INTERNATIONAL JOURNAL OF ELECTRONICS AND COMPUTER APPLICATIONS

ORIGINAL ARTICLE

AI-Based Fault Detection in PCB Layout Using Patch-Level CNN Classification and Grad-CAM Visualization

Deepali Misal[1*], Ganesh Ingale[1], Yash Chaudhari[1], Aditya Deokar[1], Siddharth Awate[1]

1 Electronics and Telecommunication, Keystone School of Engineering, SPPU, Pune, Maharashtra, India.

OPEN ACCESS
Received: 28-07-2025
Accepted: 23-11-2025
Published: 12-12-2025

Citation: Misal D, Ingale G, Chaudhari Y, Deokar A, Awate S. AI-Based Fault Detection in PCB Layout Using Patch-Level CNN Classification and Grad-CAM Visualization. 2025; 2(2):70-73. https://doi.org/10.70968/ijeaca.v2i2.ML116

* **Corresponding author.**
deepali.misal@keystonesoe.in

Funding: None

Competing Interests: None

ISSN
Electronic: 3048-8257

Abstract

Automated quality inspection of Printed Circuit Boards (PCBs) is a critical requirement in modern electronics manufacturing. Traditional manual inspection methods are slow, inconsistent, and error-prone, while industrial Automated Optical Inspection (AOI) machines are prohibitively expensive for small and medium enterprises. This paper presents an AI-based PCB fault detection system using a custom Convolutional Neural Network (CNN) trained on the DeepPCB benchmark dataset from Peking University. The proposed approach introduces XML annotation-guided patch extraction that crops 128×128 pixel regions centered on annotated defect locations, enabling the classifier to focus exclusively on defect morphology rather than irrelevant background regions. The system classifies seven categories: missing hole, mouse bite, open circuit, short circuit, spur, spurious copper, and no-defect. Data augmentation expanded the training set from 2,953 to 25,472 samples. The model achieves 9697% validation accuracy on 680 unseen samples, outperforming comparable CNN-based methods in the literature. Grad-CAM (Gradient-weighted Class Activation Mapping) provides visual explainability by highlighting defect locations with heatmap overlays. The complete system is deployed as a Flask web application with a Neon PostgreSQL cloud database for persistent inspection logging. Experimental results demonstrate the feasibility of replacing expensive hardware AOI systems with an affordable, software-based AI inspection solution.

Keywords: PCB defect detection, Convolutional neural network, Grad-CAM, Patch-based classification, Automated optical inspection, Deep learning, Quality control

Introduction

Printed Circuit Boards (PCBs) are the fundamental building blocks of virtually every electronic device — from consumer smartphones to medical equipment and aerospace systems. The global PCB market exceeded USD 75 billion in 2023, with defect-free manufacturing being a critical quality requirement across all application domains[(1)].

During PCB fabrication, several types of manufacturing defects occur due to imperfections in chemical etching, photolithography, drilling, and copper deposition processes. These defects — if undetected — cause product failures, safety hazards, and costly recalls. A single defective PCB in a medical device or automotive control system can have catastrophic consequences.

Current industrial practice employs Automated Optical Inspection (AOI) machines for PCB quality control. These systems achieve 98–99% detection accuracy but cost between INR 4 million to INR 20 million, making them inaccessible to the majority of small and medium PCB manufacturers, particularly in developing economies like India. Manual inspection, the primary alternative, achieves only 80–85% accuracy at 2–3 minutes per board[(2)].

This paper makes the following technical contributions:

1. A novel XML annotation-guided patch extraction methodology that crops precise 128×128 defect-centric patches using bounding box coordinates, dramatically improving classification accuracy over whole-image approaches.
2. A custom 7-class CNN architecture specifically designed for DeepPCB-style images (dark background, white copper traces), achieving 99.85% validation accuracy.
3. Grad-CAM explainability integrated into a production Flask web application with persistent cloud-based inspection history via Neon PostgreSQL.
4. Comprehensive evaluation including per-class precision, recall, F1-score, and confusion matrix analysis on the standard DeepPCB benchmark.

Related Work

PCB defect detection has been approached through traditional image processing and modern deep learning methods. Early works used morphological operations and template matching[3], which required perfect image registration and failed under illumination variations.

Ding *et al.*[4] proposed a multi-scale CNN achieving 92.3% accuracy on the DeepPCB dataset but required significant computational resources. Huang *et al.*[5] applied transfer learning with VGG-16 achieving 94.7% accuracy. Recent YOLO-based object detection approaches[6] detect multiple defects simultaneously but require substantial training data and GPU inference hardware.

A key limitation of existing work is the use of *whole image classification*, where full 3034×1586 pixel PCB images are resized to 224×224, causing defects occupying as few as 70×70 pixels (0.05% of image area) to become sub-pixel after resizing. Our patch-based approach directly addresses this fundamental limitation.

Proposed Methodology

A. Dataset

The DeepPCB dataset[4] released by Peking University contains 1,386 high-resolution PCB images (3034×1586 pixels) captured with a 16-megapixel industrial camera. The dataset includes 2,953 annotated defects across six categories with bounding box coordinates in PASCAL VOC XML format. Additionally, 10 defect-free reference PCB images were used to construct the seventh “no-defect” class.

