Tiny Defect Detection for PCBs.

Rishikesh Dewalkar
Electrical and Computer Engineering
University of Michigan-Dearborn
dewalkar@umich.edu

Nikhil Pawar
College of Engineering and Computer Sciences
University of Michigan-Dearborn
inikhil@umich.edu

Saurabh Raokhande
College of Engineering and Computer Sciences
University of Michigan-Dearborn
saurabhr@umich.edu

December 16, 2024

Abstract

The fault detection in a Printed Circuit Board (PCB) is the most vital part of ensuring the reliability and performance of an electronic system. This paper, therefore, presents an automated PCB fault detection system that uses deep learning techniques with YOLOv8 and Faster R-CNN ResNet using PyTorch. In this research, six classes of faults have been considered to train the model: mouse bite, missing hole, open circuit, short, spur, and spurious copper. The strengths of YOLOv8 for real-time detection and Faster R-CNN ResNet for high-accuracy fault classification are combined to obtain precise fault localization and classification. This method simplifies the process of fault detection and reduces errors caused by human inspection, thus improving the efficiency of the quality control flow of PCBs. Results show that it is viable and effective to apply advanced computer vision for the detection of faults in PCBs, improving the quality standards of manufactured products.

1 Introduction

PCBs represent the backbone of modern electronic systems, being the base over which electrical connections and integration of components are performed. It is quite expected that fault detection in such important entities will help in the assurance of functionality and reliability in their performance. Traditional methods for PCB fault detection-which involve a great degree of manual visual inspection-have been found to be very time-consuming and error-prone and inefficient in dealing with complexities provided by today's production lines.

In recent years, deep learning and computer vision have made fault detection systems automate and accurately and efficiently inspect products. This paper will develop a deep learning-based PCB fault detection system based on two state-of-the-art models: YOLOv8 and Faster R-CNN ResNet. Both models are well-suited for fault detection tasks. YOLOv8 provides real-time object detection capabilities, whereas Faster R-CNN ResNet has better performance in fault classification accuracy[1] [2].

The six different fault categories commonly encountered in PCBs are included in the dataset used in this project: mouse bite, missing hole, open circuit, short, spur, and spurious copper. Training and testing these models on the dataset should enable the system to identify and locate the faults with high accuracy, thereby increasing the reliability and efficiency of the inspection process. It will bridge the gap from conventional fault detection to meet modern electronic manufacturing's requirements and provide a solid solution to enhance the quality control process of PCBs.

2 Related Work

Computer vision and deep learning have together taken the front seat in developing state-of-the-art methodologies for automated fault detection in PCBs. Conventional approaches include human inspection and rule-based algorithms with image processing, which suffer from several drawbacks: reliance on human expertise and susceptibility to variations in PCB design and lighting conditions. These are the major motivators that have made deep learning-based approaches, utilizing large-scale datasets and sophisticated models, the recent research focus for robust fault detection [3].

The most prominent approach involves defect classification using convolutional neural networks. Several works showed the efficiency of CNN-based models, like ResNet and VGG, for detecting and classifying defects in PCB images. Faster R-CNN, with a two-stage object detection framework, has also been widely adopted due to its ability to locate and classify PCB defects with high precision for complex fault patterns [4].

One-stage detectors, including the YOLO series, attracted attention for real-time performance. Combined with both the latest developments in speed and accuracy, the recent YOLOv8 can serve much better in rapid fault localization tasks such as in large-scale industrial manufacturing processes. Prior research showed the ability to use the YOLO series on efficient detection for common types of faults like solder bridges, missing holes, and open circuits occurring in the PCB designs [6].

Not only has the CNN-based model been adopted, but hybrid methods, which combine traditional image processing with deep learning, have also been explored. The various methods try to increase defect detection performance by incorporating handcrafted feature extraction techniques with neural networks and perform better in specific cases.

Yet, with these developments, imbalance in datasets, variation of fault appearance, and the necessity for high-resolution inspection hinder the complete automation of quality control. This paper enhances current related works by implementing YOLOv8 and Faster R-CNN ResNet, targeting six types of faults on PCBs. With the strengths combined, this system is robust in carrying out automated fault detection and furthers the development of intelligent quality control in the electronics manufacturing industry [1][4].

