# Defect Detection in Printed Circuit Boards: A Comparative Analysis of Object Detection Models with Depthwise Convolution Adaptation

Julio Martins, *Member, IEEE,* Josue Lopez-Cabrejos, *Member, IEEE,* Quefren Leher, *Member, IEEE,* Thuanne Paixão, *Member, IEEE,* Ana Beatriz Alvarez, *Senior Member, IEEE,* and Facundo Palomino-Quispe, *Senior Member, IEEE*

***Abstract*—Printed circuit boards (PCBs) are key components in the electronics industry, and ensuring their integrity is essential for reliable manufacturing. Automated inspection systems based on computer vision, although efficient, face challenges. In this scenario, deep learning techniques have become effective solutions for detecting defects in more modern and complex PCBs. This article presents a comparative study between the YOLOv8n, YOLOv11n and RT-DETRv2 models for identifying defects in PCBs. The experiments were conducted using the PKU-Market-PCB dataset, which includes Missing Hole, Mouse Bite, Open Circuit, Short Circuit, Spur and Spurious Copper defects. To reduce the computational cost, modified versions of YOLOv8n and YOLOv11n with Depthwise convolution blocks (YOLOv8n-DWConv and YOLOv11n-DWConv). The analysis includes quantitative and qualitative comparisons. In addition, the robustness of the models is evaluated under challenging conditions with blur and illumination gradient noise. The results indicate that YOLOv11n achieves the best overall performance, while YOLOv11n-DWConv offers a competitive balance between precision and computational efficiency.**

**Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/9899**

***Index Terms*—Objects Detection; Printed Circuit Boards Surface Defects; You Only Look Once; Depthwise Convolution; Deep Learning.**

## I. Introduction

With more than a century of evolution, printed circuit boards (PCBs) have become fundamental in the electronics industry, offering mechanical support and electrical connections for various components. Their main advantage lies in the optimization of the assembly process, allowing for greater automation and efficiency in manufacturing. With advances in semiconductor design and manufacturing, PCBs have become ultra-thin, dense and multi-layered, enabling greater performance but requiring more rigorous inspections [1].

The quality of PCBs depends on the integrity of tracks and holes, essential elements for the functioning of circuits. Initially, inspection was done manually, but the growing complexity and increased production volume made this method unfeasible, with high costs, low efficiency and difficulty in collecting data. To mitigate these problems, the industry began to adopt automated inspection systems based on computer vision, capable of identifying defects more quickly and precisely [2], [3].

Modern approaches include Automated Optical Inspection (AOI), which uses industrial cameras and image processing algorithms to detect faults such as shorts, gaps and component misalignment [4]–[6]. Despite being more efficient than traditional methods, AOI faces challenges due to the miniaturization of PCBs and the diversity of defects, from functional faults to visual imperfections that compromise the durability of boards [7], [8]. In view of this, research is being carried out with the aim of presenting recent techniques based on deep learning to increase the efficiency of PCB development and also seek proposals for reducing the computational cost of architectures [1], [6], [9].

With the advancement of deep neural networks, methods such as RT-DETRv2 [10] and the YOLO family [11] have gained prominence for combining accuracy and real-time detection [6]. YOLO architectures, in particular, stand out for dispensing with intermediate steps, integrating feature extraction and detection directly and efficiently [1]. In this work, the YOLOv8n and YOLOv11n variants were selected because they represent the latest generations of the YOLO family, with significant advances in performance and computational lightness. The choice of the "nano" (n) versions is justified by the focus on environments with limited computational resources, such as embedded systems and real-time applications. YOLOv8 has already been successfully applied to defect detection in PCBs [9], [12], [13], while YOLOv11 brings recent innovations that increase the efficiency of the architecture. Finally, RT-DETRv2 was included as a representative of Transformer-based models, with a focus on real time. Only one model from this category was included in order to maintain the objectivity of the comparative analysis, focusing on recent, representative architectures that are still

The associate editor coordinating the review of this manuscript and approving it for publication was Ruth Aguilar (*Corresponding author: Thuanne Paixão*).

J. Martins, J. Lopez-Cabrejos, Q. Leher, Thuanne Paixão, and A. B. Alvarez are with the PAVIC Laboratory, University of Acre, Rio Branco, Brazil (e-mails: julio.sousa@sou.ufac.br, josue.cabrejos@sou.ufac.br, quefren.leher@sou.ufac.br, pavic.lab@ufac.br, and ana.alvarez@ufac.br).

F. Palomino-Quispe is with the LIECAR Laboratory, Universidad Nacional de San Antonio Abad del Cusco, Cusco, Peru (e-mail: facundo.palomino@unsaac.edu.pe).

under-explored in PCB defect detection.

This article presents a comparative analysis between the YOLOv8n, YOLOv11n and RT-DETRv2 models in the PCB defect detection task. The experiments are carried out using the PKU-Market-PCB [7] dataset. In order to reduce the computational cost of YOLO architectures, Depthwise Convolution (DWConv) blocks are implemented. The evaluations consider the quantitative metrics of precision, recall and mAP. For the qualitative evaluation, a visual analysis of the samples is being carried out and, finally, a robustness test is carried out with the best architectures. The main contributions of this paper are:

- Comparative analysis between transformer-based models and YOLO for PCB defect detection.
- Investigation of YOLOv8n-DWConv and YOLOv11n-DWConv variants, with replacement of traditional convolutional blocks by blocks based on Depthwise Convolution, in the defect detection task.
- Evaluation of the robustness of the models in challenging scenarios, including blur noise and illumination variations, to measure performance under different image quality conditions.

The structure of the paper is as follows: Section II presents a review of related work. Section III describes the methodology used, including the dataset, data pre-processing, the architectures used, a description of the DWConv blocks, the modified architectures, and the evaluation methods. Section IV contains the experimental results and discussions of defect detection. Finally, the conclusion is described in Section V.

## II. Literature Review

The field of PCB defect detection has seen considerable advancements over the years, with major developments classified into three primary approaches: referential methods, non-referential methods, and hybrid methods [7]. Referential methods entail the direct comparison of a PCB reference image and the inspected test image through the utilization of image processing techniques [14]–[16]. In contrast, non-reference methods do not require reference images and instead rely on general design rules and machine learning algorithms to identify defects [1]. Hybrid methods are a combination of these two approaches, combining elements of both to improve the detection process [17]–[19].

In the context of non-reference methods for detecting defects in PCBs, several recent studies have explored variations of the YOLO architecture. In 2023, Niu et al. [1] proposed an improved version of YOLOv5, using the k-means++ algorithm, the Focal-EIoU loss function instead of GIoU and the integration of the ECA-NET module. The tests carried out with the PKU-Market-PCB dataset [7], achieved a mAP of 99.10%. In the same year, Yang et al. [6] presented a method based on YOLOv7, incorporating the SwinV2-TDD module, the Multi-Scale Feature Self-Attention enhancement factor and adjustments to the extended layer aggregation network, as well as the use of the Mish activation function. Using the same data set, the method achieved a mAP of 98.74%. In 2024, Zhang et al. [9] proposed the YOLO-RRL model, based on YOLOv8, consisting of four main modules: Robust Feature Downsampling (RFD), Reparameterized Generalised FPN (RepGFPN), Dynamic Upsampler (DySample) and Lightweight Asymmetric Detection Head (LADH-Head). Evaluated on the HRIPCB [20] set, the model increased the mAP from 93.00% to 95.20%, increased the frame rate by 12% and reduced computational complexity.

