Statistical Comparison of YOLO Algorithm Versions and Training Optimizers in PCB Soldering Splashes Detection

Peter Klco
Dept. of Mechatronics and Electronics
University of Žilina, FEEIT, Žilina, Slovakia
peter.klco@uniza.sk

Milan Smetana
Dept. of Theoretical Electrical Engineering and Biomedical Engineering
University of Žilina, FEEIT, Žilina, Slovakia
milan.smetana@uniza.sk

Dusan Koniar
Dept. of Mechatronics and Electronics
University of Žilina, FEEIT, Žilina, Slovakia
dusan.koniar@uniza.sk

Libor Hargas
Dept. of Mechatronics and Electronics
University of Žilina, FEEIT, Žilina, Slovakia
libor.hargas@uniza.sk

***Abstract*—Quality inspection of electronic boards during manufacturing is a critical step, particularly for expensive power electronics modules, underscoring the importance of reliable detection methods for PCB soldering splashes. The existing works on PCB defect detection have focused more on neural network architectural modifications, such as attention mechanisms, feature pyramid improvements, and lightweight backbone replacements. However, the choice of neural network optimizer is often treated as a secondary concern, potentially overlooking opportunities for performance improvement. Recognizing that the stochastic gradient descent (SGD) or adaptive moment estimation (Adam) algorithm can significantly influence outcomes can inspire professionals to consider optimizer selection more carefully. To the best of the authors' knowledge, a rigorous statistical comparison of these three optimizers, specifically in the context of YOLO-based soldering splash detection on PCBs, has not yet been reported in the literature. This paper aims to evaluate three YOLO versions — YOLOv8, YOLO11, and the recently released YOLO26 — combined with three optimizers (Adam, AdamW, and SGD) for automated soldering splash detection on specific power electronics PCBs. To ensure statistical robustness, 10 independent training runs with different random seeds were performed for each of the 9 model–optimizer combinations. Results reveal significant differences across YOLO versions and optimizer choices, challenging the assumption that the latest YOLO architecture always performs best. Including YOLO26 offers valuable benchmarking for similar applications. All results are supported by two-way ANOVA analysis, with normality testing of distributions.**

Keywords—deep neural network; optimizer; YOLO algorithm; automated detection; soldering splashes; quality inspection

I. INTRODUCTION

Soldering splashes are a recurring defect in the manufacturing of power-electronics printed circuit board (PCB) modules: excess solder ejected during reflow can bridge conductive paths, and any board on which a splash is found in a restricted area must be reworked before the module can be shipped. Automated optical inspection built on convolutional neural networks, and in particular the YOLO (You Only Look Once) family of one-stage detectors, has become the default approach for this task because it combines competitive accuracy with real-time inference speed.

Two practical hyperparameters appear whenever a YOLO detection model is deployed for a new inspection task: which architecture version to use, and which optimizer to train it with. Both are usually chosen independently from a single training run rather than repeated across seeds, since training on large datasets can be time-consuming and computationally demanding. Researchers often use default values of YOLO model training hyperparameters.

Recent solder-joint detectors built on X-ray imaging [1], attention-guided feature fusion [2], or a modified YOLO11 backbone [3] each report single architecture and single optimizer configuration, and a production-line detector [4] likewise reports single-configuration results. Dedicated comparative reviews of YOLO versions [5], [6], widely used Ultralytics release versions - YOLOv8, YOLO11 and, most recently, YOLO26 - [7] inspect architectural differences without testing whether they interact with training hyperparameters. Comparative work across YOLOv8 and YOLO11 on other tasks [8] further suggests architecture-dependent behavior is common. The optimizer-focused studies such as a momentum-tuning analysis of Adam on YOLOv8 [9] vary the optimizer on a single fixed architecture. This can lead to the question of whether the best optimizer for a solder-splash detector likewise depends on YOLO model architecture.

This paper addresses that question with a full factorial, statistically validated design rather than a single point estimate. Three YOLO architectures are crossed with three optimizers, each of the nine cells repeated across ten pseudo-random seeds ($n = 90$ runs). Beyond the factorial design itself, the analysis is deliberately over-determined: every ANOVA effect is cross-checked with a non-parametric or variance-robust alternative wherever a normality or homogeneity assumption is violated. This level of statistical analysis remains uncommon in PCB-defect-detection papers, most of which report accuracy figures from a single run without significance testing. The contributions of this work are therefore (i) the first reported model×optimizer interaction analysis for YOLO-based solder-splash detection, (ii) a fully reproducible 90-run factorial design cross-validated against four independent statistical tests, and (iii) practical guidance - AdamW as a safe default, and SGD specifically to avoid with newer/larger architectures - for practitioners deploying automated PCB inspection.

II. MATERIALS AND METHODS

A. Dataset and Image Acquisition

The experiments use the annotated solder-splash dataset containing 272 images and single-class ("solder blob") annotation protocol described in a previous study [10]. In short, PCB images of SEMITRANS-type power modules were acquired in a controlled testing chamber with a 15 MPix color camera under homogeneous lighting, tiled into 640×640 px sub-images, so that no annotated object is cut by a tile boundary, and split into training and validation subsets.