3 Proposed Approach

The proposed system uses state-of-the-art deep learning models, namely YOLOv8 and Faster R-CNN ResNet, for automatic detection and classification of six common PCB fault types: mouse bite, missing hole, open circuit, short, spur, and spurious copper. The approach is divided into two major parts: dataset collection and system explanation, each contributing to the overall efficiency and accuracy of the detection system.

3.1 Dataset Collection

This dataset consists of high-resolution images in both categories of fault-free and faulty PCBs. These images have been created based on the publicly available dataset enhanced with synthetic data, to consider variations in the appearance of faults and changes in environmental conditions. That being said, the images collected must undergo pre-processing operations like bounding with a box, besides being labeled concerning any fault using labeling tools like LabelImg. The collection can thus be used as training data for deep learning models. Apart from increasing variability within this dataset for helping generalization on the models in unseen fault types, some of the augmentation was performed on data by doing the rotation, flipping, scaling, and enhancing contrast of that image. It therefore divides the whole dataset into three subsets: 80% for training, 10% for validation, and 10% for testing. This balanced split will make sure that the models are adequately trained while having enough data to test and evaluate their performance [7].

3.2 System Explanation

This is done through the integration of two deep-learning models that have optimized fault detection and classification. It has used a real-time object detection model called YOLOv8 because of its high ability to detect and locate faults from PCB images with great velocity. It operates as a one-pass architecture, hence predicting location and class all in one go, thereby best fitting for real-time applications. Faster R-CNN ResNet has been used because of its good performance in detecting and classifying faults with high accuracy. The two-stage nature of this model, with the backbone ResNet, can extract intricate features from images for high-precision fault pattern detection. Transfer learning from pre-trained weights is adopted for the training of both models. It accelerates the convergence and improves the accuracy in the detection of faults. The high speed of YOLOv8 and the high precision of Faster R-CNN ResNet are just complementary, forming a powerful combination for robust fault detection. The final system can find faults in real time with high classification accuracy and is suitable for deployment on industrial PCB inspection lines [1][7].

4 Algorithm Description

The algorithm for PCB fault detection was designed by integrating the powers of two state-of-the-art deep learning models: YOLOv8 and Faster R-CNN ResNet, with complementary strengths for accurate and efficient fault detection. In this context, the algorithm processes high-resolution images of PCBs to identify and classify six specific fault types: mouse bite, missing hole, open circuit, short, spur, and spurious copper. The flow can be outlined as follows:

1. **Image Preprocessing**: Perform preprocessing necessary to prepare the images for training: resize images by size to the input requirement of YOLOv8-for example, 640x640-Faster R-CNN; normalize image data in scale between [0,1] to unify the pixel values. Data augmentation is used to increase robustness by simulating some variations that might occur with real PCBs. Various augmentations, such as random rotations, horizontal and vertical flipping, and changes in brightness/contrast, were performed to increase variety. Such steps will make the models robust against different orientations, illuminations, and defects, which in turn increases their performance on unseen data.

2. **Model 1: YOLOv8**: YOLOv8 is the state-of-the-art model in object detection, designed to process an image of a PCB in one pass to detect and classify faults. It detects objects by extracting features with a convolutional neural network, predicting bounding boxes around faults, along with confidence scores and fault labels. This single-stage design is optimized for speed, making YOLOv8 suitable for real-time applications. Confidence thresholding filters out low-confidence detections to minimize false positives. YOLOv8 is efficient in providing fast scans of PCB images and detecting faults with reasonable accuracy, especially in less complex defect scenarios [5].

3. **Model 2: Faster R-CNN ResNet**: Faster R-CNN ResNet is a two-stage object detection model for precise and detailed fault detection. The first stage makes use of a Region Proposal Network (RPN) to extract high-quality regions of interest based on the ResNet backbone's rich feature extraction. The second stage refines the bounding boxes from these regions and classifies them into the six fault types or as background. Faster R-CNN is good for detecting subtle defects such as spurious copper and mousebite, as it analyzes features in great detail. While it is slower than YOLOv8, it offers very good accuracy and is thus important for critical fault validation [4].

