Deep Learning Based PCB Defect Classification

Paramjot Singh ee250002045

1 Introduction

Printed Circuit Board (PCB) defects can severely affect the performance and reliability of electronic devices. This project focuses on identifying and classifying PCB defects using deep learning techniques to reduce manual inspection and human error.

2 Objective

The objective of this project is to automatically classify PCB images as defective or non-defective, and further identify the type of defect such as mouse bite, missing hole, spur, etc. Automated inspection helps improve accuracy and efficiency in PCB manufacturing industries.

3 Types of PCB Defects

3.1 Mouse Bite Defect

Mouse bite is a PCB defect where a part of the copper trace is chipped off, resembling a bite.

3.2 Missing Hole Defect

Missing hole defect occurs when the required drill hole for a lead component is absent.

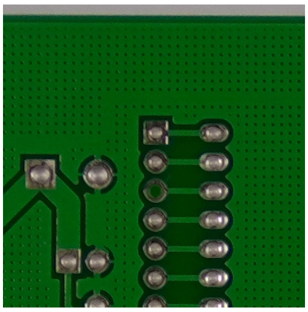

Figure 1: Missing Hole Defect

3.3 Open Circuit Defect

An open circuit defect is caused by a break in the conductive copper path, preventing current flow.

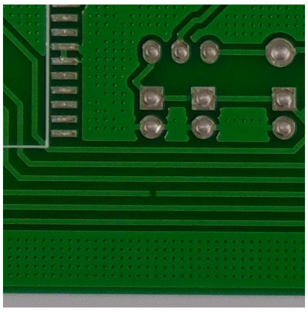

Figure 2: Open Circuit Defect

3.4 Short Circuit Defect

Short circuit defect occurs when unintended electrical connections form between copper paths.

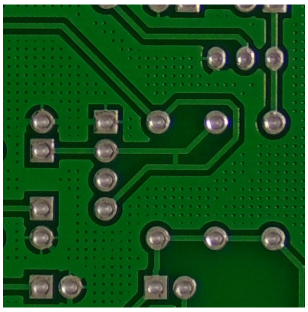

Figure 3: Short Circuit Defect

3.5 Spur Defect

Spur defect is characterized by unwanted copper projections left on the PCB surface.

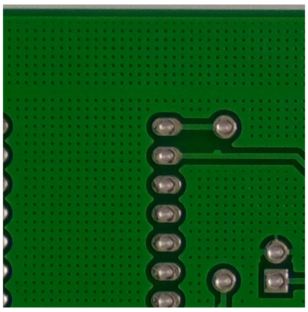

Figure 4: Spur Defect

3.6 Spurious Copper Defect

Spurious copper defect involves unwanted copper patches that may cause short circuits.

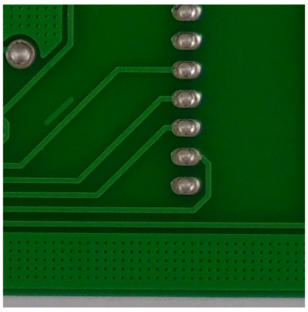

Figure 5: Spurious Copper Defect

4 Justification for Using Deep Learning

PCB defects are often small, subtle, and visually similar, making manual inspection unreliable. Traditional image processing methods require handcrafted features and precise conditions. Deep learning models automatically learn complex features and provide better generalization.

5 Dataset Creation and Augmentation

High-resolution PCB images often contain multiple defects distributed all across the board. To tell about individual defect regions precisely, a custom interactive cropping tool was developed using OpenCV and Tkinter.

The tool allows manual selection of defect locations, followed by automatic generation of multiple cropped image patches using spatial jittering. For each selected defect, three 600×600 image patches were generated by randomly shifting the crop center within a given range. This approach increases dataset diversity while preserving defect-centric information.

Defect Type	**Image Count**
Mouse Bite	1059
Missing Hole	1194
Short Circuit	969
Open Circuit	984
Spur	1101
Spurious Copper	1068
Total	6375

Table 1: Dataset Distribution

6 PreProcessing and Data Splitting

All images were resized to 224×224 pixels to match the input requirements of the neural network. To improve generalization and prevent overfitting, data augmentation techniques such as horizontal flipping, vertical flipping, random rotations, and slight brightness variations were applied during training.

