

# Object detection in printed circuit board quality control: comparing algorithms faster region-based convolutional neural networks and YOLOv8

Jaja Kustija<sup>1</sup>, Diki Fahrizal<sup>2</sup>, Muhamad Nasir<sup>3</sup>, Andi Adriansyah<sup>4</sup>, Muhammad Husni Muttaqin<sup>2</sup>

<sup>1</sup>Program Study of Electrical Engineering Education, Faculty of Technology and Vocational Education, Universitas Pendidikan Indonesia, Bandung, Indonesia

<sup>2</sup>Master of Electrical Engineering, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung, Indonesia

<sup>3</sup>Research Center for Environment and Clean Technology, BRIN, Bandung, Indonesia

<sup>4</sup>Department of Electrical Engineering, Universitas Mercu Buana, Jakarta, Indonesia

## Article Info

### Article history:

Received Jul 1, 2024

Revised Dec 29, 2024

Accepted Jan 16, 2025

### Keywords:

Automated optical inspection  
Comparison of algorithm  
Faster region-based  
convolutional neural networks  
Printed circuit board defect  
detection  
YOLOv8

## ABSTRACT

Along with the development of electronic technology, the integration of numerous components on printed circuit board (PCB) boards has resulted in increasingly complex and intricate layouts. Small defects in traces can lead to failures in electronic functions, making the inspection of PCB surface layouts a critical process in quality control. Given the limitations of manual inspection, which struggles to detect such defects due to their size and complexity, there is a growing need for a PCB inspection system that utilizes automated optical inspection (AOI) based on deep learning detection. This research develops and compares two deep learning algorithms, faster region-based convolutional neural networks (R-CNN) and YOLOv8, to identify the most effective algorithm for detecting defects on PCB layouts. The findings of this study indicate that the YOLOv8 algorithm outperforms faster R-CNN, with the YOLOv8x variant emerging as the best model for defect detection. The YOLOv8x model achieved performance scores of 0.962 (mAP@50), 0.503 (mAP@50:95), 0.953 (Precision), 0.945 (Recall), and 0.949 (F1-score). These results provide a strong foundation for further research into the application of AOI for PCB defect detection and other quality control processes in manufacturing, using optimized deep learning models.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Jaja Kustija

Program Study of Electrical Engineering Education, Faculty of Technology and Vocational Education, Universitas Pendidikan Indonesia

40154, Bandung, Indonesia

Email: jaja.kustija@upi.edu

## 1. INTRODUCTION

Printed circuit boards (PCBs) are essential components in modern electronics providing the foundation for electrical connections through conductive channels, pads, and solder joints. However, PCB surfaces often suffer from defects such as missing holes, mouse bites, open circuits, shorts, spurs, and spurious copper [1], [2]. These flaws can lead to electronic device malfunctions and reduced efficiency, issues that are further exacerbated by the growing complexity of PCB designs driven by the portable electronics industry [3]. Traditional manual inspection methods, while initially cost-effective, have become inadequate due to their subjectivity, inconsistency, and susceptibility to inspector fatigue, making them unsuitable for modern high-volume production environments [4], [5]. To address these limitations, automated

optical inspection (AOI) systems have emerged as a technological leap in quality control for PCB manufacturing. AOI systems analyze high-resolution PCB images using algorithms to identify defects based on predefined criteria, including reference-based comparisons, compliance checks, or hybrid approaches [6]. The integration of deep learning techniques has further revolutionized AOI capabilities, enabling higher accuracy and real-time analysis. Despite these advancements, challenges remain in validating models with real-world data and addressing limitations in handling untrained datasets [7].

The advent of deep learning techniques has revolutionized computer vision, enhancing the capabilities of AOI systems. Recent studies, such as those utilizing the Skip-Connected Convolutional Autoencoder model, have achieved defect detection accuracies exceeding 90%. However, these models still face challenges, including the need for real-world validation and limitations in handling untrained data [8]. Additionally, semi-supervised learning (SSL) models that leverage both labeled and unlabeled data with different augmentations have shown improved performance over purely Supervised Learning models, with error increases of less than 0.5% [9]. Despite differences in datasets and object classification, this research provides valuable insight that inform the development of more advanced algorithmic models, such as those explored in this research.

This research examines two advanced deep learning algorithm, faster R-CNN and YOLOv8, for PCB defect detection. Faster R-CNN employs a region proposal network (RPN) for accurate object recognition [10], while YOLOv8, the latest iteration in the YOLO series, excels in real-time detection with high speed and precision [11]. A bibliometric analysis of 105 journal articles indexed by Google Scholar and Scopus further highlights the importance of these algorithm in advancing deep learning applications for PCB defect detection [12]. As shown in Figure 1, “deep learning,” “YOLOv8,” and “faster R-CNN” are central nodes in this field, demonstrating their pivotal contributions to real-time applications and efficient image processing.

This research aims to analyze and compare the performance of faster R-CNN and YOLOv8 in detecting various types of PCB defects, such as missing holes, mouse bites, open circuits, shorts, spurs, and spurious copper. By doing so, this research contributes to the development of efficient and effective deep learning algorithms for PCB defect identification, offering solutions to the increasing complexity of visually undetectable faults. Furthermore, it provides valuable insights into the application of computer vision and deep learning in industrial settings, paving the way for more advanced detection systems capable of handling diverse PCB defects.

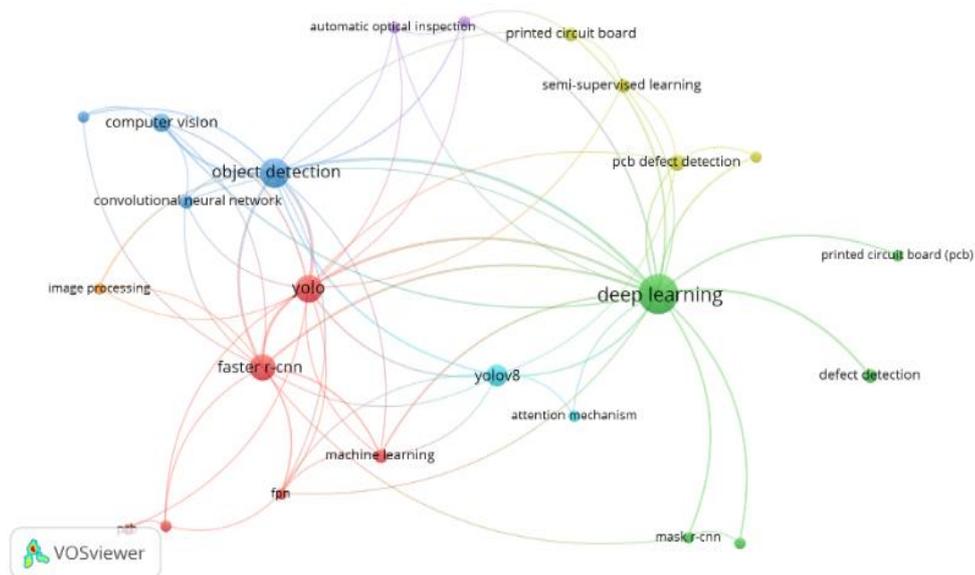


Figure 1. Analysis network visualization VOSviewer

## 2. OVERVIEW OF SELECTED ALGORITHM

### 2.1. Faster R-CNN algorithm

One model or method that can be used to detect objects in an image is the faster region-based convolutional neural network or what is known as faster R-CNN. This model is a development of the previous algorithm, namely fast R-CNN. The goal of faster R-CNN is to increase the accuracy of object detection by reducing the number of proposal regions generated [13]. To achieve this goal, this algorithm