4. **Integration and Ensemble Approach**: The integration of YOLOv8 and Faster R-CNN balances the trade-off between speed and precision in fault detection. YOLOv8 works as the main detector for fast, real-time detection of PCB defects, while Faster R-CNN confirms and increases the accuracy by handling complex and minute faults. The output is then combined with an ensemble approach, where higher confidence is given to the faults detected by both models. Discrepancies in the predictions are resolved, taking advantage of the high precision of Faster R-CNN for comprehensive and reliable fault detection in industrial applications [2].

5. **Post-Processing**: The combined outputs undergo post-processing to finalize the detections. NMS removes the redundant bounding boxes of the same fault and keeps the one with higher confidence. The final outputs include annotated PCB images showing bounding boxes, labels, and confidence levels, as well as a detailed textual report of fault types, coordinates, and confidence scores. From these, one gets crystal clear insights into the detected defects, thus making them actionable by industrial-quality teams.

6. **Performance Optimization**: To ensure the highest possible detection accuracy, transfer learning is applied to initialize models with pre-trained weights from large datasets like COCO and ImageNet. Loss functions such as bounding box regression loss and cross-entropy loss are optimized for accurate bounding box detection and fault classification. Careful tuning of hyperparameters including learning rate, batch size, and epoch count has been done in order to strike a balance between precision, recall, and computational efficiency. These optimizations ensure that the models perform well on the PCB fault detection task without overfitting.

7. **Evaluation Metrics**: The performance of the algorithm is analyzed on several key metrics. Precision measures the model's ability to reduce false positives, while recall measures its ability to reduce false negatives. F1-score is a proper measure that balances precision and recall; thus, it gives an all-round view of accuracy in detection. Mean Average Precision estimates the overall effectiveness of detecting and locating various types of faults. With systematic analyses based on these metrics, models are refined to ensure robust industrial-grade fault detection capabilities.

5 Model Architecture

The architecture of the PCB fault detection model leverages the speed and efficiency of YOLOv8 with the precision and depth of Faster R-CNN ResNet to ensure robust fault detection for several types of defects. This is elaborated on in detail below.

5.1 YOLOv8 Architecture

YOLOv8 is the state-of-the-art single-stage object detection model, optimized for real-time performance with high accuracy. Its architecture consists of three main parts:

- **Backbone: Feature Extraction** The backbone is a deep convolutional neural network, usually based on CSPDarkNet (Cross-Stage Partial Network), which extracts low-level and high-level features from the input PCB images. The backbone uses CSP modules to improve the gradient flow by reducing redundancy and minimizing computation. This makes the network lightweight and effective. The module processes the images to enhance the important regions and patterns, such as edges of the defects like mouse bites or spurious copper.

- **Neck: : Feature Aggregation** It applies the Path Aggregation Network to enhance multi-scale feature learning in the Neck. PANet fuses features selected from deeper layers of the backbone, guaranteeing that the model perceives a wide range of defect sizes, from tiny spurs to larger ones like missing holes. The neck focuses on preserving both spatial and semantic information, which is critical for defect classification and localization.

- **Head: Fault Detection** It predicts the coordinates of bounding boxes, class labels corresponding to six types of faults, and their confidence scores. YOLOv8 depends on anchor-free mechanisms that make the predictions flexible and less computational. The head allows classification using logistic regression and refinement of the bounding box with regression techniques. Its fast process makes YOLOv8 suitable for real-time PCB fault detection [2].

5.2 Faster R-CNN ResNet Architecture

Faster R-CNN is a two-stage object detection model, mainly for high precision. Unlike the one-pass detection in YOLOv8, Faster R-CNN focuses on regional proposals and detailed analysis, thus being more appropriate for the detection of subtle defects in PCB images. The architecture includes:

- **Backbone: ResNet** ResNet pretrained on large datasets such as ImageNet is utilized as the backbone to extract hierarchical features from the PCB images. The skip connections solve the vanishing gradient problem of ResNet, enabling the model to go deeper and capture more intricate details. This capability is of utmost importance in detecting those defects that are not visually salient, such as open circuits or spurious copper.