In 2025, Huang et al. [21] combined an improved version of YOLOv11 with generative adversarial networks (GANs) for synthetic image generation, expanding the HRIPCB set and improving the model's performance. The solution obtained a mAP of 95.00% and a recall of 87.00%. Also in 2025, Wang et al. [22] developed the lightweight YOLOv8-PCB algorithm, incorporating the C2f-SHSA (attention) and C2f-IdentityFormer modules. Evaluated with the Peking University Open Lab on Human-Robot Interaction (PKU-HRI) set [23], the method obtained a recall of 94.00% and mAP of 96.10%. Also in 2025, Tang et al. [24] improved YOLOv8n with a multi-module collaborative optimization approach. The model incorporated mechanisms such as SCSA attention, the Unified-IoU loss function, the MobileNetV4 backbone with MobileMQA module, and the ASF-SDI Neck structure, aiming to improve the detection of small targets with lower computational cost. Evaluated on the PKU-Market-PCB dataset, it achieved 98.80% accuracy, 99.20% recall, and mAP@50 of 99.10%.

In contrast to the research presented, this paper explores YOLOv11n and RT-DETRv2, models that have been insufficiently investigated, in comparison with YOLOv8n in the task of defect detection in PCBs. In addition, this study investigates the use of DWConv blocks to reduce the computational cost of the YOLOv8n and YOLOv11n models, evaluating their impact within the architectures. Furthermore, a robustness analysis of the behavior of the evaluated models is performed.

## III. Materials and Methods

### A. Dataset

The PKU-Market-PCB public dataset was selected for the experimental analysis due to its extensive utilization among papers related to the task of defect detection in PCB circuits [1], [6], [7], [25]. The public PKU-Market-PCB dataset [7] was used for the experiments because it is widely adopted in the literature [1], [6], [7], [25] and because it most closely represents real-world application scenarios.

PKU-Market-PCB [7] is a comprehensive image analysis framework that addresses a wide range of defects, including the Missing Hole, the presence of Spurious Copper, Short, Spur, Mouse Bite, and Open Circuit. Each sample in the dataset is characterized by a unique defect type, with an average image size of 2777 x 2138 pixels. Examples of samples with the defects present in the dataset are shown in Fig. 1.

### B. Data Pre-processing

The original PKU-Market-PCB dataset contains 1,386 annotated images, including samples with different orientations (such as rotations), but does not provide a standard division between training, validation, and testing. Thus, following the

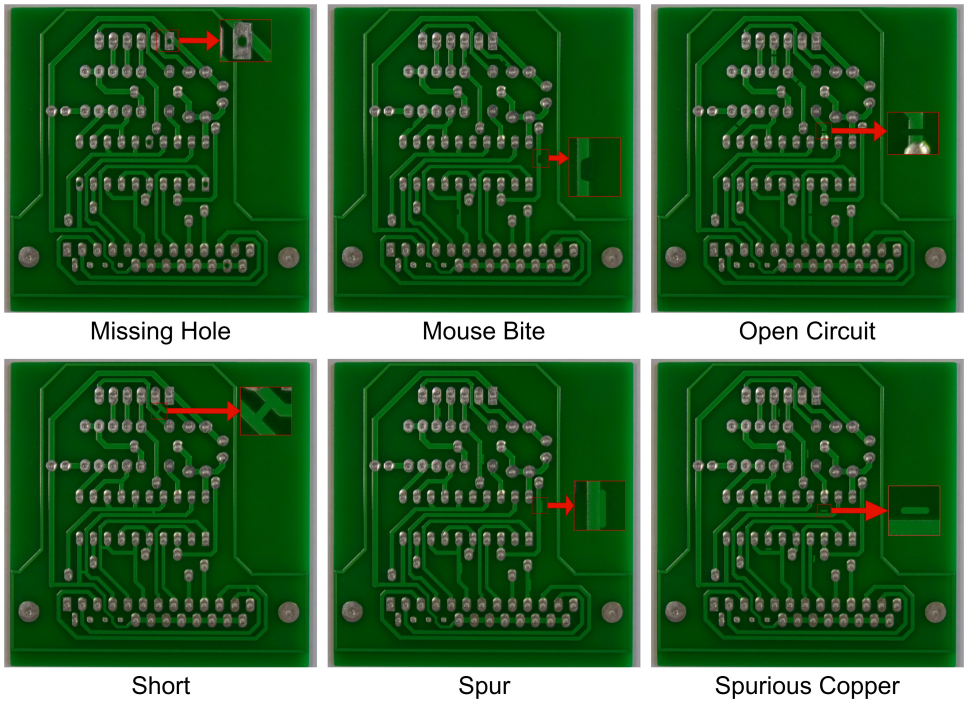


Fig. 1. Types of defects found in the PKU-Market-PCB dataset. [7].

approach proposed by Niu et al. [1], a data augmentation procedure was adopted with cropping into 600 × 600 pixel subimages from the original images. This process resulted in the creation of training, validation, and test sets containing 5,981, 2,990, and 997 images, respectively, as detailed in Table I.

TABLE I
SUMMARY OF PCB DATASETS

| Set | Image Resolution | Number of Patches |
|---|---|---|
| Train | 600 × 600 | 5981 |
| Validation | 600 × 600 | 2990 |
| Test | 600 × 600 | 997 |

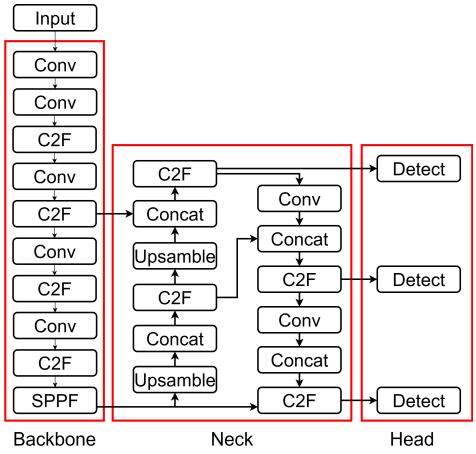


Fig. 2. YOLOv8 architecture.

### C. YOLOv8

The YOLOv8 [26] developed by Ultralytics and launched in 2023, represents the eighth generation of the YOLO family, aimed at detecting objects in real time. The range includes five variants, YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x, which vary in complexity, precision and computational performance. Among the main innovations of YOLOv8 are the adoption of modern structures, such as the Feature Pyramid Network (FPN) and the Path Aggregation Network (PAN), which improve the detection of objects at multiple scales.

As illustrated in Fig. 2, the YOLOv8 architecture is made up of three main blocks: backbone, neck and head. The backbone extracts features from the image at different levels, from simple patterns such as edges to more abstract information. The neck connects the backbone to the head, refining the extracted features through operations such as upsampling and concatenation, as well as the use of FPN and PAN networks. The head makes the final predictions, generating the coordinates of the bounding boxes, the object classes and their respective confidence scores [27].

