Liu, M. Liu, Y. Gao, and H. Liu, "CNN-based reference comparison method for classifying bare PCB defects," *Journal of Engineering*, vol. 2018, no. 16, pp. 1528–1533, 2018, doi: 10.1049/joe.2018.8271.

[9] V. A. Adibhatla, J.-S. Shieh, M. F. Abbod, H.-C. Chih, C.-. C. Hsu, and J. Cheng, "Detecting defects in PCB using deep learning via convolution neural networks," in *2018 13th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, 2018, pp. 202–205, doi: 10.1109/IMPACT.2018.8625828.

[10] B. Hu and J. Wang, "Detection of PCB surface defects with improved Faster-RCNN and feature pyramid network," *IEEE Access*, vol. 8, pp. 108335–108345, 2020, doi: 10.1109/ACCESS.2020.3001349.

[11] A. S. Chaithanya and L. N. Devi, "Localization of defects on PCBs with diverse acquisitions," *Journal of Image and Graphics*, vol. 12, no. 1, pp. 53–65, 2024, doi: 10.18178/joig.12.1.53-65.

[12] M. Calabrese, L. Agnusdei, G. Fontana, and Others, "Application of mask R-CNN for defect detection in printed circuit board manufacturing," *Research Square*, 2023, doi: 10.21203/rs.3.rs-3491517/v1.

[13] X. Hu, “Support vector machine and its application to regression and classification,” M.S. thesis, Graduate College, Missouri State University, Springfield, Missouri, USA, 2017.
[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.
[15] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” *GitHub*. 2019, Accessed: Dec. 08, 2023. [Online]. Available: https://github.com/facebookresearch/detectron2.
[16] “Robotics institute dataset,” *Peking University*. 2023, Accessed: Feb. 16, 2023. [Online]. Available: https://robotics.pkusz.edu.cn/resources/datasetENG/.
[17] Label Studio, “Get started - Label Studio quick start guide,” *Label Studio*. 2023, Accessed: Sep. 11, 2023. [Online]. Available: https://labelstud.io/guide/get_started.html#Quick-start.
[18] A. S. Chaithanya and L. N. Devi, “Precise PCB defect detection via instance segmentation using YOLOv7 and YOLOv8,” *Journal of Electronic Imaging*, vol. 34, no. 3, p. 33018, May 2025, doi: 10.1117/1.JEI.34.3.033018.
[19] “Convert data to COCO JSON,” *roboflow*. 2023, Accessed: Oct. 15, 2023. [Online]. Available: https://roboflow.com/formats/coco-json.
[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
[21] A. S. Chaithanya and M. Rachana, “Identification of diseased papaya leaf through transfer learning,” *Indian Journal of Science and Technology*, vol. 16, no. 48, pp. 4676–4687, 2023.
[22] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.
[23] “Roi align layer,” *Medium*. 2023, Accessed: Nov. 27, 2023. [Online]. Available: https://firiuza.medium.com/roi-pooling-vs-roi-align-65293ab741db.
[24] “Turn faster R-CNN into mask R-CNN,” *Think Autonomous*, 2023. https://www.thinkautonomous.ai/blog/instance-segmentation/ (accessed Dec. 09, 2023).
[25] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017, doi: 10.1109/TPAMI.2016.2572683.
[26] “Probabilistic losses.” 2023, Accessed: Oct. 26, 2023. [Online]. Available: https://keras.io/api/losses/probabilistic_losses/.

BIOGRAPHIES OF AUTHORS

Aravalli Sainath Chaithanya is a research scholar in the Department of Electronics and Communication Engineering (ECE) at University College of Engineering, Osmania University, Hyderabad, and an assistant professor at Rajiv Gandhi University of Knowledge Technologies (RGUKT), Basar, Telangana. He holds an M.Tech. in VLSI system design and a B. Tech in ECE from JNTU Hyderabad. With nine years of teaching experience, his research interests include image processing, neural networks, computer vision, machine learning, and VLSI. He specializes in digital system design, system-on-chip (SoC) design, and verification. He can be contacted at email: chaitanya.aravalli@gmail.com.

Lavadya Nirmala Devi received her B.Eng., M.Eng., and Ph.D. degrees in electronics and communication engineering from Osmania University, Hyderabad, India, where she is currently a professor in the Department of ECE. With over 20 years of teaching experience, she has expertise in subjects such as signals and systems, digital signal processing, neural networks, AI/ML, IoT, wireless communications, and research methodology. Her research interests include ad-hoc networks, wireless communication, IoT, signal processing, and machine learning. She is actively involved in research projects funded by MeiTY, DST, and UGC, and collaborates with TIHAN at IIT Hyderabad on autonomous navigation. She can be contacted at email: nirmaladevi@osmania.ac.in.

Putty Srividya received her B.Tech. in ECE from JNTUH, and M.E. in signal processing and Ph.D. from Osmania University, Hyderabad. She is an assistant professor in the Department of ECE at University College of Engineering, Osmania University, with over 15 years of teaching experience. Dr. Srividya has served as a TPC member for CIIS (2019–2024) and is a member of IEEE, the signal processing society, and the sensors council. Her research has been published in reputed journals like ACM and Springer, and she has chaired sessions at various conferences and expos. Her research focuses on wireless sensor networks, signal processing, IoT, and machine learning. She can be contacted at email: puttysrividya8@gmail.com.