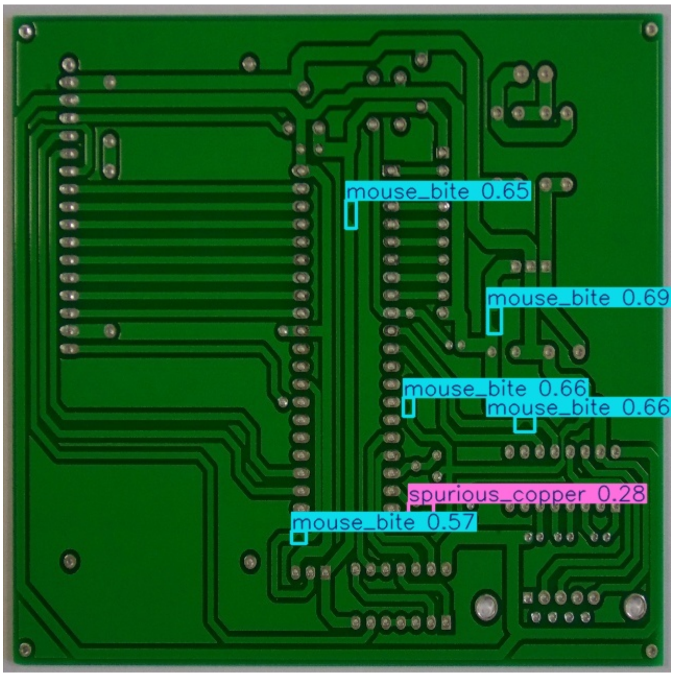

Figure 1: Fault Prediction

- **Region Proposal Network (RPN):** RPN of the RPN finds the RoIs from the feature maps generated by ResNet. These are regions that are likely to contain faults. It makes use of anchors, and predefined bounding boxes, and scores them with an "objectness score" in order to propose candidate regions. Overlapping regions are merged using non-maximum suppression (NMS), reducing the redundancies and keeping only the most relevant proposals.

- **RoI Pooling** Once the RoIs have been identified, a pooling layer extracts fixed-size feature maps for each region; this makes the features consistent in their representation regardless of the size of the region. This simplifies further processing while guaranteeing very high accuracy in analyzing each fault candidate.

- **Bounding Box Refinement and Classification** In the second stage, RoI feature maps are processed to refine bounding boxes and classify the detected regions into one of the six fault types mouse bite, spur, etc.- or as background. The coordinates of the bounding box are further refined, and classification confidence scores are obtained by softmax layers. This two-stage processing ensures exceptional precision in defect detection [1].

