

EPID: The Enfield PCB Inspection Dataset for Visual Defect Detection

Nikola Pižurica^{1,2*†}, Nikola Milović^{2*†}, Igor Jovančević^{1,2†},
Amirshayan Nasirimajd^{3†}, Walter Quadrini^{3†}

^{1*}Computer Science Center, Faculty of Natural Sciences and Mathematics, University of Montenegro, Cetinjska 2, Podgorica, 81000, Montenegro.

²Fain Tech, Serdara Jola Piletića 8, Podgorica, 81000, Montenegro.

³Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Piazza Leonardo Da Vinci 32, Milano, 20133, Italy.

*Corresponding author(s). E-mail(s): nikola.p@ucg.ac.me;
nikola.milovic@faintech.me;

Contributing authors: igorj@ucg.ac.me;
amirshayan.nasirimajd@polimi.it; walter.quadrini@polimi.it;

[†]These authors contributed equally to this work.

Abstract

We present EPID, the Enfield PCB Inspection Dataset, a high-resolution image collection designed for benchmarking visual defect detection systems in printed circuit boards (PCBs). The dataset consists of 446 annotated images split into two subsets: a primary training set of 345 images and a separate validation set of 101 images. All images depict progressive physical damage to components such as integrated circuits (ICs) and capacitors, supporting temporal modeling and low-data learning scenarios. Each component is manually labeled as defective, non-defective, or ignored. EPID enables research in object detection, neural architecture search, and robust model generalization for industrial inspection tasks.

Keywords: Printed Circuit Boards, Visual inspection, Defect detection, Dataset, Object detection, ICs, Capacitors, Temporal modeling, Low-data learning, EPID

1 Introduction

Printed circuit boards (PCBs) are critical components in virtually all electronic systems, and ensuring their quality is essential for industrial reliability, safety, and performance. Automated visual inspection of PCBs has become increasingly important for identifying defects early in the production cycle, but modern deep learning approaches depend heavily on access to large, well-annotated datasets.

While several PCB-related datasets exist, most are limited in scope, fail to reflect realistic industrial conditions, or lack granular annotations at the component level. In particular, publicly available datasets rarely capture the progressive nature of physical damage, nor do they support studying defect emergence over time, both of which are essential for developing robust and generalizable machine learning models in manufacturing contexts.

To address these gaps, we introduce the Enfield PCB Inspection Dataset (EPID), a high-resolution, manually annotated image collection designed for benchmarking visual defect detection systems under realistic and constrained data conditions. EPID includes two subsets: a training set with 345 images of 107 unique boards showing incremental damage across multiple stages per PCB, and a validation set with 101 additional images spanning 33 unique boards.

The dataset supports key tasks such as object detection, temporal modeling, and low-data learning. It also enables evaluation of methods such as neural architecture search (NAS), anomaly localization, and generalization across PCB geometries and damage types. All components are labeled as defective, non-defective, or ignored, with clear metadata and reproducible acquisition protocols.

EPID fills an important gap in the landscape of visual inspection benchmarks by offering realistic, component-level annotations and damage progression in a compact, accessible format for academic and industrial research.

2 Dataset Description

The Enfield PCB Inspection Dataset (EPID) consists of 446 high-resolution images capturing a range of physical defects in printed circuit board (PCB) components. The dataset is divided into two subsets: a training set with 345 images and a validation set with 101 images. Both subsets feature annotations for integrated circuits (ICs) and capacitors, enabling fine-grained analysis at the component level. Although both subsets were created using the same acquisition protocol, they were produced independently by two different teams. This deliberate separation introduces natural variability in execution while preserving methodological consistency. As a result, the validation set provides a more robust test of model generalization across operators and subtle shifts in data characteristics, even under standardized conditions. All images are captured from a fixed top-down perspective using a smartphone camera (3840×2160 resolution) under naturally varying lighting conditions. The types of artificially introduced defects fall into two main categories: **(1) IC-related defects**, including burn marks, melted or detached pins, and darkened or disfigured package surfaces; and **(2) capacitor-related defects**, most notably physical rupturing or popping of the capacitor canister. These defects were designed to simulate common fault conditions

observed in industrial PCB failure cases, while also enabling clear visual annotation for supervised learning tasks. Example images of these defect types are shown in Figure 1.