replaces the selective search method in the fast R-CNN model with a region proposal network. The faster R-CNN model is generally similar to the fast R-CNN model but only changes part of the proposal region [14]. Region proposal network (RPN) works through the following steps:

- Using a  $3 \times 3$  convolutional layer with padding 1 to change the CNN output into a new output with a number of channels.
- Next, several labeled boxes are created for each pixel in the feature map with different scales and aspect ratios.
- For each labeled box, binary class prediction (object or background) is performed, and the feature vector is placed at the center of each labeled box.
- After the non-maximum suppression process, the bounding box predicted as an object is used as a proposal region.

The classification loss  $L_{cls}$  and regression loss  $L_{reg}$  are computed as (1), (2):

$$L_{cls}(p_i, p_i^*) = -\log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \quad (1)$$

$$L_{reg}(t_i, t_i^*) = \text{smooth}_{L1}(t_i - t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 \times \frac{1}{\sigma^2} & \text{if } |t_i - t_i^*| < \frac{1}{\sigma^2} \\ |t_i - t_i^*| - 0.5 & \text{otherwise} \end{cases} \quad (2)$$

The position vector  $t_i$ , represents the projected offset of the anchor in the region proposal network. The practical offset,  $t_i^*$  is a vector of the same dimensions as  $t_i$  and is compared with the ground truth. The total loss function is computed with  $L_{cls}$  and  $L_{reg}$ , as (7) shown [15].

$$L_{total}(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (3)$$

It is important to note that the region proposal network is trained simultaneously with other components of the faster R-CNN algorithm model. The region proposal network was developed to produce high quality region proposals, so that it can detect objects with a smaller amount of data. The architecture of the faster R-CNN algorithm is illustrated in Figure 2.

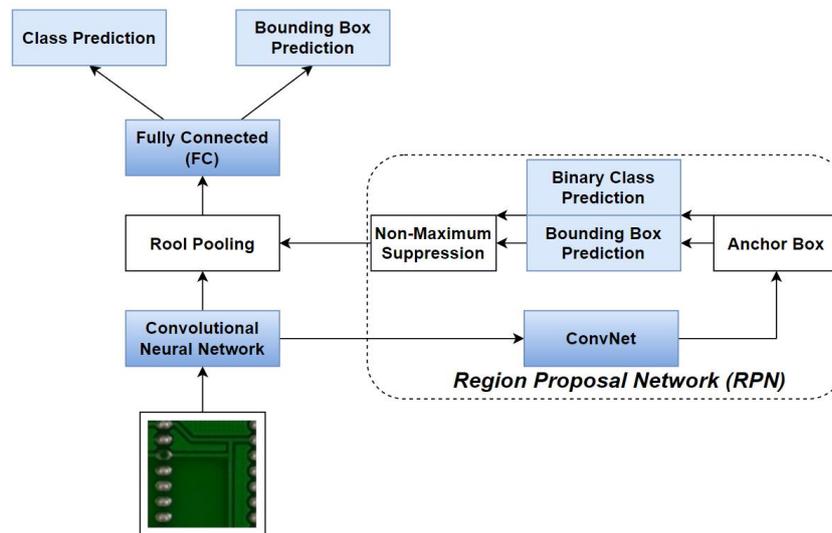


Figure 2. Architecture faster R-CNN algorithm

## 2.2. YOLOv8 algorithm

YOLOv8 is a significant breakthrough in the YOLO series, noted for its real-time object detection capabilities. The model architecture is composed of four primary components: the input module, backbone feature extraction network, neck network, and detection module. These components work together seamlessly to process images, extract features, and forecast object classes and bounding boxes with great efficiency, making YOLOv8 a solid solution for many object recognition workloads [16].

The input module of YOLOv8 covers crucial functionality such as picture input, data augmentation, and adaptive anchor box computations. The backbone feature extraction network leverages structures like Conv+Bn+SiLU (CBL), CSPLayer\_2Conv (c2F), and spatial pyramid pooling-fast (SPPF) to extract meaningful features from images. These structures enhance the model's ability to recognize and classify objects accurately. Meanwhile, the neck network employs the path aggregation network (PAN) structure to fuse object information across different scales. This approach makes YOLOv8 highly effective in detecting objects of varying sizes by ensuring that feature maps from multiple layers are utilized optimally. Finally, the detection module combines classification and regression tasks by using advanced loss functions such as varifocal loss (VFL) for classification and distribution focal loss (DFL) with complete intersection over union (CIoU) loss for regression, resulting in precise and reliable object detection [17].

In addition to its architectural advantages, YOLOv8 offers flexibility in its configurations, allowing users to adjust parameters such as width, depth, and ratio to meet specific performance and computational requirements. The model is available in five configurations, including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x each catering to different application needs [18]. Evaluations have shown that YOLOv8m achieves the highest mAP at 87.72%, while YOLOv8s provides a balance between performance and efficiency with a smaller file size and fewer layers. These characteristics make YOLOv8 suitable for a wide range of scenarios, from resource-constrained environments to high-performance applications. The overall architecture of the YOLOv8 model, including its backbone, neck, and head components, is illustrated in Figure 3.