### D. YOLOv11

YOLOv11 [28] was designed for real-time object detection, while maintaining the unified network structure characteristic of the YOLO family. This approach allows detection and classification to be carried out in a single step, which simplifies the process compared to two-step architectures and makes training and inference more efficient. The version includes five variants, YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l and YOLOv11x. Compared to YOLOv8, YOLOv11 features significant improvements, such as the new C3k2 block, the

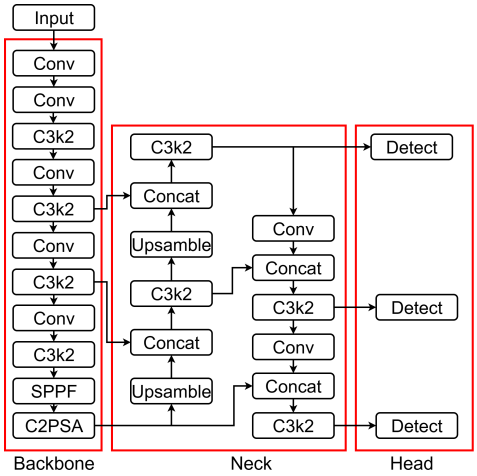


Fig. 3. YOLOv11 architecture.

C2PSA spatial attention mechanism and a more efficient integration of features at multiple scales. These changes increase precision and reduce computational complexity, while maintaining the model's adaptability.

The YOLOv11 architecture, illustrated in Fig. 3, is divided into three parts: backbone, neck and head. The backbone introduces C3k2, which replaces YOLOv8's C2f block, using two smaller convolutions instead of one larger one, which reduces the computational load without losing performance. The SPPF block is maintained, now integrated with C2PSA, which reinforces the network's spatial attention, allowing greater focus on relevant regions of the image. C3k2 is also used in neck, improving feature aggregation, while C2PSA contributes to the detection of small or partially visible objects.

### *E. RT-DETRv2*

RT-DETRv2, proposed by Lv et al. [10], is an end-to-end real-time object detector that improves RT-DETR [29], with a focus on performance, practicality and high inference speed. Its architecture consists of three main modules: a convolutional backbone, an efficient hybrid encoder and a Transformer decoder with auxiliary heads. Fig. 4 shows a simplified version of the architecture, where the input image is processed by a convolutional backbone, which extracts feature maps at multiple scales. These representations feed the hybrid encoder, made up of the AIFI (Attention-based Intra-scale Feature Interaction) module, which applies local self-attention at a lower computational cost, and the CCFF (Cross-scale Feature Fusion) module, which fuses different scales using optimized convolutions.

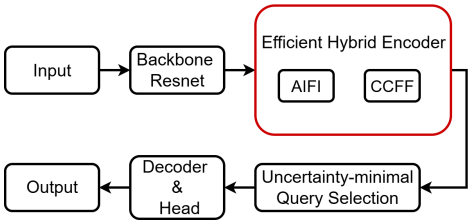


Fig. 4. RT-DETRv2 architecture.

Next, the Uncertainty-minimal Query Selection module selects queries based on the joint confidence of the class and location predictions, reducing uncertainty and improving the quality of the queries provided to the decoder. The Transformer decoder refines these queries iteratively with deformable attention and adaptive sampling points. Auxiliary prediction heads are used to stabilize training and improve precision. In the end, the model provides the object categories and bounding boxes directly, without the need for post-processing.

### *F. Depthwise Convolution*

DWConv [30] is a technique in which each input channel is processed separately by a filter. Unlike traditional convolution, which applies filters combining information from all channels, DWConv only considers spatial correlations within each channel, without mixing data between them. So, if the input has $N$ channels, $N$ filters will be applied, one for each channel. Fig. 5 shows the difference between the two approaches. In traditional convolution, the filters extract spatial patterns and combine information between channels. In DWConv, each channel is treated independently, which reduces the number of parameters and the computational cost, making the process more efficient.

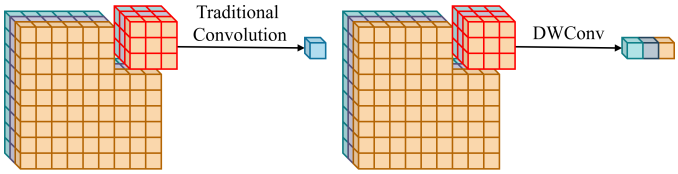


Fig. 5. Comparison between Standard Convolution and Depthwise Convolution.

In traditional convolution, represented by Equation 1, a filter $K$ with dimensions $W \times W \times M \times N$ is applied to the input map $F$ with dimensions $D_f \times D_f \times M$. For each position $(k, l)$ and output channel $n$, the value $O_{k,l,n}$ is obtained as follows:

$$O_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m} \tag{1}$$

This process captures spatial patterns and mixes information between channels, producing an output map with $N$ channels.

In DWConv, described by Equation 2, a filter $\hat{K}$ of dimensions $3 \times 3 \times M$ is applied separately to each channel $m$ of the input map $F$. The output value $\hat{O}_{k,l,m}$ is calculated as:

$$\hat{O}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,l+j-1,m} \tag{2}$$

Each output depends only on data from the same channel at the input, with no information exchanged between channels. While this reduces complexity, it can also limit the model's ability to capture relationships between channels.

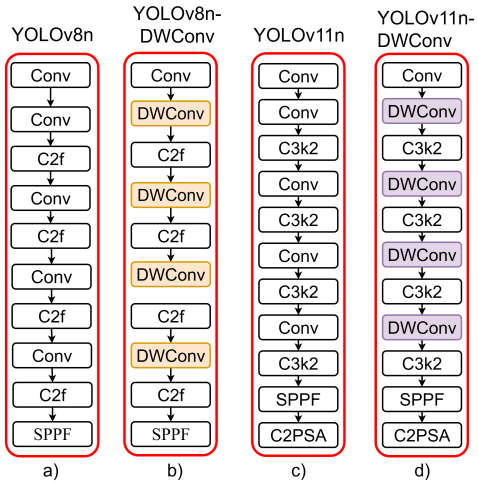


Fig. 6. Backbone models YOLOv8n, YOLOv8n-DWConv, YOLOv11n and YOLOv11n-DWConv.

### G. YOLO Architectures with Depthwise Convolution

Based on [31] and [32], specific modifications were implemented in the backbones of the YOLOv8n and YOLOv11n architectures, aiming to reduce computational cost. In these changes, the standard convolutional blocks (Conv) were replaced by depthwise separable convolutions (DWConv), a lighter and more efficient alternative. The decision to modify the backbone was strategic, considering that this part of the network deals with high-resolution feature maps and is therefore responsible for a large part of the computational load.

Fig. 6 illustrates these modifications. Figs. 6.a and 6.c show the original backbones of the YOLOv8n and YOLOv11n models, while Figs. 6.b and 6.d show the modified versions, YOLOv8n-DWConv and YOLOv11n-DWConv, highlighting the blocks in which DWConv was used to improve model efficiency.

