PROJECT ABSTRACT

PCB Defect Detection using CNN

Automated optical inspection for bare PCBs: template-aligned defect detection and classification across the six DeepPCB defect types (open, short, mousebite, spur, pinhole, spurious copper) with bounding-box localization. Delivered built-to-order with training notebook, inference pipeline, weights, evaluation report, web demo, PPT and viva kit.

01. Title

PCB Defect Detection using CNN.

02. Introduction / Background

Manual visual inspection of printed circuit boards is slow, tiring and inconsistent — inspectors miss hairline opens and tiny solder bridges after hours at the microscope. Automated optical inspection (AOI) compares each board against a golden reference and flags deviations. The DeepPCB dataset (Huang et al., Huawei Noah's Ark Lab) provides 1,500 template/tested image pairs annotated with bounding boxes for six defect classes, mirroring how real AOI stations work: align, difference, classify. This project implements that pipeline with a modern single-shot detector.

03. Problem Statement

Small PCB fabrication units cannot afford commercial AOI machines, yet ship defects — opens, shorts, mousebites — that cause field failures. Students learning industrial AI need a realistic defect-detection project with genuine annotations rather than toy data. This project trains a detector on DeepPCB's 1,000 training pairs, evaluates per-class mAP@0.5 on the 500-pair test split, and delivers an inspection demo that localizes defects with boxes, severity and rework actions.

04. Objectives

- Align tested board images to their golden templates via feature matching.
- Train a YOLO-style detector on DeepPCB for six defect classes with bounding boxes.
- Apply augmentation (mosaic, HSV jitter, affine) suited to PCB imagery.
- Evaluate per-class and overall mAP@0.5 on the 500-pair test split.
- Build an inspection demo: upload a board photo, get defect boxes, severity and rework guidance.

05. Proposed Methodology

(1) Each tested image is registered to its template with ORB feature matching + homography so comparison is pixel-aligned. (2) A YOLO-style single-shot detector with a lightweight backbone (CSPDarknet-tiny or MobileNet, configurable) is trained on 640×640 inputs to predict boxes + classes. (3) Mosaic and photometric augmentation simulate lighting and placement variation. (4) Non-maximum suppression filters overlapping boxes at inference. (5) Each detection maps to a severity level and a rework action (e.g. solder bridge → desoldering wick) for the operator UI.

06. System Architecture / Working

The pipeline has three stages: alignment (template matching produces a registered pair), detection (the CNN outputs candidate boxes with class scores), and decision (NMS + confidence threshold yields the final defect list). The demo overlays boxes on the board photo with class and confidence labels, computes a PASS/FAIL verdict, and lists per-defect severity and rework actions. Because the detector learns the template-vs-tested difference rather than memorizing boards, it generalizes to unseen layouts — the central claim the evaluation tests.

07. Hardware / Software Requirements

- Python 3.10+, PyTorch, Ultralytics-style YOLO tooling or equivalent
- DeepPCB dataset (1,500 annotated pairs)
- GPU strongly recommended for detector training; 8 GB RAM
- Flask demo backend; HTML/CSS/JS frontend

08. Expected Outcomes

A trained PCB defect detector, per-class mAP@0.5 measured on the 500-pair test split, an alignment + inference pipeline, and an inspection web demo with PASS/FAIL verdicts. The report documents which defect classes are hardest (typically pinhole/spur) with the measured numbers.

09. Applications

- Low-cost AOI for small PCB fabrication units
- Incoming quality checks for electronics assemblers
- Teaching aid for industrial-AI and quality-engineering courses

10. Future Scope

- Extend to assembled-PCBA defects (missing/tombstoned components)
- Add solder-joint quality grading
- Edge deployment on a Jetson-class device at the inspection station