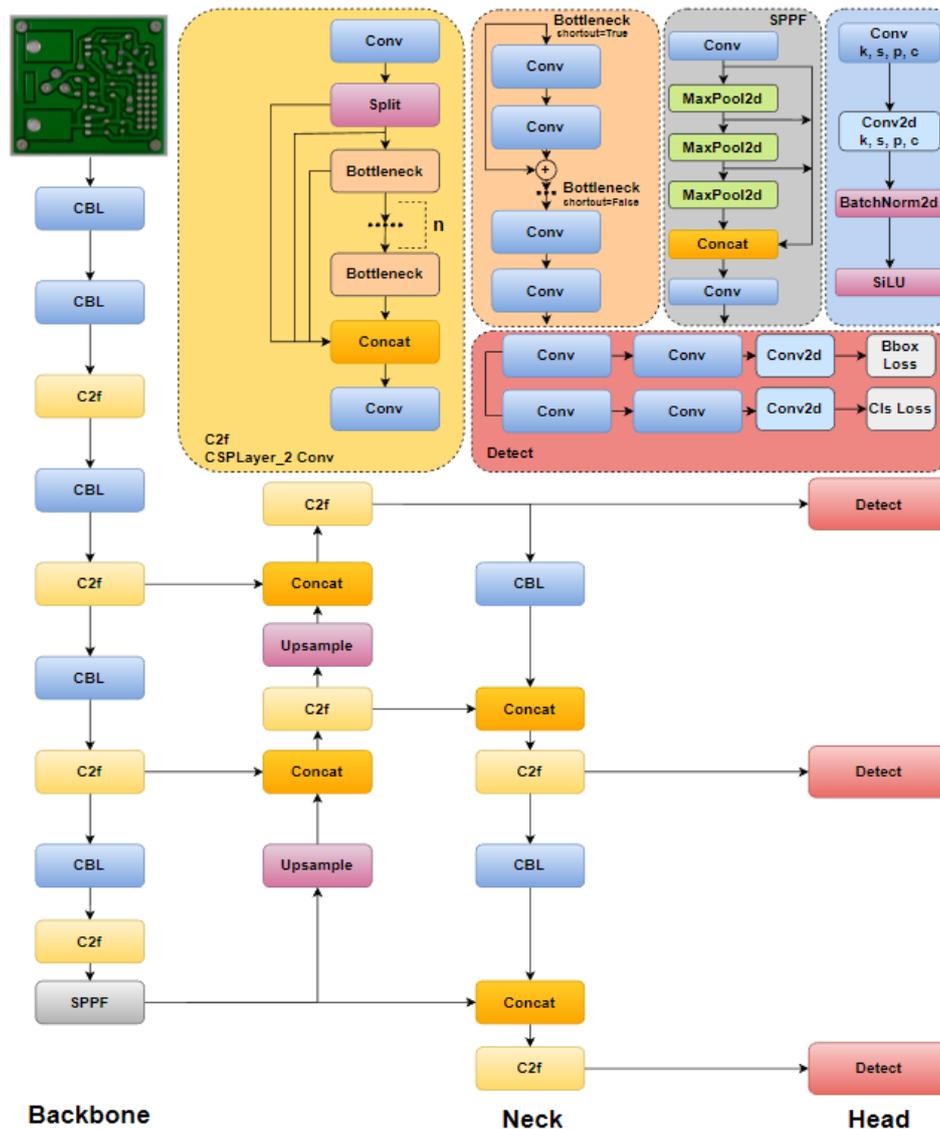


Figure 3. Architecture YOLOv8 algorithm

### 2.3. Selecting hyperparameter

To train the defect recognition system on PCBs, we applied two dataset approaches: non-augmented and augmented datasets. A key stage in the model construction process before training is setting the hyperparameters, as these can considerably affect the pace of training convergence and the quality of the resulting model. Table 1 displays the selected hyperparameters utilized in training the faster R-CNN and YOLOv8 algorithms in this research.

In addition to the selected hyperparameters, this research utilized pre-trained weights for model initialization, which improved both training efficiency and accuracy in defect detection. The faster R-CNN models employed ResNet50 FPN backbones, while YOLOv8 included five configurations ranging from YOLOv8n (6.2 MB) to YOLOv8x (131 MB). Among these, YOLOv8x demonstrated superior performance due to its higher model capacity and optimized weight configurations. By selecting relevant hyperparameters and applying pre-trained weights, we strive to improve both the training speed and the accuracy of the model, ensuring the highest possible performance in defect detection on PCB layouts.

Table 1. Setting hyperparameter algorithm

Hyperparameter	Faster R-CNN	YOLOv8
Input image size	640×640 (px)	640×640 (px)
Epochs	50	100
Learning rate	0.001	0.01
Batch size	8	16
Workers	4	8
Device	Cuda	Cuda
Weights decay	Null	0.0005

## 3. RESEARCH METHOD

### 3.1. The research stage

This research was undertaken through a series of stages meant to test the efficacy of the faster R-CNN and YOLOv8 algorithms in finding defects in PCB layouts. The research process is summarized in the flowchart shown in Figure 4. The approach begins with a detailed literature research, focusing on both defect detection utilizing faster R-CNN and YOLOv8 algorithms and generic defect identification techniques for PCB layouts. This foundational review was critical for comprehending the core theories and related advancements, directing the ensuing experimental design [19].

The dataset for this research, sourced from Huang *et al.* [20], consist of 10 PCBs with 693 images, representing six defect types, including missing hole, mouse bite, short, open circuit, spur, and spurious copper. Images were manually labeled using Roboflow tools, generating .xml files for faster R-CNN and .txt files for YOLOv8. The labeled data was divided into training (85%), validation (10%), and testing (5%) subsets. Data augmentation techniques, such as 90-degree rotations, small-angle rotations (-15° to +15°), and horizontal/vertical shears, were applied to enhance dataset diversity and reduce overfitting. Separate training sessions were conducted for Faster R-CNN and YOLOv8 using augmented and non-augmented datasets. Model performance was evaluated using a confusion matrix, focusing on key metrics like precision, recall, F1-score, IoU, and mAP. These evaluations, combined with training times and detection speeds, provided a thorough assessment of each algorithm's capabilities for PCB defect detection [21], [22].

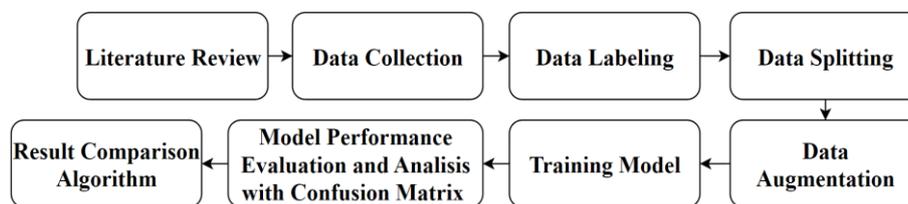


Figure 4. The procedure research

### 3.2. Dataset collection

In this research, the dataset for defect detection in PCB layouts comprises 693 images captured with a 16-megapixel HD industrial camera equipped with a complementary metal-oxide-semiconductor (CMOS) sensor. The original resolution of 4608×3456 pixels was resized to 640×640 pixels to standardize input size

for each method [20]. This resolution was chosen based on preliminary experiments, balancing computational efficiency with the preservation of essential features for accurate defect detection. The dataset includes six defect types, as summarized in Table 2, the images are distributed evenly across defect classes, encompassing 2,593 defect points. This balanced distribution is crucial for training a robust model capable of generalizing effectively across all defect types.