### H. Metrics and Validation

This subsection presents the metrics used to quantify and qualify the performance of the implemented models, which include the precision, recall, average precision (AP), and mean average precision metrics and the confidence score.

*1) Precision:* Precision [33] is defined as the ratio between the number of True Positives (TP) and the total number of positive detections, which includes TP and False Positives (FP), and can be calculated mathematically according to Equation 3.

$$P = \frac{TP}{TP + FP} \tag{3}$$

Where TP is the number of samples correctly classified as positive and FP is the number of samples incorrectly classified as positive.

*2) Recall:* Recall [34] measures the model's ability to correctly identify all relevant examples. It is defined as the ratio between the number of TP and the total number of positive examples, which includes TP and False Negatives (FN). Recall is expressed by Equation 4.

$$R = \frac{TP}{TP + FN} \tag{4}$$

Where FN is the number of examples incorrectly classified as negative.

*3) Average Precision:* Average Precision [12] is determined by the area under the precision and recall curve. This area is obtained by integrating P with respect to R varying from 0 to 1, as shown in Equation 5.

$$AP = \int_0^1 P(R)\, dR \tag{5}$$

*4) Mean Average Precision:* Mean Average Precision (mAP) [35] is widely used to assess the quality of object detection models. It takes into account both classification precision and the location of detected objects. The mAP is calculated according to Equation 6, which is the average of the mean precisions.

$$mAP = \frac{\sum_{i=1}^{n} AP(i)}{n} \tag{6}$$

*5) Confidence Score:* The confidence score in detection models, such as YOLO, varies between 0 and 1 and indicates the model's level of certainty that the predicted box really does correspond to an object in the image [36]. Values close to 1 indicate high confidence, while values close to 0 indicate low confidence in the detection.

*6) Computational Cost Metrics:* The parameters and GFLOPs indicate, respectively, the size of the model parameters and the computational complexity, and are essential metrics for assessing the complexity of the model and the need for resources [37].

## IV. Experimental Results

### A. Hardware and Software

The computing environment utilized for training the implemented models consists of Python 3.11.9, PyTorch 2.5.1, the Red Hat Enterprise Linux 8.9 operating system, and an NVIDIA A100 GPU. The hyperparameters selected for the training process include the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.01, a batch size of 160, and a total of 1000 epochs. In addition, in order to avoid superfluous training and optimize the convergence of the model, early stopping was implemented with a patience of 100 epochs, ending the training if there was no improvement in the results within this interval. For the model validation and testing phase, a computing environment was utilized with a GeForce RTX 3050 GPU, the Windows 11 operating system, and a batch size of 4.

### B. Comparison of the Original YOLO Architectures

The PKU-Market-PCB dataset was used to test the RT-DETRv2, YOLOv8n and YOLOv11n models. Quantitative analysis was carried out on the basis of the metrics and qualitative evaluation on the samples of board defects. In addition, a computational cost analysis was carried out to assess the efficiency of each model.

*1) Quantitative Evaluation:* The results of the precision, recall, mAP0.5 and mAP0.5:0.95 metrics of the YOLOv8n, YOLOv11n and RT-DETRv2 models in detecting PCB defects are shown in Table II. The results indicate that the YOLOv11n model outperforms YOLOv8n and RT-DETRv2 overall in the test set, including all faulty patches, with respect to most metrics.

TABLE II
QUANTITATIVE PERFORMANCE OF YOLOV8N, YOLOV11N AND RT-DETRV2 MODELS

| Model | P | R | mAP0.5 | mAP0.5:0.95 |
|---|---|---|---|---|
| YOLOv8n | 98.40 | 98.60 | **99.30** | 75.60 |
| YOLOv11n | **99.00** | 99.20 | **99.30** | **81.10** |
| RT-DETRv2 | – | **99.80** | 99.20 | 67.50 |

This superiority of YOLOv11n can be attributed to improvements in its architecture, such as the C3k2 block and the C2PSA spatial attention mechanism, which make feature extraction more efficient and defect detection more precise.

*2) Qualitative Assessment:* A qualitative assessment was conducted on six samples of representative defects present in the PCBs, each sample corresponding to one patch in the data set. Figure 7 shows the results of the analysis, highlighting in orange the cases of failure of each model in the images.

It is observed that the evaluated models show different performance in detecting defects in PCBs. YOLOv8n had average performance, with variation in the types of failures, including false negatives in samples 1 and 6 and false positives in samples 3 and 5. YOLOv11n showed more consistent performance than YOLOv8n with a more predictable error profile, with a predominance of false positives, especially in samples 3 and 4, in addition to a false negative in sample 1. RT-DETRv2, in turn, showed the best results in this analysis, with one false negative in sample 1 and one false positive in sample 2.

*3) Evaluation of Computational Cost:* Considering the number of parameters and GFLOPs as metrics for analyzing computational cost, Table III shows the metrics for the YOLOv8n, YOLOv11n and RT-DETRv2 models. It can be seen that YOLOv11n has fewer parameters and consequently uses fewer GFLOPs than the other methods, making it a version with a lower computational cost.

TABLE III
COMPUTATIONAL COST OF THE YOLOV8N, YOLOV11N AND RT-DETRV2 MODELS

| Model | Parameters | GFLOPs |
|---|---|---|
| YOLOv8n | 3,006,818 | 8.10 |
| YOLOv11n | **2,583,322** | **6.30** |
| RT-DETRv2 | 18,831,000 | 29.69 |

*4) Validation and Comparative Analysis:* To ensure the robustness and generalization of the results, the performance of the models was evaluated using the k-fold cross-validation technique with k=5. In addition, in order to verify the existence of significant differences in performance metrics, two statistical tests were used: the paired T-test, used when comparing two models, and ANOVA, applied when comparing the three models. In both cases, a significance level of 5% was adopted, i.e., if the p-value is less than 0.05, a statistically significant difference between the models is considered to exist. Table IV summarizes the quantitative performance, presenting the mean and standard deviation of the metrics for each model across the 5 folds.

TABLE IV
QUANTITATIVE PERFORMANCE OF YOLOV8N, YOLOV11N AND RT-DETRV2 MODELS FOR VALIDATION KFOLD

| Model | P | R | mAP0.5 | mAP0.5:0.95 |
|---|---|---|---|---|
| YOLOv8n | $98.50 \pm 0.23$ | $98.52 \pm 0.29$ | $99.08 \pm 0.10$ | $72.52 \pm 0.26$ |
| YOLOv11n | $98.52 \pm 0.33$ | $98.58 \pm 0.30$ | $99.12 \pm 0.10$ | $73.04 \pm 1.35$ |
| RT-DETRv2 | – | $99.86 \pm 0.05$ | $99.00 \pm 0.25$ | $64.86 \pm 0.31$ |

For the Precision (P) metric, the paired T-test between YOLOv8n and YOLOv11n indicated that there is no statistically significant difference ($p = 0.8757$). In this metric, ANOVA was not applied, as there was no data available for RT-DETRv2. In the Recall (R) metric, ANOVA indicated a statistically significant difference ($p < 0.0001$), mainly attributed to the high performance of RT-DETRv2. For themAP@0.5 metric, ANOVA did not identify a significant difference between the models ($p = 0.5971$), indicating similar performance. Finally, in the "mAP@0.5:0.95" metric, ANOVA revealed a statistically significant difference ($p < 0.0001$), with YOLOv11n performing better than the others.