Fig. 1: Examples of defect types included in the dataset. Left: IC with burn marks; Middle: IC with dismantled pins; Right: Popped capacitor.

Each PCB in the training set was subjected to multiple damaging operations, with images captured after each step. This progressive approach enables modeling of defect emergence over time and supports sequence-based or low-data learning strategies. Defects include burn marks, melted or detached pins, popped capacitors, and combinations thereof.

The validation set includes 101 images representing 33 distinct PCBs, with an average of approximately 3 images per board. As with the training set, annotations follow a consistent policy: components of interest are labeled as **defective** or **non-defective**. Components that are extremely small, unidentifiable, or irrelevant to the target tasks (e.g., passive SMT elements or LED-like parts) are excluded from annotation.

The complete dataset provides:

- **446 high-resolution JPG images** (345 training, 101 validation),
- **Component-level annotations** with bounding boxes and defect labels,
- **Step-wise metadata logs and naming conventions** that enable reconstruction of progressive damage sequences,
- **Clear image naming conventions** aligned with acquisition steps.

Together, these elements make EPID suitable for benchmarking visual inspection models under realistic and constrained data scenarios.

Descriptive Statistics and Insights

We summarize per-PCB (RunID) statistics computed from the step-wise damage log in Table 1. On average, each PCB is captured over multiple steps, with a varying portion of its components recorded as damaged. This variety provides a rich basis for analyzing component damage in different configurations.

Metric	Value
Avg. ratio of damaged ICs (per PCB)	0.238
Avg. ratio of damaged capacitors (per PCB)	0.123
Avg. number of ICs per PCB	1.907
Avg. number of capacitors per PCB	5.000
Avg. number of steps per PCB	3.224
Min/Max steps per PCB	1 / 6
Min/Max ICs per PCB	0 / 8
Min/Max capacitors per PCB	0 / 31

Table 1: Aggregate per-PCB statistics from `damage_log.csv`.

Figure 2 shows the distribution of IC counts per PCB. Boards range from those with no ICs to those with up to eight, with a substantial representation of boards containing one, two or three ICs. Figure 3 presents the capacitor counts, spanning from none to 31, reflecting a wide diversity in PCB layouts.

The dataset contains a mixture of boards where certain component types may be absent, as well as boards with high component counts. This diversity supports the evaluation of algorithms under different levels of visual complexity and instance density. The observed proportions of damaged components - approximately 24% for ICs and 12% for capacitors - ensure that both intact and damaged instances are well represented, which can be advantageous for training and testing balanced detection models.