Table 2. Distribution of number of images and defect points of HRIPC dataset

Reference PCB	Adjusted size	Defect type					
		Missing hole	Mouse bite	Open circuit	Short	Spur	Spurious copper
Image 1	640×640	20	20	20	20	20	20
Image 4	640×640	20	20	20	20	20	20
Image 5	640×640	10	10	10	10	10	10
Image 6	640×640	10	10	10	10	10	10
Image 7	640×640	10	10	10	10	10	10
Image 8	640×640	10	10	10	10	10	10
Image 9	640×640	10	10	10	10	10	10
Image 10	640×640	5	5	6	6	5	6
Image 11	640×640	10	10	10	10	10	10
Image 12	640×640	10	10	10	10	10	10
Number image (amount of defect)		115 (497)	115 (492)	116 (482)	116 (491)	115 (488)	116 (503)
				693 (2593)			

### 3.3. Overall system configuration

The pipeline of the PCB defect detection system proposed in this research is illustrated in Figure 5. The process begins with pre-processing, where the PCB dataset is standardized using an auto-orientation technique to align all images according to defined standards [23]. The images are then resized to 640×640 pixels, balancing computational efficiency with the resolution required of accurate defect detection. Data augmentation techniques, such as rotations and shearing, are applied to enhance dataset diversity, simulating real-world variations in PCB manufacturing and inspection [24].

Following pre-processing and augmentation, the dataset is used to train two deep learning algorithms, faster R-CNN and YOLOv8. Both algorithms are trained on augmented and non-augmented datasets to evaluate their performance under different conditions. This approach provides insights into each model's strengths and limitations in handling data variability. Once trained, the models predict defects in PCB layouts, generating a defect detection map. This map highlights detected defects based on predefined thresholds, clearly indicating whether a PCB is defective or normal. Such outputs are critical for practical applications, ensuring quick and reliable defect identification in real-world scenarios [25].

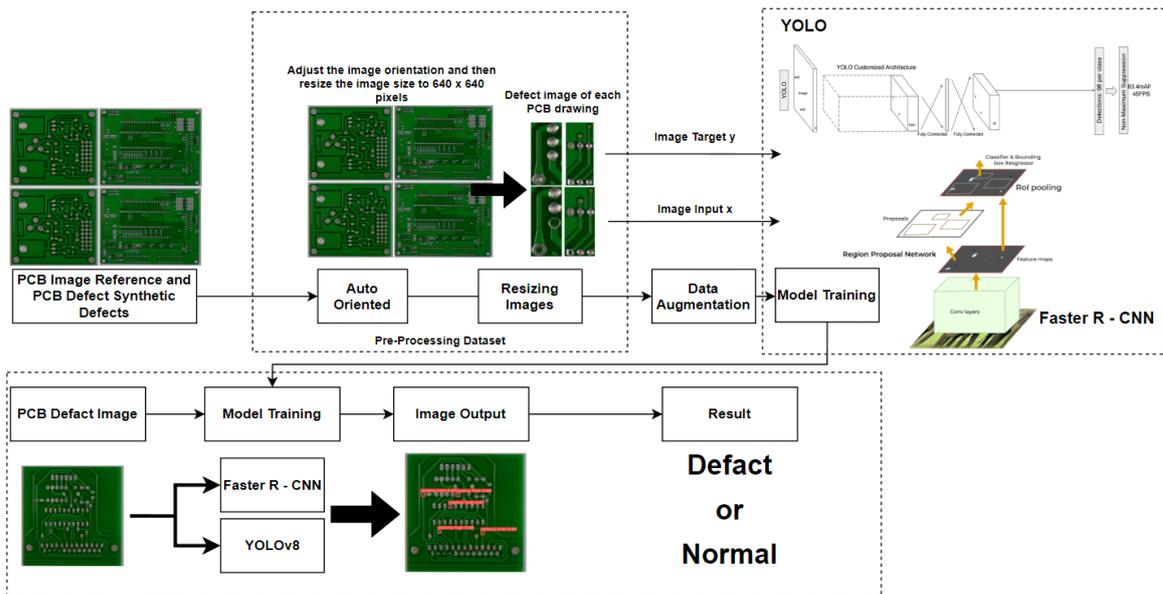


Figure 5. Overall system configuration

### 3.4. Pre-processing dataset

Data preparation was undertaken with two primary objectives, to correct image orientation and to normalize image dimension. The “*Auto-Orient*” feature was utilized to automatically fix the orientation based on EXIF metadata, ensuring uniform visual alignment across all photos. Subsequently, all photographs were downsized to 640×640 pixels using the “*Resize Image*” tool, standardizing their proportions for the training process. These preparation steps are critical for ensuring that the dataset is consistent and suitable for model training. Proper orientation and size uniformity are critical for delivering accurate and dependable training outcomes [26].

Step 1: Automatically correct image orientation, the *Auto-Orient* function identifies and corrects image orientation based on EXIF metadata to ensure visual consistency. The image is rotated as required based on the recorded orientation of 180°, 270°, and 90°.

Step 2: Normalize image dimensions, all images in the dataset are resized to 640×640 pixels using the function *Resize Image*, to ensure size uniformity before training the neural network model.

### 3.5. Dataset augmentation

Training neural network models often requires large-scale datasets due to the large complexity of model parameters [27]. However, product defects during the manufacturing process tend to be rare and the variety of defect types may change during mass production. In this research, the use of un-augmented and augmented datasets is taken into consideration to determine the level of performance and influence with these two methods. We apply augmentation techniques, 90° rotate (clockwise, counter-clockwise, upside down), rotation (between -15° and +15°), and shear (±10° horizontal, ±10° vertical). Clockwise, each image is rotated 90° clockwise to simulate a change in orientation. Counter-clockwise, a similar process is performed but in the opposite direction to add orientation variation. Upside down, the image is rotated by 180° to fully create the best image conditions, adding more orientation variations. Then, random rotation between -15° and +15° to simulate positional imperfections that may occur during the production or inspection process. Furthermore, shear augmentation is carried out to simulate the effects of pressure or tension on the PCB which can affect the shape or relative position of components and defects.