Looking at the results of the statistical evaluation, it can be concluded that YOLOv11n performs statistically equivalent to YOLOv8n in terms of precision, recall, and mAP@0.5 metrics, but significantly outperforms the other models in mAP@0.5:0.95. Although RT-DETRv2 stood out in recall, its inferior performance in mAP@0.5:0.95 limits its overall accuracy.

### *C. Comparing YOLO Architectures with Depthwise Convolution*

The performance of the and YOLOv11n-DWConv models, in which the traditional convolution blocks have been replaced by DWConv blocks, is evaluated using quantitative metrics, qualitative analysis, and computational cost comparisons.

*1) Quantitative Evaluation:* The quantitative results of the YOLOv8n-DWConv and YOLOv11n-DWConv models in the task of detecting defects in PCBs on the dataset patches are shown in the Table V. The results for the complete test set demonstrate that YOLOv11n-DWConv performs better when analyzing the precision metrics, mAP0.5 and mAP0.5:0.95. For the recall metric, YOLOv8n-DWConv demonstrates a marginally higher percentage.

When comparing the performance results of the YOLOv11n and YOLOv11n-DWConv models, shown in Tables V and II, it can be seen that YOLOv11n obtained superior results. This can be explained by the traditional convolutional layer, which processes all input channels simultaneously, allowing for more efficient feature extraction and better generalization.

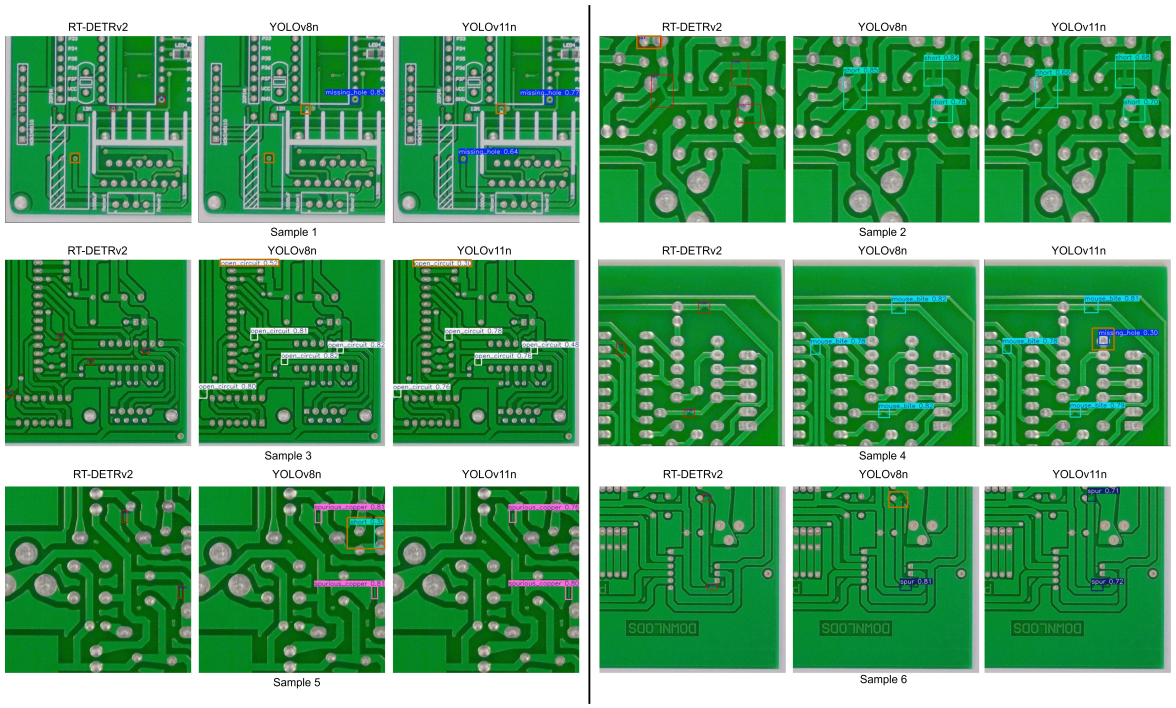


Fig. 7. Samples chosen for qualitative analysis between RT-DETRv2, YOLOv8n and YOLOv11n.

TABLE V
QUANTITATIVE PERFORMANCE FOR EACH CLASS OF THE YOLOV8N-DWCONV AND YOLOV11N-DWCONV MODELS

| Model | P | R | mAP0.5 | mAP0.5:0.95 |
|---|---|---|---|---|
| YOLOv8n-DWConv | 97.80 | **99.20** | **99.10** | 71.90 |
| YOLOv11n-DWConv | **98.20** | 98.40 | **99.10** | **72.10** |

In contrast, DWConv applies filters independently to each channel, significantly reducing the number of parameters and computational operations. However, this approach limits the interaction between channels, making it difficult to capture more complex contextual information. As a result, the DWConv block makes the YOLOv8n-DWConv and YOLOv11n-DWConv models less sensitive to the fine details of PCB defects.

*2) Qualitative Assessment:* Fig. 8 presents a qualitative analysis of the results in six representative samples, highlighting in orange the cases of failure of each model in the images.

The analysis of the YOLOv8n-DWConv and YOLOv11n-DWConv models also showed distinct patterns in the detection of defects in PCBs. YOLOv8n-DWConv showed average performance, with a predominance of false negatives in samples 1 and 2 and false positives in samples 3 and 6, with the latter also showing classification errors. In sample 4, the model generated overlapping detections with different classes in the same region, indicating confusion in the classification. YOLOv11n-DWConv was more consistent, with fewer failures, but with false negatives in samples 1 and 5, and a classification error in sample 2.

*3) Evaluation of Computational Cost:* Table VI shows the number of parameters and the speed of execution of the floating point operations of the proposed models. The YOLOv11n-DWConv model proved to be faster, with 4.90 GFLOPs and 2,103,802 parameters, when compared to YOLOv8n-DWConv, presenting the best computational cost in the task of detecting defects in PCBs.

TABLE VI
COMPUTATIONAL COST OF THE YOLOV8N-DWCONV AND YOLOV11N-DWCONV MODELS

| Model | Parameters | GFLOPs |
|---|---|---|
| YOLOv8n-DWConv | 2,619,458 | 7.20 |
| YOLOv11n-DWConv | **2,103,802** | **4.90** |

Comparing the computational cost of the models shown in Table III and VI, it can be seen that YOLOv11n-DWConv is lighter and faster than YOLOv11n, with a reduction in parameters and GFLOPs of more than 20%.

*4) Validation and Comparative Analysis:* Table VII shows the results of k-fold cross-validation (k=5) and T-test for the YOLOv8n-DWConv and YOLOv11n-DWConv models, with the mean and standard deviation of each metric.