B. XML Annotation-Guided Patch Extraction

The core novelty of our approach is the patch extraction methodology. For each annotated defect bounding box with coordinates (x_{min}, y_{min}, x_{max}, y_{max}), we compute the defect centroid:

$$c_x = \frac{x_{min}+x_{max}}{2}, c_y = \frac{y_{min}+y_{max}}{2} \quad (1)$$

A 128×128 patch is then extracted centered at (c_x, c_y) with 40-pixel padding:

$$P = I[c_y - 104 : c_y + 104, c_x - 104 : c_x + 104] \quad (2)$$

where I is the original PCB image. This ensures each patch contains the complete defect morphology with sufficient surrounding context for classification. For the no-defect class, random patches are extracted from defect-free reference images.

C. Data Augmentation

Given the limited size of the DeepPCB dataset (115– 116 images per class), we apply seven augmentation operations to each training patch: horizontal flip, vertical flip, 90 ° rotation, 180 ° rotation, 270 ° rotation, brightness variation ($\alpha \in [0.7,1.3]$), and contrast variation ($\beta \in [0.7,1.3]$). This expands the training set from 2,021 to 25,472 patches.

D. CNN Architecture

The proposed CNN architecture, summarized in (Table. 1), consists of four convolutional blocks followed by global average pooling and fully connected layers. The architecture is specifically designed for DeepPCB-style images with dark backgrounds and high-contrast copper traces.

Each convolutional layer uses ReLU activation and is followed by Batch Normalization (BN). Global Average Pooling replaces Flatten to reduce parameters and overfitting. The final Dense layer uses softmax activation for 7-class probability output.

E. Training Configuration

The model was trained using the following hyperparameters:

- Optimizer: Adam with initial learning rate $\eta = 0.001$
- Loss function: Categorical cross entropy
- Batch size: 64
- Epochs: 30 (early stopping at epoch 25)
- Learning rate schedule: ReduceLROnPlateau (factor=0.3, patience=3)
- Normalization: Per-sample mean-std normalization
- Hardware: NVIDIA T4 GPU (Google Colab)
- Training time: ≈30 minutes

Table 1: Proposed CNN Architecture

Layer	Output Shape	Parameters
Input	128×128×3	0
Conv2D (32) + BN	128×128×32	928
Conv2D (32) + BN	128×128×32	9,280
MaxPool + Dropout(0.2)	64×64×32	0
Conv2D (64) + BN	64×64×64	18,496
Conv2D (64) + BN	64×64×64	36,928
MaxPool + Dropout(0.2)	32×32×64	0
Conv2D (128) + BN	32×32×128	73,856
Conv2D (128) + BN	32×32×128	147,584
MaxPool + Dropout(0.25)	16×16×128	0
Conv2D (256) + BN	16×16×256	295,168
MaxPool + Dropout(0.25)	8×8×256	0
GlobalAvgPool	256	0
Dense(256) + BN + Drop(0.4)	256	65,792
Dense(128) + Dropout(0.3)	128	32,896
Dense(7) Softmax	7	903
Total Parameters		684,711

F. Grad-CAM Visualization

For model explainability, we implement Grad-CAM[7]. Given a predicted class c, the importance weight for feature map k in the last convolutional layer is:

$$\alpha_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (3)$$

The class-discriminative localization map is:

$$L_{\text{Grad}-CAM}^c = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$

where A^k are the activation maps of the last convolutional layer (conv2d 6 with 256 filters). The resulting heatmap is resized to input dimensions and overlaid on the original patch using a jet colormap with 55% heatmap opacity.

G. System Deployment

The complete system is deployed as a Flask web application. The backend loads model weights, performs preprocessing, runs inference, generates Grad-CAM, and saves results. A Neon PostgreSQL cloud database stores all prediction records including timestamp, defect class, confidence, severity, and image filenames for persistent inspection logging accessible from any device.

Experimental Results

A. Dataset Split

Table 2: Dataset Distribution After Patch Extraction and Augmentation

Class	Raw	Train	Val	Test
Missing Hole	497	2,776	74	76
Mouse Bite	492	2,752	73	75
Open Circuit	482	2,696	72	73
Short Circuit	491	2,744	73	75
Spur	488	2,728	73	74
Spurious Copper	503	2,816	75	76
No Defect	800	3,960	240	240
Total	3,753	20,472	680	689

B. Training Performance

The model converged at epoch 25 with early stopping. (Table. 3) shows the training progression at key epochs.

Table 3: Training Progression at Key Epochs

Epoch	Train Acc	Val Acc	Train Loss	Val Loss
1	49.63%	35.29%	1.2557	18.8674
4	55.02%	63.86%	1.0272	0.8569
7	89.67%	81.82%	0.2861	0.5619
8	94.28%	97.27%	0.1627	0.0964
13	99.49%	99.85%	0.0151	0.0059
25	99.92%	99.55%	0.0029	0.0199

C. Classification Performance

(Table. 4) presents per-class precision, recall, and F1-score on the held-out test set of 689 samples.

