

# EDGE-BASED DEEP LEARNING SYSTEM FOR AUTOMATED PCB COMPONENT INSPECTION

B. Jaswanth<sup>1</sup>, D. Chandiran<sup>1\*</sup>, J. Mohammed Sami<sup>1</sup>, Dr. K. Gobivel<sup>2\*</sup> and Dr. S. Gayathri Monicka<sup>2</sup>

<sup>1</sup>Department of Mechatronics Engineering, KCG College of Technology, Karapakkam, Chennai 600 036, India

## Abstract

Automatic Optical Inspection (AOI) is essential in Switched Mode Power Supply (SMPS) PCB manufacturing, where missing components can lead to functional failure and safety risks. This paper presents a low-cost embedded deep learning-based AOI system for real-time component verification using a Raspberry Pi 5 as the edge computing platform and a YOLOv11 object detection model to identify critical components such as the fuse, capacitor, MOV, and optocoupler. A dataset of approximately 500 PCB images was collected under varying lighting and orientation conditions and annotated for supervised training. The trained model was optimized and deployed in NCNN format to enable efficient inference on resource-constrained hardware. The complete system integrates conveyor-based PCB handling, Region of Interest (ROI) extraction, and a Finite State Machine (FSM) to ensure stable and synchronized operation. Operating at a 640 × 640 resolution, the system achieves 5–10 FPS with an average inspection time of approximately 3 seconds per PCB. Based on detection results, automatic PASS/FAIL classification and actuator-based sorting are performed.

**Keywords:** Automatic Optical Inspection (AOI), SMPS PCB Inspection, Embedded Vision System, Deep Learning, YOLOv11, Raspberry Pi 5, NCNN Deployment, Real-Time Object Detection, Finite State Machine (FSM), Conveyor-Based Automation,

## 1. Introduction

Modern electronics manufacturing environments face significant challenges in maintaining consistent PCB quality during high-volume production. Manual component inspection requires dedicated personnel and direct visual examination of each board. This approach creates bottlenecks during continuous production and increases the risk of undetected defects reaching end customers. Manufacturing facilities require autonomous, non-contact inspection solutions that operate continuously without human intervention.

This paper presents an edge-based automated inspection system for SMPS PCB component verification using embedded deep learning. The system performs real-time detection of missing components and automated rejection of defective boards using a Raspberry Pi 5 as the edge computing platform. The system integrates mechanical conveyor design, embedded inference, vision-based detection, and actuator control into a cohesive solution.

- Lightweight YOLOv11n model deployed in NCNN for real-time edge inference
- ROI extraction pipeline isolating the PCB area prior to component detection
- FSM-based conveyor control ensuring synchronized and stable inspection operation

---

\*Corresponding author: [chandiran1611@gmail.com](mailto:chandiran1611@gmail.com), [gobivel@gmail.com](mailto:gobivel@gmail.com)

- Actuator-based rejection mechanism enabling automatic PASS/FAIL sorting without manual intervention

## 1.1 System Architecture Overview

The proposed system consists of three primary subsystems: the mechanical handling module, the vision and inference module, and the control and output interface. The mechanical handling module employs a conveyor belt driven by a DC motor as part of a custom-designed conveyor mechanism. The vision and inference module incorporates a Pi Camera Module 3 paired with a YOLOv11n object detection model deployed via the NCNN framework on the Raspberry Pi 5. The control and output interface comprises PASS/FAIL classification logic and an actuator-based rejection mechanism that operate in a synchronized closed-loop manner.

## 2. Hardware Design and Specifications

### 2.1 Mechanical Structure and Conveyor Design

The inspection system is built around a custom conveyor assembly that carries PCBs one at a time through a fixed imaging zone. The structure holds the belt, camera mount, and sorting mechanism together in a compact setup. The camera is fixed directly above the conveyor to maintain a consistent top-view angle and working distance for every board, ensuring stable and uniform image capture throughout operation.

When a PCB reaches the inspection position, the conveyor stops briefly to allow a clean image to be captured before moving again. Based on the inspection result, a rotating bin mechanism at the end of the conveyor line responds accordingly — rotating to the PASS side to collect acceptable boards and rotating to the FAIL side to collect defective boards. This approach ensures that both passing and failing boards are automatically sorted into their respective collection bins without any manual handling, keeping the entire inspection and sorting cycle smooth and fully automated.

### 2.2 Multi-Motor Configuration and Control System

A single DC motor drives the conveyor belt responsible for transporting PCBs through the inspection zone. The motor operates at 12 V, 2 A and is controlled through a relay-based switching circuit that allows the Raspberry Pi 5 to start and stop the conveyor at the appropriate inspection positions. This configuration enables precise conveyor control, including forward movement, controlled stops at the inspection zone, and resumption after the PASS/FAIL decision is made.