TABLE VII
QUANTITATIVE PERFORMANCE FOR EACH CLASS OF THE YOLOV8N-DWCONV AND YOLOV11N-DWCONV MODELS FOR KFOLD

| Model | P | R | mAP0.5 | mAP0.5:0.95 |
|---|---|---|---|---|
| YOLOv8-DWConv | $98.34 \pm 0.29$ | $98.32 \pm 0.27$ | $98.90 \pm 0.14$ | $69.16 \pm 0.44$ |
| YOLOv11-DWConv | $98.22 \pm 0.21$ | $98.44 \pm 0.14$ | $99.06 \pm 0.08$ | $69.30 \pm 0.14$ |

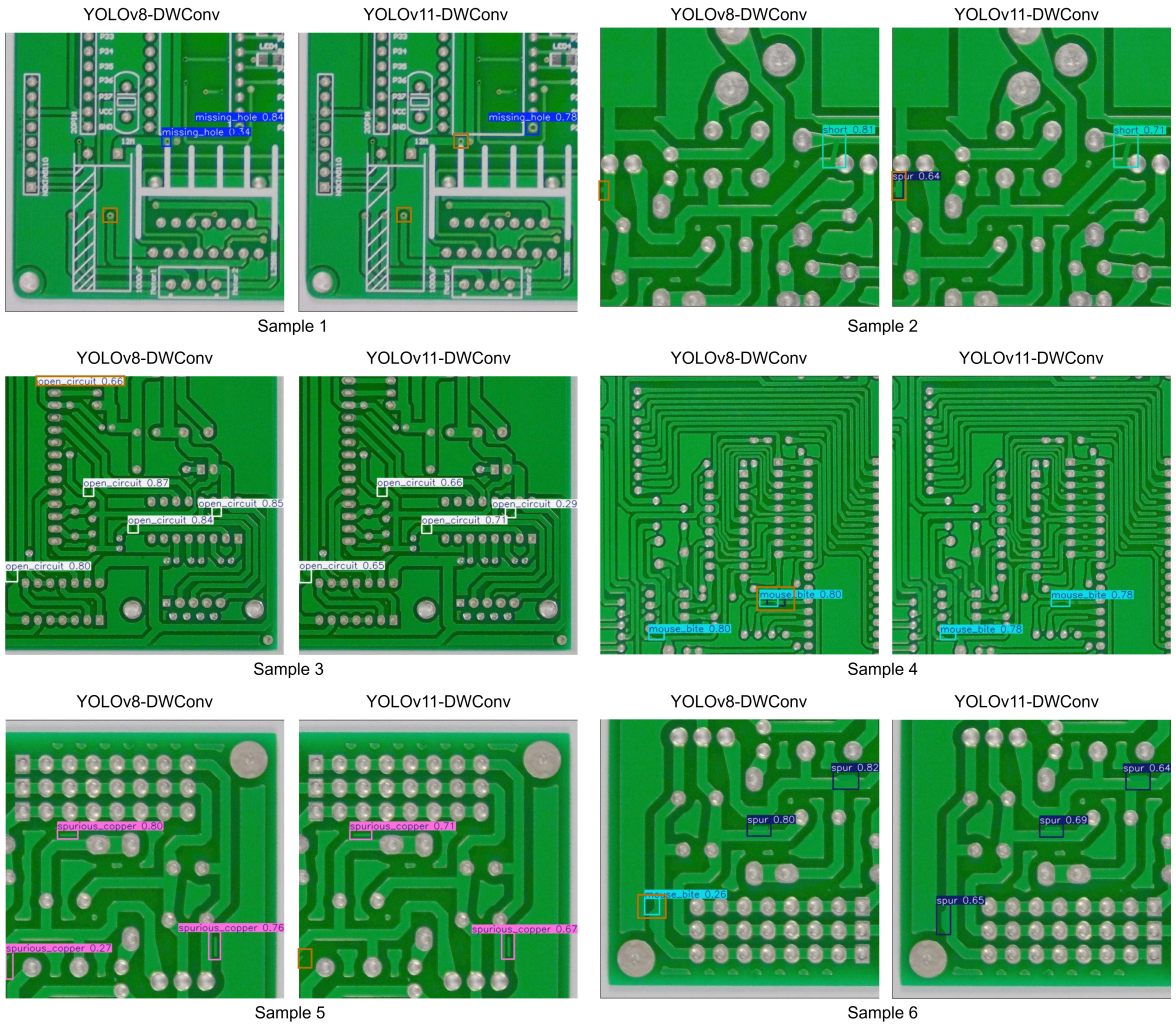


Fig. 8. Samples chosen for qualitative analysis between YOLOv8n-DWConv and YOLOv11n-DWConv.

The results obtained with the Paired T-test showed that there is no statistical difference between the two models in any of the metrics of precision ($p = 0.3883$), $recall$(p=0.2355), mAP@0.5 ($p = 0{,}1202$), and mAP@0.5:0.95 ($p = 0{,}5520$).

### *D. Robustness Test*

A complementary analysis was conducted to investigate the generalization ability of the models in scenarios with visual degradations, using the test set with 997 images, highlighting that these images were not used in training. To this end, two types of noise were applied in a controlled and consistent manner to the test set: Gaussian blur, using a filter with $\sigma = 4$ and a $5 \times 5$ pixel kernel, and a linear lighting gradient with an intensity of 0.25, simulating smooth and continuous variations in lighting. All models evaluated in this research were included in the analysis, and the quantitative results of defect detection in PCB boards are presented in Table VIII.

The results in Table VIII show that YOLOv11n has the best overall performance, maintaining high precision, recall, and mAP values in all tested conditions. This indicates that its architecture is more robust to variations in lighting and blurring. Models with DWConv, although lighter, showed significant drops, especially in situations with uneven lighting. RT-DETRv2 maintained high recall but performed poorly with blurred images, which may limit its use in environments with visual noise. YOLOv8n had intermediate performance, stable in ideal scenarios but sensitive to degradation. Thus, YOLOv11n stands out as the most balanced option for prac-

TABLE VIII
DETECTION PERFORMANCE UNDER VARIOUS NOISE DEGRADATION CONDITIONS

| Condition | Model | Precision | Recall | mAP@0.5 | mAP@0.5:0.95 |
|---|---|---|---|---|---|
| No Degradation | YOLOv8n | 98.40 | 98.60 | 99.30 | 75.60 |
| | YOLOv8n-DWConv | 97.80 | 99.20 | 99.10 | 71.90 |
| | YOLOv11n | **99.00** | **99.20** | **99.30** | **81.10** |
| | YOLOv11n-DWConv | 98.20 | 98.40 | 99.10 | 72.10 |
| | RT-DETRv2 | - | 99.80 | 99.20 | 67.50 |
| Illumination Gradient | YOLOv8n | 98.50 | 98.30 | 99.20 | 74.80 |
| | YOLOv8n-DWConv | 96.60 | 94.00 | 98.40 | 66.50 |
| | YOLOv11n | **98.80** | **99.10** | **99.30** | **80.00** |
| | YOLOv11n-DWConv | 96.40 | 93.40 | 98.40 | 67.20 |
| | RT-DETRv2 | - | 99.80 | 99.20 | 67.60 |
| Blur | YOLOv8n | 97.00 | 96.00 | 98.20 | 68.10 |
| | YOLOv8n-DWConv | 94.30 | 94.30 | 98.30 | 67.20 |
| | YOLOv11n | **98.80** | **99.10** | **99.30** | **78.10** |
| | YOLOv11n-DWConv | 97.80 | 94.90 | 98.70 | 67.70 |
| | RT-DETRv2 | - | 99.10 | 96.50 | 54.50 |

tical applications, combining accuracy and robustness.