The dataset was split into training and validation sets in an 80:20 ratio.

7 Model Design

7.1 Model Architecture

A ResNet-50 architecture pre-trained on ImageNet was used for defect classification. Residual connections allow effective training of deep networks by ditching vanishing gradient issues.

The final fully connected layer was replaced with a custom classification head consisting of a dense layer, ReLU activation, dropout for regularization, and an output layer matching the number of defect classes.

7.2 Training Strategy

The model was trained using the Adam optimizer with a learning rate of 1×10^{-4}. Cross-entropy loss was used as the objective function. Training was performed for 15 epochs with a batch size of 32.

Model performance was evaluated on a held-out validation set after training completion.

8 Evaluation

Model performance was evaluated using precision, recall, F1-score, and confusion matrices. Additionally, class-wise precision-recall curves were generated to analyze performance across defect categories.

Both validation and test sets were evaluated separately to ensure robustness and avoid data leakage.

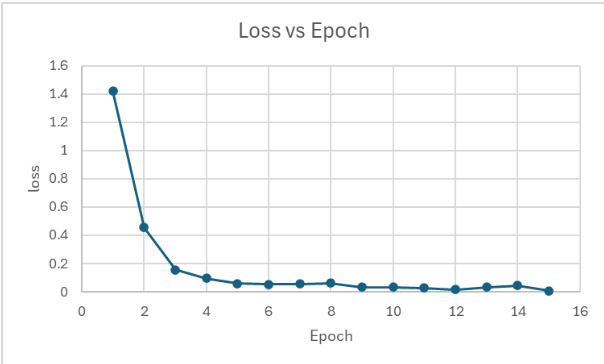

Figure 6: Loss vs Epoch

Defect Type	Precision	Recall	F1-score
Missing Hole	1.00	0.99	1.00
Mouse Bite	0.99	0.93	0.96
Open Circuit	0.97	0.98	0.98
Short Circuit	1.00	0.99	0.99
Spur	0.94	1.00	0.97
Spurious Copper	0.98	0.99	0.98

Table 2: Class-wise performance of the fine-tuned ResNet-50 model

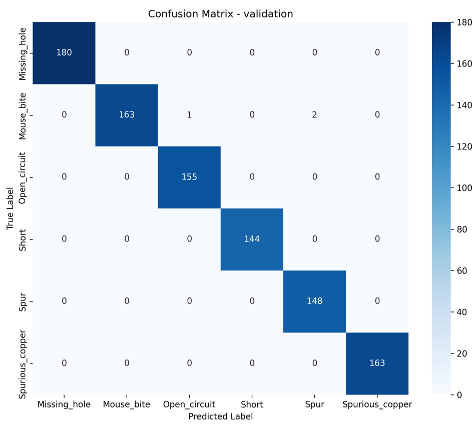

Figure 7: Validation set

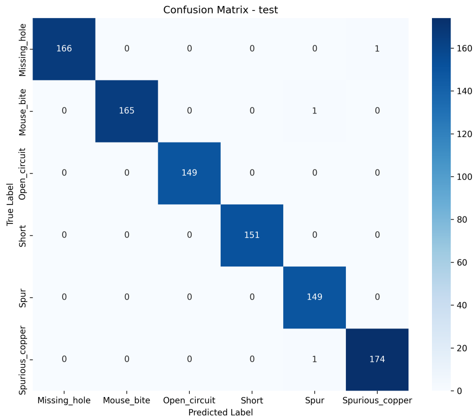

Figure 8: Test set

9 Model Explainability

To interpret model predictions, Gradient-weighted Class Activation Mapping (Grad-CAM) was used. Heatmap was used in order to predict the localized defect.

An interactive inspection tool was developed to visualize defect localization on full PCB images, allowing qualitative verification of model attention.