Control signals originate from the Raspberry Pi 5, which executes the Finite State Machine (FSM) logic governing the entire inspection cycle. The FSM coordinates conveyor movement, image capture, and actuator response based on PCB position and inference results. A servo motor drives the rotating bin mechanism, receiving a positional command from the Raspberry Pi 5 to rotate toward the PASS or FAIL side depending on the detection outcome.

### 2.3 Vision and Sensing Module

The vision module incorporates a Raspberry Pi Camera Module 3 mounted vertically above the inspection zone, with a ring light mounted concentrically around the camera stand to provide uniform, shadow-free illumination across the PCB surface at approximately 800 lux. The camera is positioned at a fixed working distance with stable mounting to minimize image distortion and motion blur during acquisition.

Image capture is triggered when the conveyor halts at the predefined Region of Interest (ROI) position. The ROI is defined as a fixed rectangular crop centred on the PCB footprint, determined during initial system calibration. At runtime, a full-resolution frame is captured and a fixed pixel-coordinate crop is applied, then resized to 640×640 pixels before being passed to the NCNN inference engine. This eliminates background clutter, reduces the detector search space, and improves per-class detection confidence. The ROI boundaries are stored in a configuration file and can be recalibrated without retraining the model. The YOLOv11n model then inspects for the presence of four critical components: fuse, high-voltage capacitor,

MOV, and optocoupler, and classifies each board as PASS or FAIL accordingly. Camera-based detection provides rapid, non-contact assessment of component presence across the entire PCB surface, making it well suited for low-cost embedded deployment without specialized sensing equipment.

### **3. Software Architecture**

#### **3.1 Embedded Control Firmware Architecture**

The Raspberry Pi 5 runs the inspection control software developed in Python using the OpenCV and NCNN libraries. The software implements a Finite State Machine (FSM) executing three sequential tasks: (1) conveyor control task, (2) image acquisition and inference task, and (3) sorting and output task.

The conveyor control task manages belt movement and stop signals based on PCB position at the ROI. The image acquisition and inference task captures the top-view frame, extracts the ROI, and passes it through the YOLOv11n model for component detection. The sorting and output task receives the PASS/FAIL decision, triggers the servo motor, and displays the result before signalling the conveyor to resume. The YOLOv11n model was trained using the Ultralytics framework on Google Colab with a Tesla T4 GPU for 100 epochs, batch size of 16, and initial learning rate of 0.01 using SGD optimizer with cosine annealing decay..

#### **3.2 Operator Output and Display System**

The inspection system provides real-time visual feedback to the operator through a connected display. The output interface shows the live camera feed with bounding box overlays indicating detected components, along with the PASS or FAIL classification result for each inspected board. Detection confidence scores are shown alongside each bounding box to indicate model certainty for each component class.

Historical inspection counts including total boards inspected, total passed, and total failed are maintained on screen throughout the production session, giving the operator a running summary without interrupting the inspection flow.

#### **3.3 System Communication and Control Protocol**

The proposed system employs GPIO-based communication between the Raspberry Pi 5 and the servo motor controlling the rotating bin mechanism. The Pi Camera Module 3 interfaces via the CSI-2 interface for high-speed image data transfer. Conveyor motor control is handled through relay switching signals from the Raspberry Pi 5 GPIO pins.

The FSM defines the following operational states: IDLE (waiting for PCB), STOP (conveyor halted at ROI), INSPECT (image capture and inference), SORT (servo rotation to PASS or FAIL side), and RESUME (conveyor restart for next board). State transitions are event-driven based on inference completion and actuator feedback. Average end-to-end cycle time from conveyor stop to resume measures approximately 3 seconds per board.

## **4. Experimental Evaluation**

### **4.1 Testing Environment and Conditions**

Testing occurred in a controlled indoor laboratory environment representative of small-scale electronics manufacturing settings. Environmental parameters: Temperature range 24–28 °C, Relative humidity 50–65%, Ambient light levels 400–800 lux. Ring light illumination was set to approximately 800 lux for primary testing and reduced to 400 lux in a secondary

test to evaluate robustness under lower ambient conditions. The conveyor was operated on a flat stable surface to assess baseline inspection performance without external disturbances

## 4.2 Detection and Sorting Performance Results

During live testing, multiple SMPS boards were inspected under ring light illumination mounted directly above the conveyor, providing uniform and shadow-free lighting across the PCB surface. The system was tested with both complete and intentionally incomplete boards. The laptop display showed real-time bounding box overlays for each detected component as the PCB stopped under the camera. Following inference, the servo motor rotated the bin correctly to the PASS side for complete boards and to the FAIL side for boards with missing components, with no manual intervention required throughout the entire cycle.