## V. Conclusion and Future Work

This article presents a comparative analysis of the RT-DETRv2, YOLOv8n and YOLOv11n models for identifying defects in PCBs, as well as an evaluation of the performance and computational cost of the YOLOv8n-DWConv and YOLOv11n-DWConv architectures, which replace traditional convolutional blocks with Depthwise Convolution blocks. In addition, the robustness of the models was evaluated in degraded images with blur and illumination gradient noise. In terms of computational cost, YOLOv11n-DWConv was the fastest and lightest model, reducing the number of GFLOPs from 6.30 to 4.90, representing a reduction of more than 20% compared to the YOLOv11n model. This was possible thanks to the replacement of traditional blocks with Depthwise Convolution, which reduced the number of parameters and sped up defect detection. In terms of performance, YOLOv11n showed a more significant result in the metrics evaluated with a precision of 99.00, recall of 99.20 and mAP0.5 of 99.30. In the robustness analysis simulating challenging situations, YOLOv11n proved to be more efficient. In contrast, the YOLOv11n-DWConv model, even with its computational lightness, offered a competitive performance with a precision of 98.20, recall of 98.40 and mAP0.5 of 99.10. Future work may focus on building a more diverse dataset with different types of defects, applying fine-tuning techniques, comparative statistical analyses of errors by class, and other lightweight architectures. Model compression strategies such as knowledge distillation and quantization-aware training may also be explored. Finally, an evaluation of implementation in real industrial inspection scenarios is also envisaged.

## Acknowledgments

The authors gratefully acknowledge support from the PAVIC Laboratory, benefited from SUFRAMA fiscal incentives under Brazilian Law No. 8387/1991.

## References

[1] J. Niu, H. Li, X. Chen, and K. Qian, "An improved yolov5 network for detection of printed circuit board defects," *Journal of Sensors*, vol. 2023, no. 1, p. 7270093, 2023. doi: 10.1155/2023/7270093 .

[2] Y.-S. Deng, A.-C. Luo, and M.-J. Dai, "Building an automatic defect verification system using deep neural network for pcb defect classification," in *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*. IEEE, 2018, pp. 145–149. doi: 10.1109/ICFSP.2018.8552045 .

[3] J. P. Nayak and B. Parameshachari, "Effective pcb defect detection using stacked autoencoder with bi-lstm network." *International Journal of Intelligent Engineering & Systems*, vol. 15, no. 5, 2022. doi: 10.22266/ijies2022.1031.05 .

[4] Z. Xiao, Z. Wang, D. Liu, and H. Wang, "A path planning algorithm for pcb surface quality automatic inspection," *Journal of Intelligent Manufacturing*, vol. 33, no. 6, pp. 1829–1841, 2022. doi: 10.1007/s10845-021-01766-3 .

[5] W.-C. Wang, S.-L. Chen, L.-B. Chen, and W.-J. Chang, "A machine vision based automatic optical inspection system for measuring drilling quality of printed circuit boards," *IEEE access*, vol. 5, pp. 10 817–10 833, 2016. doi: 10.1109/ACCESS.2016.2631658 .

[6] Y. Yang and H. Kang, "An enhanced detection method of pcb defect based on improved yolov7," *Electronics*, vol. 12, no. 9, p. 2120, 2023. doi: 10.3390/electronics12092120 .

[7] R. Ding, L. Dai, G. Li, and H. Liu, "Tdd-net: a tiny defect detection network for printed circuit boards," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 2, pp. 110–116, 2019. doi: 10.1049/trit.2019.0019 .

[8] W.-Y. Wu, M.-J. J. Wang, and C.-M. Liu, "Automated inspection of printed circuit boards through machine vision," *Computers in industry*, vol. 28, no. 2, pp. 103–111, 1996. doi: 10.1016/0166-3615(95)00063-1 .

[9] T. Zhang, J. Zhang, P. Pan, and X. Zhang, "Yolo-rrl: A lightweight algorithm for pcb surface defect detection," *Applied Sciences*, vol. 14, no. 17, p. 7460, 2024. doi: 10.3390/app14177460 .

[10] W. Lv, Y. Zhao, Q. Chang, K. Huang, G. Wang, and Y. Liu, "Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer," 07 2024. doi: 10.48550/arXiv.2407.17140 .

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91 .

[12] T. Yuan, Z. Jiao, and N. Diao, "Yolo-ssw: An improved detection method for printed circuit board surface defects," *Mathematics*, vol. 13, no. 3, 2025. doi: 10.3390/math13030435 . [Online]. Available: https://www.mdpi.com/2227-7390/13/3/435

[13] H. Wang, S. Shen, and M. Li, "Pcb defect detection algorithm based on improved yolov8," in *2024 5th International Conference on Electronic Communication and Artificial Intelligence (ICECAI)*. IEEE, 2024, pp. 323–327. doi: 10.1109/ICECAI62591.2024.10675028 .

[14] P. Malge and R. Nadaf, "Pcb defect detection, classification and localization using mathematical morphology and image processing tools," *International journal of computer applications*, vol. 87, no. 9, 2014. doi: 10.5120/15240-3782 .

[15] Y. Chang, Y. Xue, Y. Zhang, J. Sun, Z. Ji, H. Li, T. Wang, and J. Zuo, "Pcb defect detection based on pso-optimized threshold segmentation and surf features," *Signal, Image and Video Processing*, vol. 18, no. 5, pp. 4327–4336, 2024. doi: 10.1007/s11760-024-03075-7 .

[16] W. Huang and P. Wei, "A pcb dataset for defects detection and classification," *arXiv preprint arXiv:1901.08204*, 2019. doi: 10.48550/arXiv.1901.08204 .

[17] Z. Qu, J. Shen, R. Li, J. Liu, and Q. Guan, "Partsnet: A unified deep network for automotive engine precision parts defect detection," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, 2018, pp. 594–599. doi: 10.1145/3297156.3297190 .

[18] 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, 2021. doi: 10.3390/s21154968 .

[19] S. Ray and J. Mukherjee, "A hybrid approach for detection and classification of the defects on printed circuit board," *International Journal of Computer Applications*, vol. 121, no. 12, 2015. doi: 10.5120/21595-4691 .

[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, 2020.

[21] J. Huang, F. Zhao, and L. Chen, "Defect detection network in pcb circuit devices based on gan enhanced yolov11," *arXiv preprint arXiv:2501.06879*, 2025. doi: 10.48550/arXiv.2501.06879 .