### 3.6. Dataset splitting

Following preprocessing and data augmentation, the dataset was separated into training, validation, and testing sets. The split was accomplished proportionally, 85% for training, 10% for validation, and 5% for testing. Table 3 shows the distribution of photos and defects across these subsets for both augmented and non-augmented dataset. The proportion 85/10/5 split is a generally established method in machine learning, giving adequate data for training while guaranteeing that the validation and testing sets are representative of the whole dataset. This divide supports robust model evaluation and reduces the possibility of overfitting.

Table 3. Splitting non-augmented dataset and augmented dataset

	Non-augmentation		Augmentation	
	Number of images	Number of defects	Number of images	Number of defects
Train	589	2148	1767	7523
Valid	69	293	69	293
Test	35	152	35	152
Total	693	2593	1871	7968

### 3.7. Performance evaluation

The performance of the defect detection models was evaluated using a confusion matrix, which provides insights into the models’ predictions across six defect classes, including missing hole, mouse bite, short, open circuit, spur, spurious copper. The confusion matrix is a table with four cells that show the number of accurate and erroneous guesses for each item type. The confusion matrix has cells for true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

Key metrics such as mean average precision (mAP), precision, recall, and F1-score were derived from the confusion matrix. Precision measures the proportion of correctly identified positive instances, while recall evaluates the model’s ability to detect all relevant positive instances. The F1-score combines precision and recall to provide a balanced assessment of the model’s performance. Additionally, mAP utilizes the intersection over union (IoU) to evaluate the similarity between predicted and ground truth bounding boxes [28].

$$mAP@α = \frac{1}{n} \sum_{i=1}^n AP_i \text{ for } n \text{ classes} \quad (4)$$

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

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

$$F1 - Score = \frac{2*(precision*recall)}{(precision+recall)} \quad (7)$$

## 4. RESULTS AND DISCUSSION

### 4.1. Perform result with non-augmentation dataset

The objective of this investigation is to ascertain the optimal models and procedures for un-augmented or unmodified datasets. Table 4 show as summarized depicts the performance analysis outcomes of both algorithms on various model variations utilizing an unaltered dataset methodology. The examined metrics consist of mAP@50, mAP@50:95, precision, recall, and F1-Score for each setup of the model.

Table 4. Best perform model with non-augmentation dataset

Model	Varian	mAP@50	mAP@50:95	Precision	Recall	F1-Score
Faster R-CNN	<i>ResNet50 FPN</i>	0.755	0.302	0.216	0.326	0.257
	<i>ResNet50 FPN v2</i>	0.763	0.325	0.243	0.346	0.283
	<i>v8n</i>	0.873	0.418	0.884	0.848	0.866
	<i>v8s</i>	0.924	0.461	0.951	0.894	0.921
YOLO	<i>v8m</i>	0.937	0.486	0.954	0.901	0.927
	<i>v8l</i>	0.963	0.518	0.946	0.925	0.935
	<i>v8x</i>	0.949	0.491	0.956	0.920	0.938

Faster R-CNN with the ResNet50 FPN v2 backbone achieved a mAP@50 of 0.763 and a mAP@50:95 of 0.325, making it the best-performing variant among Faster R-CNN models. In contrast, the YOLOv8 models consistently outperformed faster R-CNN, with YOLOv8l achieving the highest mAP@50 of 0.963 and mAP@50:95 of 0.518. These results emphasize YOLOv8l superior ability to detect PCB defects, showcasing higher accuracy and robustness across all metrics.

The findings align with prior research, such as Yusro *et al.* [29], which highlighted YOLO-based models as the preferred choice for real-time object detection due to their speed and precision. This study extends those findings, demonstrating YOLOv8l notable advantages in generalization, particularly for detecting small defects. The results underscore YOLOv8l efficiency and suitability for PCB surface flaw detection tasks, especially in scenarios requiring high precision and speed.

### 4.2. Perform result with augmentation dataset

After analyzing the performance on the dataset without augmentation, it is crucial to examine how the two algorithms respond to the conditions of an expanded dataset. This comparison offers vital insights into the resilience and adaptability of each algorithm under increasingly complicated and varied data circumstances. Table 5 show the performance of both algorithms when assessed on the augmented dataset, using the same performance assessment metrics as in the non-augmented dataset research.

Table 5. Best perform model with augmentation dataset

Model	Varian	mAP@50	mAP@50:95	Precision	Recall	F1-Score
Faster R-CNN	<i>ResNet50 FPN</i>	0.754	0.275	0.286	0.372	0.323
	<i>ResNet50 FPN v2</i>	0.826	0.338	0.316	0.411	0.357
	<i>v8n</i>	0.914	0.434	0.948	0.841	0.892
	<i>v8s</i>	0.927	0.488	0.939	0.906	0.922
YOLO	<i>v8m</i>	0.949	0.481	0.949	0.916	0.932
	<i>v8l</i>	0.951	0.480	0.954	0.909	0.931
	<i>v8x</i>	0.962	0.503	0.953	0.945	0.949

On the augmented dataset, faster R-CNN with the ResNet50 FPN v2 backbone achieved its highest mAP@50 score of 0.826 and mAP@50:95 of 0.338, indicating an improvement compared to its performance on non-augmented data. This demonstrates that augmentation enhances its ability to detect defects on PCB surfaces. Meanwhile, YOLOv8x showed the best overall performance, achieving a mAP@50 of 0.962 and

mAP@50:95 of 0.503, further solidifying its position as the most effective model for fault detection on PCB layouts. The YOLOv8x model outperformed faster R-CNN across all metrics, showcasing superior robustness and versatility when trained on heterogeneous data. The significant improvement in mAP@50:95 highlights YOLOv8x ability to generalize better, particularly for detecting smaller defects. These findings are consistent with previous studies, such as Yao *et al.* [30], which emphasized the importance of augmentation in improving the performance of deep learning models for object detection.

### 4.3. Analysis comparison algorithm

This section compares the faster R-CNN and YOLOv8 algorithms, focusing on their speed and accuracy in detecting PCB defects. Faster R-CNN employs a two-stage approach with a RPN, which is effective in identifying small and complex objects but results in slower performance. Training times for Faster R-CNN are recorded at 1.85 seconds per iteration and 408.85 seconds per epoch due to its computationally intensive architecture. In contrast, YOLOv8 utilizes a single-shot detection approach, predicting bounding boxes and object probabilities directly, significantly reducing training times to 1.13 seconds per iteration and 113 seconds per epoch. This highlights YOLOv8 efficiency and suitability for real-time applications.