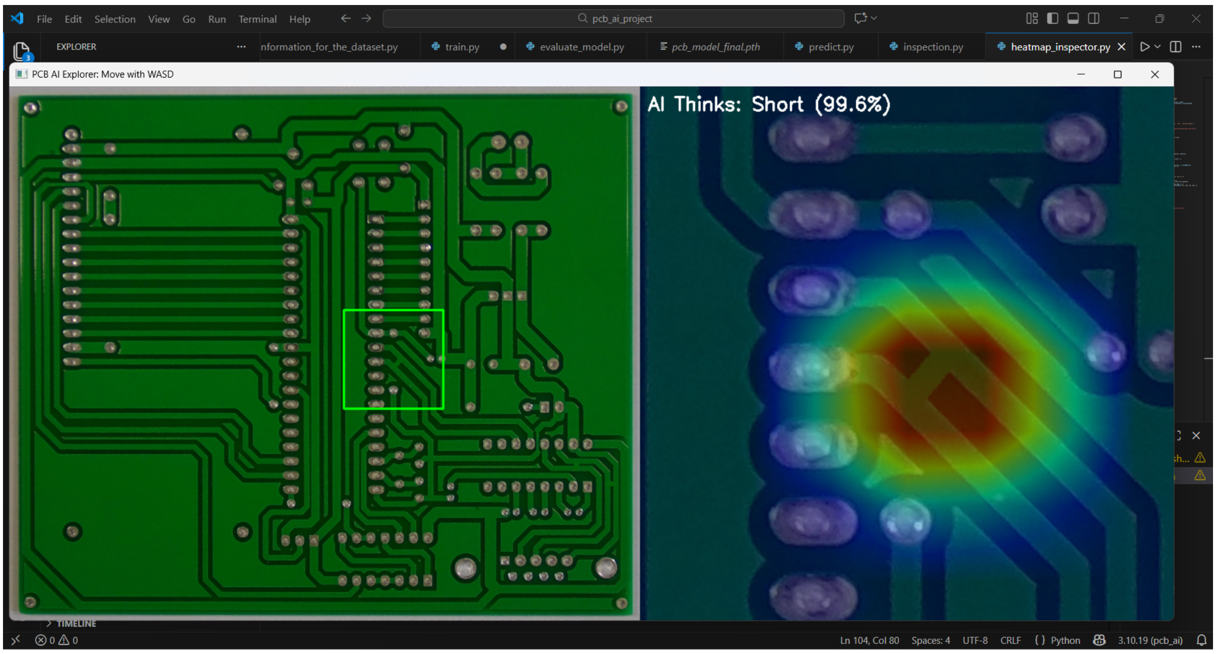

Figure 9: Heatmap View

10 Conclusion

This work demonstrates an end-to-end deep learning pipeline for PCB defect classification, from dataset creation to model explainability. The results indicate that deep learning-based inspection systems can significantly improve accuracy and reliability compared to manual inspection methods.

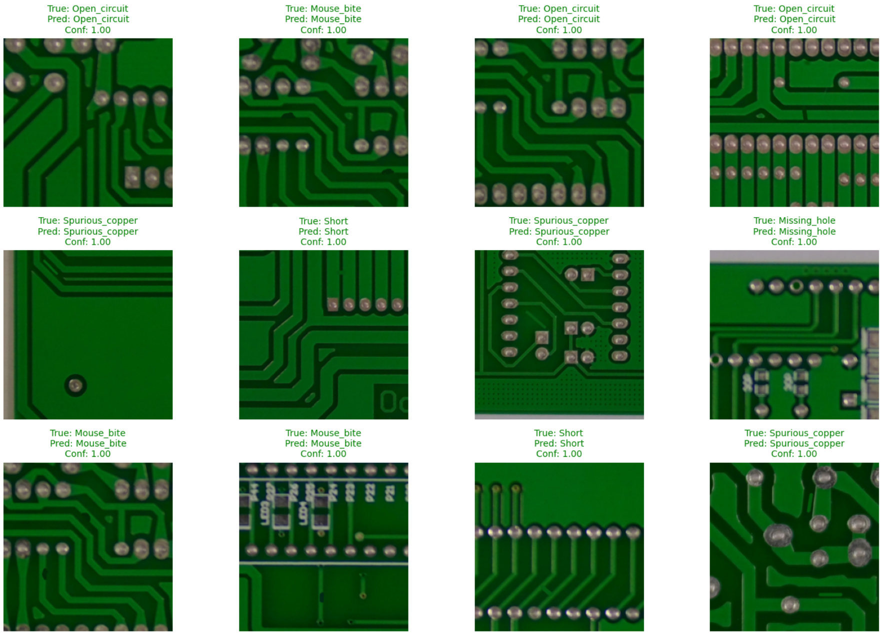

Figure 10: A few examples