[22] J. Wang, X. Xie, G. Liu, and L. Wu, "A lightweight pcb defect detection algorithm based on improved yolov8-pcb," *Symmetry*, vol. 17, no. 2, 2025. doi: 10.3390/sym17020309 . [Online]. Available: https://www.mdpi.com/2073-8994/17/2/309

[23] S. Xu, Y. Li, and Q. Liang, "Bare pcb defect detection based on improved yolov5 algorithm," *Packaging Engineering*, vol. 43, no. 15, pp. 33–41, 2022. doi: 10.1109/safepress58597.2023.10295682 .

[24] Y. Tang, R. Liu, and S. Wang, "Yolo-sumas: Improved printed circuit board defect detection and identification research based on yolov8," *Micromachines*, vol. 16, no. 5, p. 509, 2025.

[25] M. Yuan, Y. Zhou, X. Ren, H. Zhi, J. Zhang, and H. Chen, "Yolo-hmc: An improved method for pcb surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–11, 2024. doi: 10.1109/TIM.2024.3351241 .

[26] M. Sohan, T. Sai Ram, R. Reddy, and C. Venkata, "A review on yolov8 and its advancements," in *International Conference on Data Intelligence and Cognitive Informatics*. Springer, 2024, pp. 529–545. doi: 10.1007/978-981-99-7962-2-39 .

[27] M. Hussain, "Yolov5, yolov8 and yolov10: The go-to detectors for real-time vision," *arXiv preprint arXiv:2407.02988*, 2024. doi: 10.48550/arXiv.2407.02988 .

[28] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024. doi: 10.48550/arXiv.2410.17725 .
[29] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat yolos on real-time object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 16 965–16 974. doi: 10.1109/CVPR52733.2024.01605 .
[30] Y. Guo, Y. Li, L. Wang, and T. Rosing, "Depthwise convolution is all you need for learning multiple visual domains," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 8368–8375. doi: 10.1609/aaai.v33i01.33018368 .
[31] K. Chen, X. Zhou, and J. Ren, "Dlf-yolo: A dynamic synergy attention-guided lightweight framework for few-shot clothing trademark defect detection," *Electronics*, vol. 14, no. 11, p. 2113, 2025.
[32] H. T. Dinh and E.-T. Kim, "A lightweight network based on yolov8 for improving detection performance and the speed of thermal image processing," *Electronics*, vol. 14, no. 4, p. 783, 2025.
[33] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors," *arXiv preprint arXiv:2411.00201*, 2024. doi: 10.48550/arXiv.2411.00201 .
[34] Y. Gong, Z. Chen, W. Deng, J. Tan, and Y. Li, "Real-time long-distance ship detection architecture based on yolov8," *IEEE Access*, 2024. doi: 10.1109/ACCESS.2024.3445154 .
[35] D. Wang, J. Tan, H. Wang, L. Kong, C. Zhang, D. Pan, T. Li, and J. Liu, "Sds-yolo: An improved vibratory position detection algorithm based on yolov11," *Measurement*, vol. 244, p. 116518, 2025. doi: 10.1016/j.measurement.2024.116518 .
[36] J. Bento, T. Paixão, and A. B. Alvarez, "Performance evaluation of yolov8, yolov9, yolov10, and yolov11 for stamp detection in scanned documents," *Applied Sciences*, vol. 15, no. 6, 2025. doi: 10.3390/app15063154 . [Online]. Available: https://www.mdpi.com/2076-3417/15/6/3154
[37] H. Yan, H. Zhang, F. Gao, H. Wu, and S. Tang, "Research on deep learning model enhancements for pcb surface defect detection," *Electronics*, vol. 13, no. 23, p. 4626, 2024. doi: 10.3390/electronics13234626 .

**Julio Martins** (Member, IEEE) is currently pursuing a Bachelor's degree in Electrical Engineering at the Federal University of Acre (UFAC), Brazil. He is actively involved as a researcher in the PD project Applied Research in Computer Vision and Intelligence (PAVIC-Lab), a collaborative initiative between UFAC, Motorola, Flextronics and FUNDAPE. His research interests include object detection models.

**Josue Lopez-Cabrejos** (Member, IEEE) received his Bachelor's degree in Electronic Engineering and Telecommunications (2024) from the National Technological University of Lima Sur (UNTELS), Peru. He is currently pursuing a Master's degree in Computer Science at the Federal University of Acre (UFAC). His research interests include computer vision with a focus on Vision Transformer-based models. He is currently participating as a researcher in the PD Project Applied Research in Computer Vision and Intelligence (PAVIC-Lab), a partnership between UFAC, Motorola, Flextronics and FUNDAPE.

**Quefren Leher** (Member, IEEE) received his B.S. degree in Electrical Engineering from the Federal University of Acre, Brazil, in 2024. Currently pursuing an M.S. in Computer Science at the same institution, with a focus on intelligent computational systems. His research interests include computer vision, with a focus on generative models, and the application of Python in engineering and artificial intelligence. He is currently participating as a researcher in the PD Project Applied Research in Vision and Computational Intelligence (PAVIC-Lab) of the partnership between UFAC, Motorola, Flextronics, and FUNDAPE.

**Thuanne Paixão** (Member, IEEE) received her Bachelor's degree in Information Systems from the Federal University of Acre (2019), a specialization in Information Security from the Estácio University Center in Ribeirão Preto (2021), and a Master's degree in Computer Science from the Federal University of Acre (2023). She has experience in areas related to artificial intelligence, robotics, and information security. She is currently participating as a researcher in the PD Project Applied Research in Vision and Computational Intelligence (PAVIC-Lab) of the partnership between UFAC, Motorola, Flextronics, and FUNDAPE.

**Ana Beatriz Alvarez** (Senior Member, IEEE) received her degree in electronic engineering from the Universidad Nacional del Altiplano, Puno-Peru, in 2000, and M.Sc. and Ph.D. degrees in electrical engineering from the University of Campinas (UNICAMP), Campinas-SP, Brazil, in 2005 and 2011, respectively. In 2012, she was a postdoctoral researcher in the Department of Computer Engineering and Automation (DCA-FEEC), UNICAMP, Brazil. Since 2013, it has been an effective research professor at the Center for Exact and Technological Sciences of the University of Acre (UFAC) Brazil, and since 2022 coordinates the Center for Applied Research in Computer Vision and Intelligence (PAVIC-Lab) of the CCET-UFAC, Rio Branco-AC, Brazil. She is the author of three book chapters and, several full scientific articles, and holds a software registration. My research interests include computational intelligence, machine learning, and computer vision. Specifically, my focus is on image restoration and synthesis, image enhancement, and analysis.

**Facundo Palomino-Quispe** (Senior Member, IEEE) received his degree in Electronic Engineering from the Universidad Privada de Tacna, Peru. Master of Science in Electronic Engineering with mention in Automation and Instrumentation and Doctor of Science: Mechatronics Engineering from Universidad Nacional San Agustín de Arequipa (UNSA), Peru. He is a professor at the Universidad Nacional de San Antonio Abad del Cusco (UNSAAC), Cusco, Peru. Director of the Institutional Laboratory for Research, Entrepreneurship and Innovation in Automatic Control Systems, Automation and Robotics - LIECAR. Research Professor RENACYT: P0049773 Level V. His main areas of interest are Mobile Robotics, Control, Automation, Artificial Intelligence and Embedded Systems.