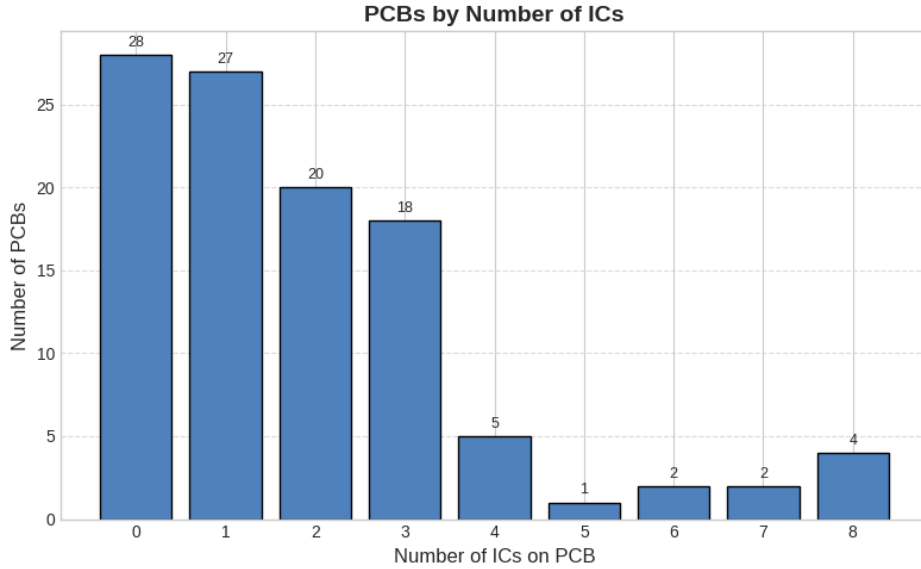


Fig. 2: PCBs by number of ICs (histogram across unique boards).

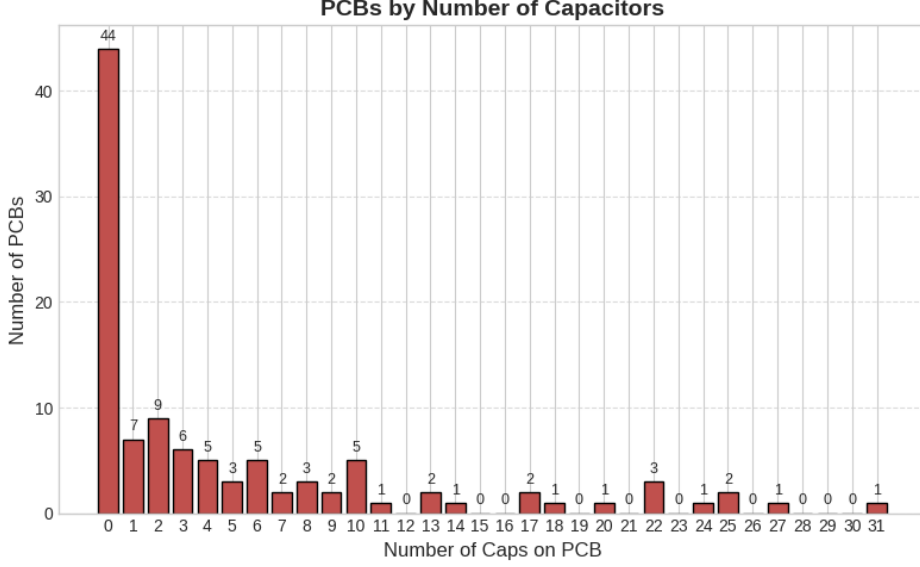


Fig. 3: PCBs by number of capacitors (histogram across unique boards).

Exploration insights:

1. **Diverse component counts.** The wide range of IC and capacitor counts enables benchmarking models across both sparse and dense component arrangements.
2. **Balanced representation of states.** The presence of both damaged and intact components in meaningful proportions supports training and evaluation in realistic defect detection scenarios.
3. **Step-wise sequences.** An average of 3.2 images per PCB allows exploration of algorithms that leverage limited temporal progression, such as change detection or step-aware feature learning.
4. **Variation in board composition.** The dataset includes PCBs that range from having no instances of a particular component type to others with a high number of components. This variation supports evaluation of inspection methods across both minimal and densely populated layouts.

3 Data Acquisition and Annotation

The EPID dataset was acquired using a controlled, stepwise process in which physical defects were intentionally applied to printed circuit boards (PCBs). All PCBs were imaged from a fixed top-down perspective using a high-resolution smartphone camera (3840×2160 pixels), with minimal environmental interference and only naturally varying lighting conditions. Each board was photographed multiple times throughout

the damage procedure, resulting in a sequence of images capturing incremental defect progression.

Defect Creation Procedure

The damaging process was carried out by a two-person team: an **actuator**, who physically applied the defects, and a **supervisor**, who oversaw the process and operated the annotation planning script. For each PCB, the supervisor executed a custom Python script, `generate_damage_policy.py`¹, which generated a randomized sequence of step-by-step instructions specifying which components to damage and how. This script accounted for the number of ICs and capacitors present on the board and returned a damage plan indicating component indexes and damage types for each step.