The YOLOv11n model achieved an overall classification accuracy of 94%, precision of 96.2%, recall of 91.7%, and Map at 0.5 of 93.8% across all four component classes. Per-class AP values were: fuse 95.1%, high-voltage capacitor 94.3%, MOV 92.6%, and optocoupler 93.2%. The false positive rate was 3.8% and the false negative rate was 8.3%. False negatives are critical in safety applications as an undetected missing fuse or MOV can lead to field failures. Analysis revealed that false negatives occurred primarily when a component was partially occluded by an adjacent taller component or when the PCB was slightly misaligned beyond the nominal ROI boundary, accounting for approximately 5% of test samples.

Inference performance on the Raspberry Pi 5 achieved approximately 9.3 FPS with four CPU threads enabled and 3.2 FPS at single-thread operation. At four threads, average inference time per frame was 108 ms; at single thread it was 312 ms. The average end-to-end inspection time per board, including conveyor stop, image capture, inference, PASS/FAIL decision, and servo actuation, was approximately 3.1 seconds. No detection accuracy degradation was observed when ambient lighting was reduced from 800 lux to 400 lux, confirming robustness of the ring light setup.

## 4.3 System Reliability and Stability

Extended operation testing evaluated system performance over a continuous 4-hour runtime under repeated inspection cycles. The system maintained stable FSM operation without software crashes or reset events. Conveyor stop positioning remained consistent across all cycles. The YOLOv11n model produced repeatable inference results with no observed performance degradation over the test duration. The Raspberry Pi 5 CPU temperature remained within safe operating bounds throughout, confirming thermal stability for sustained use

# 5. Discussion

## 5.1 System Performance Analysis

Experimental results demonstrate that the proposed system meets design specifications for real-time PCB component inspection at the edge. The YOLOv11n model delivers reliable detection performance across all four component classes. Classification accuracy of 94% and precision of 96.2% confirm that the system produces dependable PASS/FAIL decisions suitable for production-line use. The end-to-end inspection time of 3.1 seconds per board aligns with the operational tempo of small- and medium-scale manufacturing environments.

The FSM-based control architecture successfully coordinates conveyor motion, image capture, model inference, and servo actuation into a smooth and uninterrupted inspection cycle. The rotating bin mechanism correctly sorted all test boards, confirming that the mechanical and software subsystems operate in reliable coordination. GPIO-based communication between the Raspberry Pi 5 and actuators demonstrated consistent response with no missed trigger events during testing.

## 5.2 Advantages over Existing Systems

Compared to manual inspection, the proposed system eliminates operator fatigue and subjective judgment entirely. The ring light setup ensures consistent illumination independent of ambient conditions, directly addressing a common failure point in vision-based inspection systems. Unlike GPU-based deep learning systems requiring expensive dedicated hardware, this system runs fully on a Raspberry Pi 5 making it accessible to small and medium manufacturers. The integrated servo-actuated rotating bin physically separates PASS and FAIL boards automatically, going beyond most research prototypes that only detect defects without completing the physical sorting step.

The novelty of the proposed system lies not merely in applying YOLO to an embedded platform, but in the complete integration of ROI-based preprocessing, NCNN quantization, FSM-controlled conveyor synchronization, and servo-actuated physical sorting into a single low-cost pipeline. Prior work such as Brahmhatt and Rane demonstrated YOLO deployment on Jetson Nano but did not include automated physical rejection or FSM-based conveyor control.

## 5.3 Current System Limitations

The current implementation exhibits notable limitations requiring future development. First, the dataset of 500 images, while sufficient for this proof of concept, limits generalization to other board layouts and component densities. Expanding the dataset to include multiple PCB variants and real production variations would improve model robustness. Second, the inspection scope is limited to missing component detection on a single SMPS board type. Other critical PCB defects such as incorrect component polarity, wrong component values, solder bridges, lifted pins, and damaged conductive traces are not currently detected. Third, validation was conducted entirely in a laboratory setting and the system has not yet been evaluated under real industrial operating conditions involving conveyor speed variation, electromagnetic interference, and continuous multi-shift operation. Deployment at a live SMPS manufacturing facility is planned as the immediate next phase of validation.

## 6. Conclusions and Future Work

This paper presented a complete edge-based automated inspection system for detecting missing components on SMPS PCBs using embedded deep learning. The system successfully integrates a custom conveyor mechanism, ring light illumination, Pi Camera Module 3, YOLOv11n model in NCNN I, FSM-based control logic, and a servo-actuated rotating bin into a unified inspection framework. Experimental validation demonstrates reliable component detection, consistent sorting performance, and stable operation over extended runtime. The proposed system validates the feasibility of deploying intelligent embedded inspection solutions in small- and medium-scale electronics manufacturing environments.