The impact of dataset augmentation on model performance is evident, as seen in Figure 6. While the non-augmented YOLOv8l achieves a precision of 0.946, recall of 0.925, and F1-score of 0.935, the augmented YOLOv8x surpasses it with metrics of 0.953, 0.945, and 0.949, respectively. This demonstrates the significance of augmentation in improving model performance. Consistent with Yao *et al.* [30], these findings highlight the augmented YOLOv8x as the most effective configuration, combining superior accuracy with processing efficiency, particularly for detecting faults on PCB layouts. In Figure 7, the augmented YOLOv8x accurately identifies defects such as spurious copper and open circuits with high confidence, outperforming both the non-augmented YOLOv8l and faster R-CNN (ResNet50 FPN v2) in detection accuracy. These results emphasize the critical role of dataset augmentation in enhancing the robustness and reliability of deep learning models for defect detection.

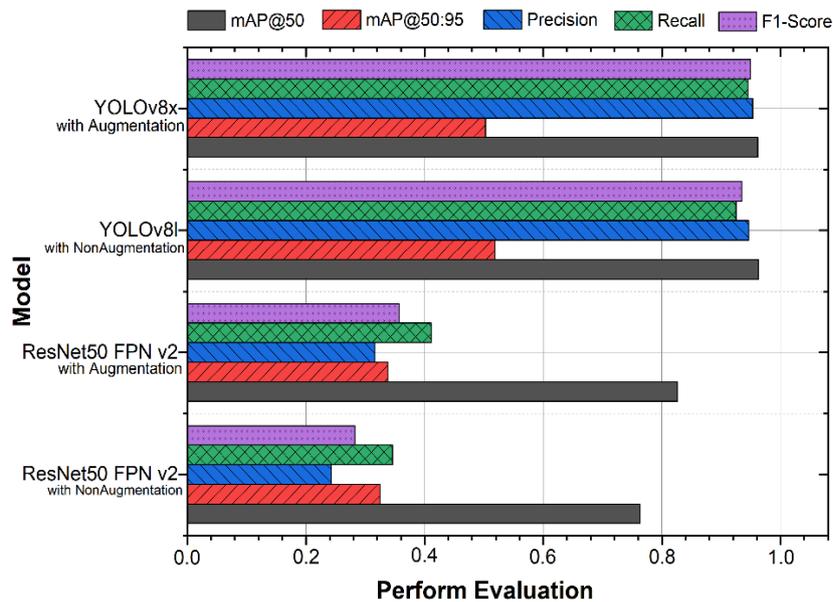


Figure 6. Analysis result comparison algorithm

In conclusion, YOLOv8x demonstrates superior performance in both speed and accuracy, making it the most robust and efficient solution for real-time PCB defect detection. Future research could focus on developing hybrid models that combine the precision of two-stage detectors like faster R-CNN with the computational efficiency of single-stage detectors like YOLOv8. Additionally, applying YOLOv8x to detect other defect types or operate in diverse industrial contexts would validate its versatility further. Evaluating these algorithms on edge devices or in distributed computing environments could unlock new opportunities for real-time defect detection in industries requiring low-latency processing.

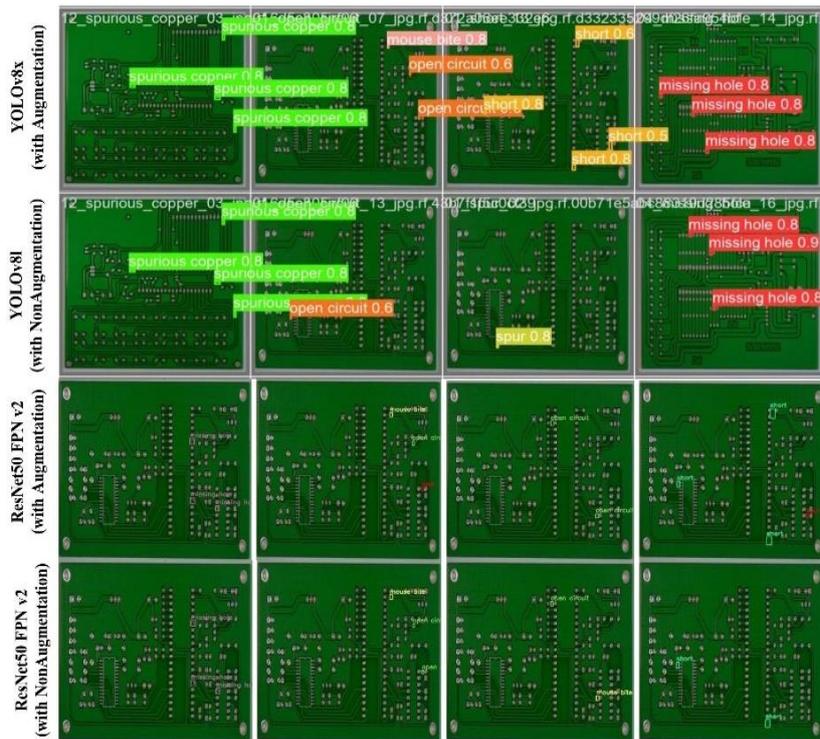


Figure 7. Detection results of advanced object detection algorithms

## 5. CONCLUSION

This research successfully developed a robust defect detection system for PCB layouts using two advanced deep learning algorithms, faster R-CNN and YOLOv8. Through evaluations on non-augmented and augmented datasets, the study highlighted their respective strengths and limitations. Faster R-CNN, particularly the ResNet50 FPN v2 variant, showed improved performance with augmented data but was limited by its slower two-stage detection process, making it less suitable for real-time applications. In contrast, YOLOv8, especially the YOLOv8x variant, consistently outperformed faster R-CNN in both speed and accuracy. Its single-shot detection mechanism enabled faster computation while maintaining high accuracy, positioning YOLOv8x as the optimal choice for real-time PCB defect detection.

The findings underscore the significance of data preparation in enhancing deep learning model robustness. YOLOv8x superior performance makes it a practical solution for improving quality control processes in industrial inspection systems, offering enhanced accuracy and speed for real-time applications. Future research could focus on developing hybrid models that combine the precision of faster R-CNN with the efficiency of YOLOv8 to balance accuracy and computational cost. Expanding the application of these models to diverse defect types and industrial settings would further validate their versatility. Additionally, deploying these algorithms on edge devices or distributed systems could unlock new opportunities for real-time defect detection in low-latency environments.

## ACKNOWLEDGMENTS

We express our sincere gratitude to all parties who contributed to the successful completion of this research. Our deepest appreciation goes to the Department of Electrical Engineering Education, Faculty of Industrial Technology and Education, Universitas Pendidikan Indonesia, for providing the necessary resources and institutional support throughout this study. Lastly, we acknowledge the efforts of the journal's editorial team and publishers in facilitating the dissemination of our findings, ensuring this research serves as a valuable reference for the academic community.

## FUNDING INFORMATION

No external funding was involved in this research. The research was self-funded, with financial support allocated from the professorial honorarium to fulfill the special duties of a professor.