The actuator followed these instructions to apply physical damage using a set of simple, reproducible tools:

- Simulating burn marks on IC packages using a permanent marker; in cases where the mark was not sufficiently visible, a light application of oil was used to enhance contrast,
- Melting or detaching IC pins using a soldering iron,
- Damaging capacitors mechanically using a screwdriver or utility knife.

After each damage step, a high-resolution image of the PCB was captured using a fixed camera setup. Every action and resulting image was logged to `damage_log.csv`, which includes the run ID, timestamp, number of components, step number, and the list of components damaged at that stage. Image filenames follow a consistent format (`<run_id>_<step_number>.jpg`) to maintain alignment with metadata.

To ensure reproducibility and traceability, each action was logged into `damage_log.csv`, which stores metadata such as run ID, timestamp, step number, component counts, and the specific damage applied at each stage. Image filenames follow the format `<run_id>_<step_number>.jpg`, directly linking each image to its corresponding metadata.

Component Indexing and Naming Conventions

Before applying any damage, each component on the PCB was indexed based on its spatial location in the initial intact image. ICs and capacitors were numbered in a left-to-right, top-to-bottom order (e.g., IC#1, Cap#1), as illustrated in Figure 4. These indexes were used consistently in both damage instructions and annotations, providing a uniform reference throughout the dataset.

¹https://github.com/Fain-Tech/lowdata-pcb-nas/blob/main/dataset/generate_damage_policy.py

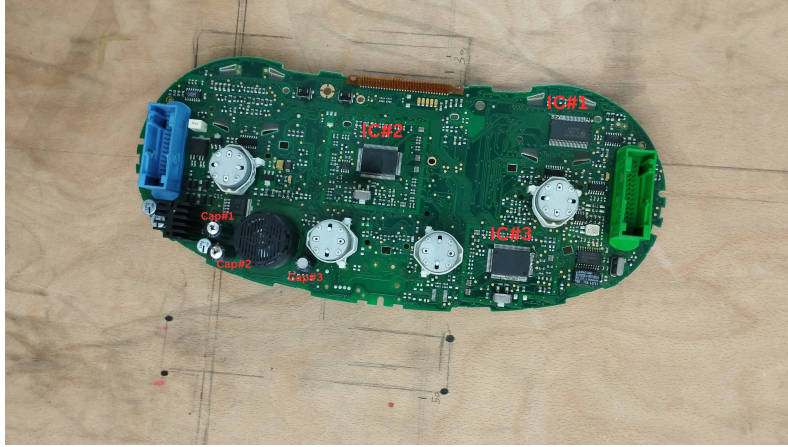


Fig. 4: Example of component indexing used during annotation and data acquisition. ICs and capacitors are labeled left-to-right, top-to-bottom.

Annotation Protocol

All images were manually annotated using the Computer Vision Annotation Tool (CVAT). Annotations consist of bounding boxes around visible ICs and capacitors. Each component is labeled according to one of three categories:

- **Defective** - visibly damaged and/or listed in the metadata log,
- **Non-defective** - visually intact and not mentioned in damage logs,
- **Ignored** - components deemed irrelevant or unidentifiable (e.g., small SMTs, passive elements, or LED-like packages).

Only components relevant to the learning objectives were annotated. Small or ambiguous elements were excluded to reduce noise and preserve annotation clarity. The annotation files are provided in a consistent format aligned with each image and its corresponding damage metadata. An example annotation interface is shown in Figure 5.



Fig. 5: Example of image annotations in CVAT. Bounding boxes are drawn around ICs and capacitors and labeled as either defective or non-defective.

Bounding box size distribution.