Planned future enhancements include:

1. Extending the training dataset to cover other electronic boards such as motor driver boards, inverter PCBs, and microcontroller-based control boards, enabling the system to generalise across multiple product types in a single production line.
2. Expanding defect detection to include mechanical components such as missing fasteners, misaligned connectors, and improperly seated heat sinks in electromechanical assemblies.
3. Building a consolidated multi-defect detection model that simultaneously identifies missing components, polarity errors, solder defects, and damaged traces in a single inference pass.
4. Implementing adaptive illumination control that automatically adjusts ring light intensity based on PCB surface reflectivity to improve detection under varying board finishes.
5. Developing a lightweight web-based dashboard for remote monitoring of inspection statistics, PASS/FAIL trends, and production throughput accessible from any device on the local network.

## 7. Circuit Diagrams and System Architecture

The following figures illustrate the hardware components and system workflow implemented in the proposed PCB inspection system. The modular design enables independent subsystem testing and facilitates future upgrades. Each figure represents a key element: the camera and imaging setup, the hardware prototype, the Raspberry Pi 5 processing unit, and the end-to-end inspection workflow.

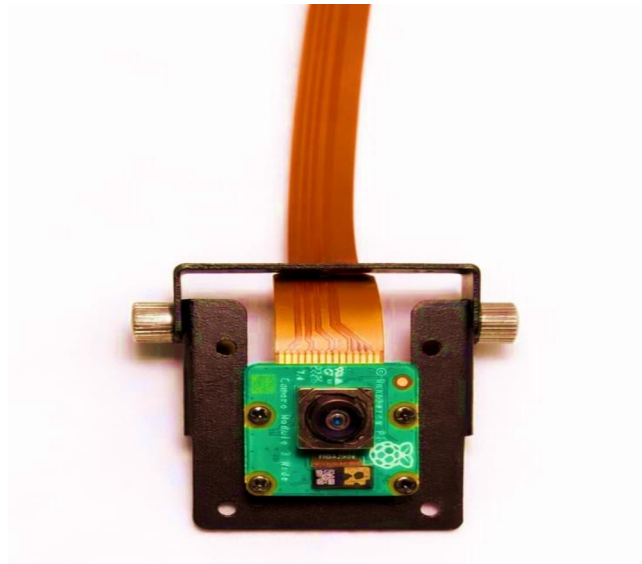


Fig.1. Pi Camera module 3 Used for inspection.



Fig.2. Raspberry Pi 5 for data processing.

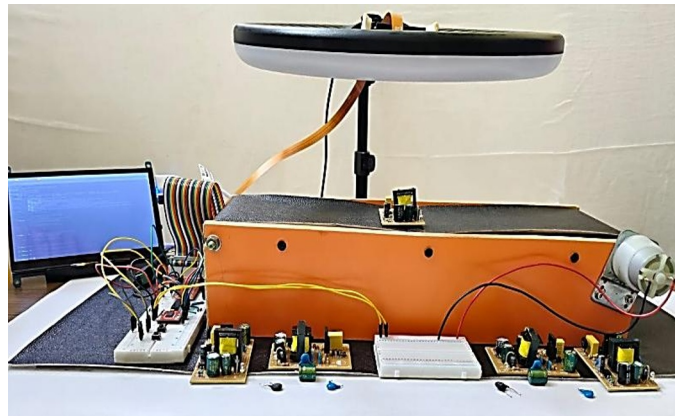


Fig.2. Hardware Prototype of PCB Inspection.

The figures above present the complete hardware implementation of the inspection system. The camera module captures high-resolution top-view images under stable lighting conditions. The Raspberry Pi 5 processes each frame locally using the NCNN-optimized YOLOv11n model, producing bounding box detections and a PASS/FAIL classification within approximately 3.1 seconds. The rotating bin mechanism physically sorts each board to the correct side based on the inference outcome. All subsystems operate under FSM supervision with event-driven state transitions, ensuring synchronized and repeatable operation across every inspection cycle.

## References

- [1] Y. Huang, C. Li, and Z. Zhang, "A real-time surface defect detection system for industrial products based on deep learning," *Sensors*, vol. **20**, no. 17, p. 5102 (2020)
- [2] V. Brahmabhatt and D. Rane, "Surface defect detection using YOLO and deep learning," *International Journal of Engineering Research & Technology*, vol. **10**, no. 06 (2021)
- [3] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD — A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF CVPR* (2019)
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in *Proc. IEEE CVPR* (2016)
- [5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004.10934 (2020)
- [6] G. Jocher et al., "YOLOv5," Ultralytics (2020). Available: <https://github.com/ultralytics/yolov5>
- [7] Ultralytics AI, "YOLOv8 documentation" (2023)
- [8] Raspberry Pi Foundation, "Raspberry Pi 5 documentation" (2023)
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR* (2016)
- [10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NeurIPS* (2015)