Table 4: Per-Class Classification Report on Test Set

Class	Precision	Recall	F1-Score
Missing Hole	0.9987	1.0000	0.9993
Mouse Bite	0.9987	0.9987	0.9987
Open Circuit	0.9986	0.9986	0.9986
Short Circuit	1.0000	0.9987	0.9993
Spur	0.9986	0.9986	0.9986
Spurious Copper	0.9987	0.9987	0.9987
No Defect	0.9958	0.9958	0.9958
Weighted Avg	0.9984	0.9984	0.9984

D. Comparison with Existing Methods

The proposed patch-based CNN significantly outperforms whole-image classification methods. The baseline ResNet50 trained on whole images achieves only 78.4% accuracy, demonstrating the critical importance of our patch extraction methodology. The 7.05% improvement over the best comparable method (YOLOv5 at 96.8%) validates the effectiveness of annotation-guided defect-centric patch classification.

Table 5: Comparison With State-Of-The-Art Methods on DeepPCB

Method	Approach	Accuracy
Ding *et al.* (4)	Multi-scale CNN	92.3%
Huang *et al.* (5)	VGG-16 Transfer	94.7%
Zhang *et al.* (6)	YOLOv5 Detection	96.8%
Baseline CNN (whole image)	ResNet-50	78.4%
Proposed Method	Patch CNN + Grad-CAM	99.85%

E. System Performance

Table 6: System Performance Metrics

Metric	Value
Validation Accuracy	99.85%
Test Accuracy	99.84%
Weighted F1-Score	0.9984
Model Parameters	684,711
Model Size	8.3 MB
Training Time (T4 GPU)	≈30 minutes
Inference Time (CPU)	3–5 seconds
Inference Time (GPU)	<100 ms

Discussion

A. Impact of Patch Extraction

The most significant contribution of this work is the XML annotation-guided patch extraction methodology. The fundamental problem with whole-image classification is that PCB defects occupy only 0.05% of the total image area. When a 3034×1586 image is resized to 128×128, a 70×70 defect becomes sub-pixel. Our approach eliminates this problem by providing the classifier with 128×128 images where the defect occupies the majority of the frame.

B. Limitations

The proposed system has two primary limitations. First, it requires XML bounding box annotations to locate defects in full PCB images, making it unsuitable for uninspected PCBs without prior annotations. Second, the model is domain specific — trained exclusively on DeepPCB-style images (dark background, bare copper traces before component assembly) and does not generalize to green solder-mask PCBs or assembled boards.

C. Future Work

Future extensions include: (1) YOLO-based object detection for annotation-free full-image defect localization; (2) domain adaptation for green solder-mask PCBs; (3) real-time video stream inspection using a camera feed; (4) federated learning for multi-factory model training without data sharing; and (5) cloud deployment on Render.com for browser-accessible inspection.

Conclusion

This paper presented an AI-based PCB fault detection system achieving 96-97% validation accuracy on the DeepPCB benchmark dataset. The key innovation is XML annotation guided patch extraction that focuses the classifier on defect centric 128×128 regions, eliminating the fundamental limitation of whole-image approaches where defects become subpixel after resizing. The integration of Grad-CAM provides visual explainability critical for industrial trust and adoption. The complete system — including CNN model, Grad-CAM visualization, Flask web application, and Neon PostgreSQL cloud logging — demonstrates the feasibility of replacing expensive hardware AOI machines (INR 4M–20M) with an affordable software-based AI inspection solution accessible to small and medium PCB manufacturers. The 6.05% accuracy improvement over the best comparable published method validates the effectiveness of the proposed patch-based approach.

References

1. Grand View Research. *Printed Circuit Board Market Size Report* 2023. *Printed Circuit Board Design Techniques for EMC Compliance*. . Available from: https://www.grandviewresearch.com/industryanalysis/printed-circuit-board-market
2. . Adibhatla VA, Chih HC, Hsu CC, Cheng J, Abbod MF, Shieh JS. Defect Detection in Printed Circuit Boards Using You-Only-Look-Once Convolutional Neural Networks. *Electronics*. 2020;9(9):1547. Available from: 10.3390/electronics9091547
3. Bhatt S, Kumari A. PCB defect detection using image processing techniques. *International Journal of Computer Applications*. vol. 168, no. 11, pp. 1–5, 2017.
4. Ding R, Dai L, Li G, Liu H. TDD-net: a tiny defect detection network for printed circuit boards. *CAAI Transactions on Intelligence Technology*. 2019;4(2):110-116. Available from: 10.1049/trit.2019.0019
5. Huang W, *et al.* PCB defect detection using transfer learning with VGG-16. *Proc. IEEE ICIEA*, 2020, pp. 1–6.
6. Zhang X, *et al.* Automatic defect detection of PCBs based on YOLO. *Applied Science*, vol. 12, no. 14, p. 7221, 2022.
7. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017;:618-626. Available from: 10.1109/iccv.2017.74
8. Tang S, He F, Huang X, Yang J. Online PCB Defect Detector On A New PCB Defect Dataset. *arXiv:1902.06197*. 2019;. Available from: 10.48550/arXiv.1902.06197