We further analyzed the relative scale of annotated objects by computing each bounding box area as a percentage of the corresponding image area (3840×2160 pixels). No annotated instance in the dataset exceeds 1% of the image area. Figure 6 shows the complete distribution, where the vast majority of boxes occupy less than 0.5% of the image. The largest group falls into the 0.0–0.1% range (874 boxes), followed by 0.1–0.2% (550 boxes), with progressively fewer instances in higher ranges. This highlights that the dataset predominantly contains very small objects, making it a suitable benchmark for evaluating fine-grained detection and small-object recognition techniques.

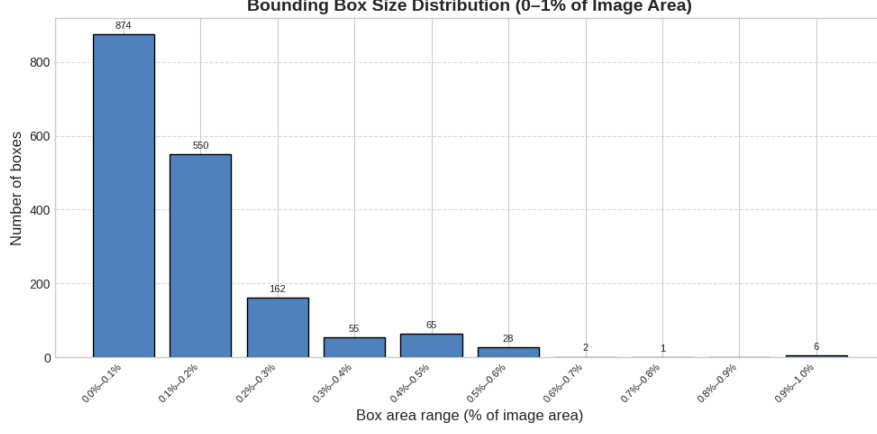


Fig. 6: Bounding box size distribution (relative to image area) for objects occupying 0%–1% of the image. Each bar represents the number of annotated instances within a given area range.

4 Use Cases and Applications

The EPID dataset is designed to support a range of tasks in both academic and industrial research related to visual inspection and quality control of electronic components. Its high-resolution images, step-wise progression of damage, and component-level annotations make it especially suitable for evaluating models under constrained data regimes and realistic defect scenarios.

- **Object Detection and Localization:** EPID enables training and evaluation of models that detect and localize defective components, such as ICs and capacitors, within complex PCB layouts.
- **Low-Data Learning:** Due to its modest size and incremental labeling, EPID is well-suited for benchmarking few-shot, semi-supervised, and transfer learning approaches in visual inspection contexts.
- **Temporal Reasoning:** The structured progression of images across damage steps allows for the study of temporal consistency and change detection models, despite the data being static images rather than video.
- **Neural Architecture Search (NAS):** The dataset has been used in practice to evaluate the performance of automatically designed neural networks under low-data constraints, offering a reproducible benchmark for architecture optimization.
- **Generalization and Robustness:** The separation between training and validation data by team, despite shared methodology, enables controlled testing of model generalization to inter-operator variability and minor acquisition shifts.

In addition to academic exploration, EPID can serve as a lightweight benchmark for industrial inspection tools targeting embedded systems, edge AI deployment, or training pipelines for real-time visual inspection in manufacturing environments.

5 Availability

The EPID dataset is publicly available on Zenodo² under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. This ensures unrestricted access for academic and industrial use, with proper attribution.

The release package includes:

- All 446 high-resolution JPG images (345 training, 101 validation),
- Component-level annotations in COCO-style JSON format,
- Step-wise metadata logs (`damage_log.csv`),
- A README file with usage instructions and license terms.

Citation information and a permanent DOI are provided on the Zenodo landing page to support reproducibility and proper dataset attribution in publications.

Acknowledgments

This work has been supported by the project *European Lighthouse to Manifest Trust-worthy and Green AI* (ENFIELD), which has received funding from the European Union’s Horizon Europe research and innovation program under grant agreement No. 101120657.

²<https://doi.org/10.5281/zenodo.16811808>