39 th International Conference On VLSI Design
3rd - 7th January 2026 | JW Marriott | Pune, India
25 th International Conference On Embedded Systems

AI-Based Defect Detection and Smart Rejection for Industrial Automation on PolarFire SoC

Abstract

Quality control in manufacturing and logistics often relies on manual inspection or power-hungry GPU solutions, leading to inefficiencies, errors, and high operational costs. This proposal introduces an *AI-based defect detection and smart rejection system implemented on the PolarFire SoC Icicle Kit*. The system leverages *FPGA-accelerated vision inference for real-time defect detection* and *RISC-V cores for actuator control, box ID tracking, and human–machine interface (HMI) management*. Items on a conveyor are captured using cameras, analyzed for damages or defects, and automatically rejected using servos if found faulty. QR scanning enables traceability of defective items, while the system also provides operator alerts and logs. The proposed solution demonstrates a scalable, low-power, and deterministic approach to inline quality inspection, applicable across packaging, logistics, pharmaceuticals, and industrial production lines.

Idea

The proposed system integrates computer vision, real-time defect detection, and automated rejection into a single compact solution on the PolarFire SoC Icicle Kit. The pipeline begins with multi-angle capture: two to three global-shutter cameras acquire each item as it passes on the conveyor so top and side faces are imaged. Cameras are synchronized with strobe lighting and the conveyor position sensor (encoder) so every frame is time-tagged with a frame ID and encoder count. That tagging is the foundation for deterministic mapping between a detection and the physical location on the belt.

The CNN inference is executed using Microchip's VectorBlox accelerator on the FPGA fabric, allowing the trained, quantized defect-detection model to run efficiently. Offloading convolution, pooling, and activation to the hardware accelerator provides low-latency, low-power inference while the RISC-V cores handle system coordination; this arrangement keeps the inspection pipeline in step with conveyor speeds without dropping frames. Practically, the model we target is a lightweight single-shot detector , SSDLite with a MobileNetV2 backbone , because it gives a good balance of bounding-box localization and runtime efficiency for edge accelerators. The network is trained by transfer learning from standard backbones on a curated dataset (public industrial defect sets plus in-line top/side captures), with heavy, production-style augmentations (brightness/contrast jitter, blur, geometric transforms). After floating-point training we quantize INT8 and export a TFLite INT8 model which is then compiled by the VectorBlox toolchain into the accelerator blob. At runtime the Fabric receives preprocessed, packed tensors so the accelerator delivers deterministic per-frame latency and only a compact result record is passed to the RISC-V subsystem. Model quality is validated both offline and operationally, and rejected snapshots are fed back into the training corpus for continuous improvement.

Once a defect is identified, the RISC-V cores perform system-level orchestration. Through AXI interfaces they receive inference results and perform tasks including:

- Issuing PWM/GPIO control signals to servos for rejecting defective items by pushing the flagged item from the conveyor belt;
- Interfacing with cameras over SPI to link product IDs (in QR code) with inspection outcomes;
- Updating inspection logs in onboard storage and transmitting logs to a higher-level system for analytics and traceability via bluetooth.
- Driving an HMI panel for operator visualization and a speaker/alert system for real-time notifications.

Traceability and model improvement are built into the flow: rejected frames and metadata are stored locally (eMMC/SD) and optionally uploaded to a server for analytics. Each logged event includes *timestamp/frame_ID, product ID, defect class/confidence,* and a *saved snapshot* to support root-cause analysis and dataset growth for future retraining. Finally, the architecture is designed to scale: adding more cameras or a depth sensor (laser line or structured light) is a matter of expanding the FPGA fabric to capture and preprocess pipelines and adjusting the accelerator input packing.

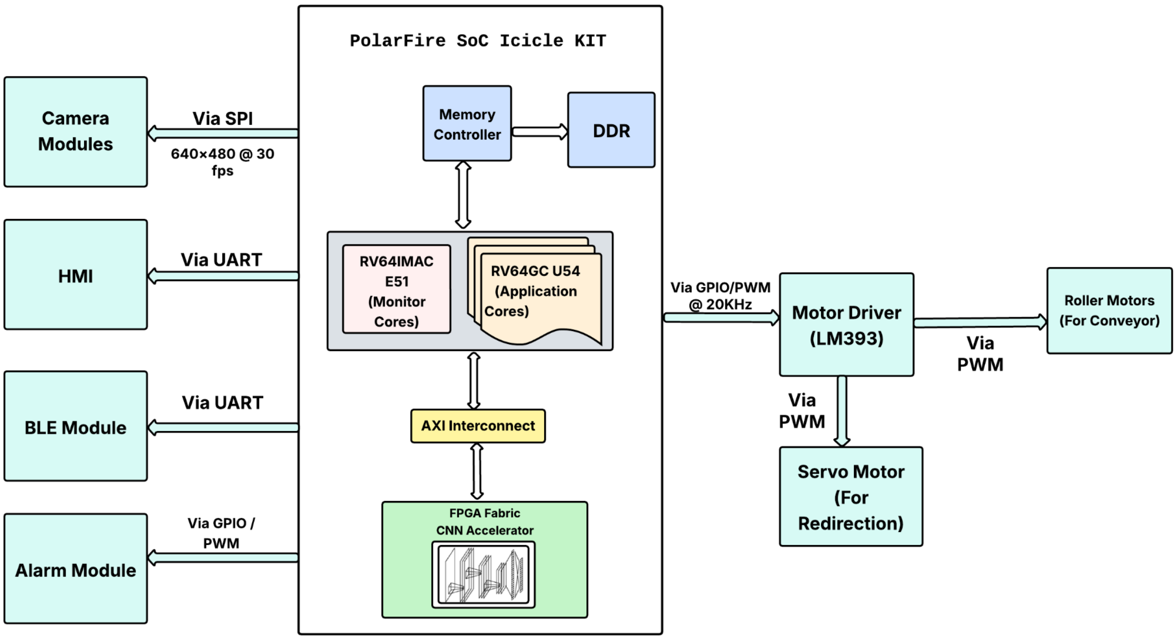

FIG 1 : Block Diagram of the Proposed Design

Application

The system is designed for real-time quality inspection and automated rejection across multiple industries:

- *Packaging & Logistics*: Detects and rejects damaged cartons or parcels before shipment.
- *Pharmaceuticals*: Ensure tamper-proof packaging integrity for medicines.
- *Food & Beverage*: Identify dented, deformed or unsealed cartons.

With future extensions such as *3D surface modelling,* the platform can scale to more complex inspection tasks in automotive and precision quality control aerospace production lines to control specification mismatch rate.

Value Add

Low Power, High Performance: FPGA-accelerated AI vision achieves real-time inspection at a fraction of the power of GPU-based solutions.

- Complete Inline Automation: Integrates vision, actuation, and traceability on a single SoC platform.
- Industrial Relevance: Directly addresses packaging, logistics, and manufacturing QC needs.
- Scalability: Extendable to 3D surface modelling and diverse industrial domains.
- Novelty: Goes beyond simple defect detection by combining real-time rejection, barcode logging, and operator feedback into a unified, edge-deployable solution.

References

1. VectorBlox Accelerator SDK - Microchip's product page for the VectorBlox SDK. Provides details on how to convert CNNs, supported platforms, and using the IP.
2. VectorBlox SDK GitHub Repository - source for the SDK itself, examples, tutorials, supported operators, quantization, simulator, etc'
3. Kaputt: A Large-Scale Dataset for Visual Defect Detection - a dataset specifically designed for benchmarking defect detection in manufacturing and logistics. Available at: https://www.kaputt-dataset.com/