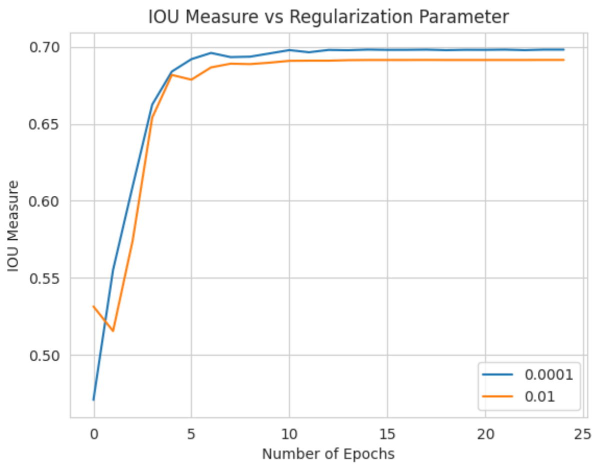

Figure 2: IOU Measure vs Regularization Parameter of RCNN

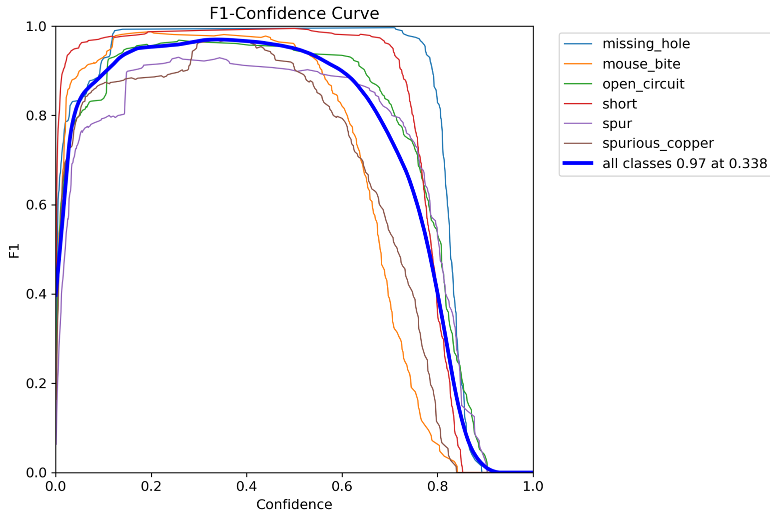

Figure 3: YOLO v8

6 Working of the Model for PCB Fault Detection

In this project, two different deep learning models, namely YOLOv8 and Faster R-CNN ResNet, were independently implemented to detect six types of faults in PCBs: mouse bite, missing hole, open circuit, short, spur, and spurious copper. The models take high-resolution PCB images for their analysis, employing different architectures, each with a performance profile for object detection to meet different needs optimized for speed and the other for accuracy and precision. Each model will be explained in detail below.

6.1 Model 1: YOLOv8 Workflow Input and Preprocessing:

First, the pipeline takes the raw images of PCBs as input. It starts with resizing them into a standard dimension, such as 416 × 416 pixels, to suit the network while preserving aspect ratios.

Then, normalization is used, scaling the pixel values to the range [0,1], which reduces computational complexity and speeds up model convergence. During training, aggressive data augmentation is used, including random rotation, flipping, cropping, and changing brightness. These are used to increase the generalization capability of the model, such that it performs well in changing lighting conditions, angles, and environmental conditions.

6.2 Feature Extraction and Object Detection:

Once pre-processing is complete, the image goes through the YOLOv8 CSPDarkNet backbone. It extracts essential features hierarchically from it, with which the model becomes capable of learning from fine-grained details such as sharp edges of spur defects to much bigger spatial features.

YOLOv8 does not require anchors, as it operates under an anchor-free architecture without any pre-defined anchor boxes; thus, detection will be more effective and enhanced to identify faults in various shapes and sizes.

The whole image is divided into a grid, and every grid cell predicts: a bounding box for the fault's location, a class label: spur, open circuit, and a confidence score of the detected region being a fault. After detection, YOLOv8 makes an output with bounding boxes, labels of the classes predicted, and their corresponding confidence scores. This information overlays on the input image such that one can clearly visualize the detected faults.

The major advantage of YOLOv8 is real-time processing, which makes it applicable for applications requiring speed, such as on-production-line quality control in PCB manufacturing. It is particularly good at detecting a variety of faults that are very prominent and well-defined, like mouse bite, spur, and short, because of its fast grid-based object detection mechanism [2].

6.3 Model 2: Faster R-CNN ResNet Workflow Faster R-CNN:

Faster R-CNN ResNet Workflow Faster R-CNN, or Regions with Convolutional Neural Networks, is coupled with a ResNet backbone-one of the most powerful object detection models that combines accuracy with flexibility. It follows a two-stage detection process: first, it proposes the regions most likely to contain objects (faults); then, it goes into details with classification and precise localization of those regions. That makes it very dependable, especially for detecting those minute faults in PCBs like spurious copper or open circuits that require a detailed look. A detailed explanation of the steps involved in Faster R-CNN ResNet will be discussed below.