### AUTHOR CONTRIBUTIONS STATEMENT

The contributions of each author to this research are detailed in accordance with the CRediT (Contributor Roles Taxonomy) framework. Each author played a distinct role in various aspects of the research, including conceptualization, methodology, investigation, data curation, and manuscript preparation. The specific contributions of each author are outlined in the following table:

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Jaja Kustija	✓	✓		✓	✓	✓				✓		✓		✓
Diki Fahrizal	✓	✓	✓				✓	✓	✓		✓		✓	
Muhamad Nasir				✓		✓	✓			✓				
Andi Adriansyah	✓	✓		✓		✓				✓	✓			
Muhammad Husni Muttaqin			✓				✓	✓	✓				✓	

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : **O**riting - **O**riginal Draft

E : **E**riting - **R**eview & **E**ditng

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

### CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known financial, personal, or professional conflicts of interest that could have influenced the work reported in this paper. There are no competing interests related to political, ideological, religious, academic, or intellectual affiliations.

### INFORMED CONSENT

This study did not involve human participants or any identifiable personal data. Therefore, obtaining informed consent was not applicable.

### ETHICAL APPROVAL

This research did not involve human participants or animals. Therefore, ethical approval was not required.

### DATA AVAILABILITY

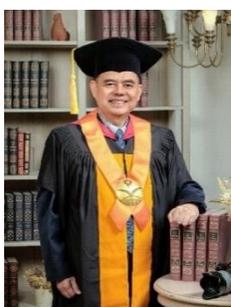
The data that support the findings of this research are available from the corresponding author upon reasonable request. Interested researchers may contact the corresponding author as documented in the research documentation or via email at [jaja.kustija@upi.edu](mailto:jaja.kustija@upi.edu), [diki15@upi.edu](mailto:diki15@upi.edu), or [23224321@mahasiswa.itb.ac.id](mailto:23224321@mahasiswa.itb.ac.id). To request access, please include a clear subject line and specify the purpose of obtaining the full research data.

### REFERENCES

- [1] M. Calabrese, L. Agnusdei, G. Fontana, G. Papadia, and A. Del Prete, "Application of mask R-CNN for defect detection in printed circuit board manufacturing." Nov. 16, 2023, doi: 10.21203/rs.3.rs-3491517/v1.
- [2] S. Oh, M. Jung, C. Lim, and S. Shin, "Automatic detection of welding defects using faster R-CNN," *Applied Sciences*, vol. 10, no. 23, p. 8629, Dec. 2020, doi: 10.3390/app10238629.
- [3] A. Legon, M. Deo, S. Albin, and M. Audette, "Detection and classification of PCB defects using deep learning methods," in *2022 IEEE 31st Microelectronics Design & Test Symposium (MDTS)*, May 2022, pp. 1–6, doi: 10.1109/MDTS54894.2022.9826925.
- [4] Y. T. Li and J. I. Guo, "A VGG-16 based faster RCNN model for PCB error inspection in industrial AOI applications," in *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, May 2018, pp. 1–2, doi: 10.1109/ICCE-China.2018.8448674.
- [5] O. Y. Chiun and N. I. R. Ruhaiyem, "Object detection based automated optical inspection of printed circuit board assembly using deep learning," *Communications in Computer and Information Science*, 2023, pp. 246–258, doi: 10.1007/978-981-99-0405-1\_18.
- [6] A. Caliskan and G. Gurkan, "Design and realization of an automatic optical inspection system for PCB solder joints," in *2021 International Conference on Innovations in Intelligent Systems and Applications (INISTA)*, Aug. 2021, pp. 1–6, doi: 10.1109/INISTA52262.2021.9548430.
- [7] Y. Yang *et al.*, "A high-performance deep learning algorithm for the automated optical inspection of laser welding," *Applied Sciences*, vol. 10, no. 3, p. 933, Jan. 2020, doi: 10.3390/app10030933.
- [8] J. Kim, J. Ko, H. Choi, and H. Kim, "Printed circuit board defect detection using deep learning via a skip-connected convolutional