6.3.1 Input and Preprocessing

The high-resolution images of the PCB are first fed into the model as input. These images are then preprocessed to make them acceptable to the model. Among the preprocessing steps are the following:

Image Resizing: The original images are resized to sizes like 800x800 pixels, preparing the data for model processing without distorting the aspect ratio. This step of resizing is quite important in maintaining consistency while preserving the details of the fault. Normalization: Pixel values of every image are scaled within a fixed range such as [0, 1] for faster and stable training. It helps in maintaining the numerical stability of the deep convolution layers. Further, data augmentation at random rotation, flipping, and cropping is done while training. This augmentation makes the model robust against different perspectives and distortions that may appear in the PCB images due to variations in real-world lighting conditions, camera angles, or noise [1]. Stage I: Region Proposal Network (RPN) Faster R-CNN starts with the Region Proposal Network (RPN), which is a crucial component responsible for highlighting regions in the image that have a high probability of having faults.

The RPN slides a small network over the feature maps generated by the ResNet backbone. Each position generates several region proposals called anchors (bounding boxes of various sizes and aspect ratios). The RPN classifies these anchors as either containing an object (fault) or being background. It then refines the coordinate of the proposed bounding boxes with low-confidence anchor discards. The IoU threshold removes regions with poor object overlap. This step outputs a reasonably small number of Region of Interest (RoI) proposals for further processing, in contrast to processing the full image, which would have significantly raised the computational cost.

6.3.2 Stage 2: Feature Extraction via ResNet Backbone:

After that, these obtained RoIs are input into the ResNet backbone serving as the feature extractor. The ResNet, or residual network, is a deep convolutional neural network architecture developed for alleviating the problem of vanishing gradients to make possible the training of very deep neural networks. This architecture realizes such functionality through the introduction of skip connections that enable gradients to directly flow through network layers.

7 Key steps involved in this stage:

RoIs from the ResNet extracts hierarchical features ranging from low-level fine details (for example, edges and patterns) to higher conceptual representations for fault kinds, which are characterized by features as missing holes and spurious copper. Their properties of invariance ensure strong representivity that would offer saliency in telling fault kinds showing subtle appearances and/or sizes. ROI Pooling The output from the ResNet backbone undergoes RoI pooling, which ensures that all the RoIs are converted into the same fixed size, e.g., 7×7. This step is important because the input sizes of the RoIs vary with their different spatial locations and sizes on the PCB image. These regions are standardized so the model can process them uniformly in subsequent layers.

7.1 Fault Classification and Localization

The pooled features are fed into fully connected layers for two tasks: Fault Classification: Each RoI is assigned a class label, such as mouse bite, missing hole, short, spur, etc., using a softmax activation function that ensures the model outputs the probability of the RoI belonging to each fault category. Bounding Box Regression: Simultaneously, the model refines the bounding box coordinates to achieve precise localization. This bounding box refinement ensures that the detected region closely matches the fault's exact location on the PCB. This two-step refinement process—classification and localization—enables the model to achieve a high degree of accuracy and precision, even for complex or small faults.

7.2 Output Generation

After classification and bounding box regression, the model outputs:

The set of bounding boxes around the detected faults. Class labels for every kind of fault. Confidence score regarding the detection, expressing confidence about the model's verdicts. These are highlighted on the input PCB image through fault highlighting with labeled bounding boxes and confidence scores. On top of that, a thorough summary of the fault types, locations, and their numbers is also provided by this very model for further analysis [1].

8 Conclusion

In the end, the proposed deep learning-based PCB fault detection system effectively overcomes the shortcomings of traditional manual inspection methods by combining the strengths of YOLOv8 and Faster R-CNN ResNet. It enhances the reliability and efficiency of the inspection process by accurately identifying and locating six common types of PCB faults, thus meeting the demands of modern electronic manufacturing. This approach not only enhances quality control but also lays the foundation for incorporating sophisticated AI-driven solutions into the workflow for producing PCBs to ensure even higher standards of performance and reliability.

9 References

1. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems (NeurIPS), 28.

2. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. CVPR.

4. Lin, T. Y., et al. (2017). Feature Pyramid Networks for Object Detection. CVPR.

5. Iqbal, M. Z., et al. (2020). Automatic Detection and Classification of PCB Defects Using CNNs. IEEE Access.

6. Paszke, A., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library.

7. Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.