- autoencoder,” *Sensors*, vol. 21, no. 15, p. 4968, Jul. 2021, doi: 10.3390/s21154968.
- [9] 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, p. 3246, Mar. 2023, doi: 10.3390/s23063246.
- [10] M. Arulmozhi, N. G. Iyer, S. Jeny Sophia, P. Sivakumar, C. Amutha, and D. Sivamani, “Comparison of YOLO and faster R-CNN on garbage detection,” in *Optimization Techniques in Engineering*, Wiley, 2023, pp. 37–49.
- [11] M. Maity, S. Banerjee, and S. Sinha Chaudhuri, “Faster R-CNN and YOLO based vehicle detection: a survey,” in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Apr. 2021, pp. 1442–1447, doi: 10.1109/ICCMC51019.2021.9418274.
- [12] T. Jian, Y. Yang, H. Baoshuai, and H. Chongqing, “PCB defect detection algorithm based on YT-YOLO,” in *2023 35th Chinese Control and Decision Conference (CCDC)*, May 2023, pp. 976–981, doi: 10.1109/CCDC58219.2023.10326719.
- [13] A. Malini, P. Priyadarshini, and S. Sabeena, “An automatic assessment of road condition from aerial imagery using modified VGG architecture in faster-RCNN framework,” *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 6, pp. 11411–11422, Jun. 2021, doi: 10.3233/JIFS-202596.
- [14] 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.
- [15] D. G. D. S. Vaishnav, and R. Mohan, “A faster R-CNN implementation of presence inspection for parts on industrial produce,” in *2021 Emerging Trends in Industry 4.0 (ETI 4.0)*, May 2021, pp. 1–4, doi: 10.1109/ETI4.051663.2021.9619228.
- [16] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-N González, “A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS,” *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
- [17] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU loss: faster and better learning for bounding box regression,” *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, pp. 12993–13000, 2020, doi: 10.1609/aaai.v34i07.6999.
- [18] R. Ren *et al.*, “Intelligent detection of lightweight ‘Yuluxiang’ pear in non-structural environment based on YOLO-GEW,” *Agronomy*, vol. 13, no. 9, p. 2418, Sep. 2023, doi: 10.3390/agronomy13092418.
- [19] T. Sucita, D. L. Hakim, R. H. Hidayatulloh, and D. Fahrizal, “Solar irradiation intensity forecasting for solar panel power output analyze,” *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 36, no. 1, pp. 74–85, Oct. 2024, doi: 10.11591/ijeecs.v36.i1.pp74-85.
- [20] W. Huang, P. Wei, M. Zhang, and H. Liu, “HRIPCB: a challenging dataset for PCB defects detection and classification,” *The Journal of Engineering*, vol. 2020, no. 13, pp. 303–309, Jul. 2020, doi: 10.1049/joe.2019.1183.
- [21] J. Kustija, D. Fahrizal, M. Nasir, D. Setiawan, and I. Surya, “Design and development of coastal marine water quality monitoring based on IoT in achieving implementation of SDGs,” *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 36, no. 3, pp. 1470–1484, Dec. 2024, doi: 10.11591/ijeecs.v36.i3.pp1470-1484.
- [22] M. A. H. Akbar, D. Fahrizal, J. Kustija, and I. Surya, “Digital technology integration in TVET for tourism: a case study for an android-based application development and implementation,” in *2024 9th International STEM Education Conference (iSTEM-Ed)*, Jul. 2024, pp. 1–6, doi: 10.1109/iSTEM-Ed62750.2024.10663108.
- [23] H. Kiya, T. Nagamori, S. Imaizumi, and S. Shiota, “Privacy-preserving semantic segmentation using vision transformer,” *Journal of Imaging*, vol. 8, no. 9, p. 233, Aug. 2022, doi: 10.3390/jimaging8090233.
- [24] P. Ghose, A. Ghose, D. Sadhukhan, S. Pal, and M. Mitra, “Improved polyp detection from colonoscopy images using finetuned YOLO-v5,” *Multimedia Tools and Applications*, vol. 83, no. 14, pp. 42929–42954, Oct. 2023, doi: 10.1007/s11042-023-17138-3.
- [25] Ö. Kaya, M. Y. Çodur, and E. Mustafaraj, “Automatic detection of pedestrian crosswalk with faster R-CNN and YOLOv7,” *Buildings*, vol. 13, no. 4, p. 1070, Apr. 2023, doi: 10.3390/buildings13041070.
- [26] A. Fava *et al.*, “Pre-processing techniques to enhance the classification of lung sounds based on deep learning,” *Biomedical Signal Processing and Control*, vol. 92, p. 106009, Jun. 2024, doi: 10.1016/j.bspc.2024.106009.
- [27] Y. Ni, J. Huo, Y. Hou, J. Wang, and P. Guo, “Detection of underground dangerous area based on improving YOLOV8,” *Electronics*, vol. 13, no. 3, p. 623, Feb. 2024, doi: 10.3390/electronics13030623.
- [28] R.-Y. Ju and W. Cai, “Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm,” *Scientific Reports*, vol. 13, no. 1, p. 20077, Nov. 2023, doi: 10.1038/s41598-023-47460-7.
- [29] M. M. Yusro, R. Ali, and M. S. Hitam, “Comparison of faster R-CNN and YOLOv5 for overlapping objects recognition,” *Baghdad Science Journal*, vol. 20, no. 3, p. 0893, Jun. 2023, doi: 10.21123/bsj.2022.7243.
- [30] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, “A real-time detection algorithm for kiwifruit defects based on YOLOv5,” *Electronics*, vol. 10, no. 14, p. 1711, Jul. 2021, doi: 10.3390/electronics10141711.

## BIOGRAPHIES OF AUTHORS



**Jaja Kustija**    received a bachelor's degree in electrical engineering education at the Indonesian University of Education in 1984, received a master's degree M.Sc. in the field of instrumentation physics at the Bandung Institute of Technology in 1992, and received his doctorate in technology and vocational education at the Indonesian University of Education in 2012. The research and publications carried out are very experienced both on a national scale indexed by SINTA and on an international scale indexed by Scopus, and have research umbrellas with other institutions or with students with major topics design, manufacture, testing monitoring and remote control and learning media based on electronics and the internet of things. Books that have been published by one of the national scale publishers include; electromagnetic field theory and applications; mechatronics books for colleges; electromagnetic field with required engineering mathematics enrichment; electrical circuit 1 Equipped with MATLAB for calculation operations. Now he is a professor at the Indonesian University of Education in the field of electrical and mechatronic engineering education. He can be contacted at email: jaja.kustija@upi.edu.



**Diki Fahrizal**    is a graduate of electrical engineering education from the Department of Electrical Engineering Education, Faculty of Technology and Vocational Studies, Universitas Pendidikan Indonesia (2024). Currently, he is pursuing a master's degree in electrical engineering at Institut Teknologi Bandung, specializing in control engineering and intelligent systems. He is actively engaged in collaborative research with lecturers, focusing on areas such as the internet of things (IoT), artificial intelligence (AI), computer vision, machine learning, data mining, and virtual reality/augmented reality (VR/AR) technologies. His dedication to research and innovation is driven by a passion for solving real-world problems through advanced technological solutions. He can be contacted at email: 23224321@mahasiswa.itb.ac.id.



**Muhamad Nasir**    received a bachelor's degree in physical chemistry at Andalas University, received a master's degree in membrane at Bandung Institute of Technology, post-doctoral in nanofiber at Kyoto Institute of Technology, and obtained a Dr. Eng. in composite nanomaterial and electrospinning, nano membrane at Tokyo Institute of Technology. His main skills are manufacturing, electrospinning, and R&D. His research and publications are very experienced both on a national scale indexed by SINTA and on an international scale indexed by Scopus. Currently, he is a Founder and Chairman at Tarifa Institute of Science Technology (TIST) (2023 - Present), Consultant at PT. TARIFA INDONESIA (March 2011 - Present), and Researcher at Research Center for Chemistry, Indonesian Institute of Science (November 1997 - Present). He can be contacted at email: muha053@brin.go.id.



**Andi Adriansyah**    is a professor in electrical engineering in Universitas Mercu Buana born in 1970. He completed his undergraduate education in electrical engineering Universitas Indonesia, Indonesia, in 1994. Then, his master's and doctoral education were completed at Universiti Teknologi Malaysia, Malaysia, in 1998 and 2007, respectively. In addition, he conducts some research in mechatronics, robotics, control and automation, artificial intelligence, and the internet of things (IoT). He can be contacted at email: andi@mercubuana.ac.id.



**Muhammad Husni Muttaqin**    a 26-year-old graduate student from Bandung, completed his bachelor's degree in electrical engineering education at Universitas Pendidikan Indonesia and is now pursuing a master's degree in electrical engineering at Institut Teknologi Bandung, specializing in control engineering and intelligent systems. He is completing his thesis on robust control for dynamic positioning systems and is dedicated to developing innovative technological solutions in control and condition monitoring systems using Python programming and microcontrollers. His research interests include creating automation systems with graphical user interfaces using Python and QML, focusing on control systems such as PID, state feedback, LQR, and  $H_\infty$ , integrated into IoT-based systems. He can be contacted at email: 23223303@mahasiswa.itb.ac.id.