

Autonomous Defect Detection System for High-Speed Assembly Lines

Anmol S. Budhewar¹, Ankita V. Karale², Meet K. Ghodasara³, Sujit B. Khalkar⁴, Om P. Dhoble⁵,
Satyam B. Aadhav⁶

^{1,2,3,4,5,6}Dept. of Computer Engineering Sandip Institute of Technology and Research Centre, Nashik

¹anmolsbudhewar@gmail.com, ²ankita.karale@sitrc.org, ³meetghodasaraa@gmail.com, ⁴sujitkhalkar28@gmail.com,

⁵omdhoble2005@gmail.com, ⁶satyamaadhav@gmail.com

Peer Review Information	Abstract
Type: Article Received: 23 July 2026 Revised: 26 Aug2026 Accepted: 20 Sept 2026 Published: 24 Oct 2026	Abstract Inspection systems for high speed assembly lines need precision, low latency, robustness to sparse defect examples and traceability at the production-event level. Current state of the art in industrial vision only addresses one layer of this problem, i.e. imaging, detector accuracy, anomaly modeling or embedded inference. This paper presents an integrated autonomous visual-inspection system combining triggered picture acquisition and photometric standardization with an RT-DETR detection core, synthetic minority-defect augmentation, confidence-aware accept/reject logic and a permanent quality-trace layer. We contextualize the contribution through a research-trace analysis over rule-based machine vision, one-stage CNN detectors, industrial anomaly detection, DETR-family end-to-end detection and modern line-scan deep-learning systems. We establish a design model that relates conveyor speed, sampling density, end-to-end latency, false-accept risk and computing cost . Experimental measurements at the project scale are not yet available, so the paper does not claim any fabricated performance but defines a reproducible evaluation protocol and analytically benchmarks the design choices against published evidence such as RT-DETR, DeepPCB, MVTec-based anomaly methods, TinyML quality inspection and line-scan inspection studies. The resulting design is a technically testable architecture for high-speed electronics and discrete-part production, where detection, rejection, logging and model monitoring are performed as one closed loop. Keywords: Industrial Visual Inspection; Defect Detection; RT-DETR; Line-Scan Camera; Synthetic Data; Edge AI; Smart Manufacturing; Traceability.

How to Cite This Article

Budhewar, A. S., Karale, A. V., Ghodasara, M. K., Khalkar, S. B., Dhoble, O. P., & Aadhav, S. B. (2026). Autonomous defect detection system for high-speed assembly lines. *International Journal on Mechanical Engineering and Robotics*, 15(1), 1–14.

Introduction

Quality control in manufacturing is shifting from rule-based machine vision and human-interpreted images to data-driven inspection systems that must operate at production speed. Classical machine vision still has its merits, such as deterministic optical control and predictable computation, but becomes fragile when defect geometry, product variances, reflectance or operating conditions change [1]. Cyber-physical manufacturing systems reinforce the expectation that inspection should not be a stand-alone camera task, inspection events should be coupled to production control, analytics and feedback [2]. Recent surveys show that industrial defect detection is rather a mixed problem of visual representation, data scarcity, generalization and deployment than a single classification problem [3], [4].

The architecture proposed in this paper is for autonomous fault detection in a high-speed assembly line. The sequence of operations in the system includes product movement, positioning, image acquisition, preprocessing, AI based flaw detection, pass/fail categorization, dashboard presentation and database storage. RT-DETR is selected as the main detection backbone and augmentation with GAN or diffusion is investigated for rare defect samples. These two themes are choices in modern study. First, Kim et al. demonstrated a high spatial resolution under continuous motion with line-scan and high-speed optical systems in a deep learning line scan inspection system for PCB related flaws [5]. Secondly, RT-DETR, an end-to-end transformer detector, removes the non-maximum suppression (NMS) from the inference path, while achieving real-time throughput [29].

But a conference quality contribution is more than the sum of existing parts. Then the research topic is formulated at the system level: how to co-design acquisition, detection, decision and traceability such that the inspection loop meets speed and quality limitations while remaining trainable under defect scarcity? This is contrasted to detector only research where a very accurate model may still fail in operation due to unstable picture creation, latency higher than line takt time, confidence thresholds not calibrated properly, or decisions not being logged for downstream quality assessments.

The paper makes four contributions: Firstly, it introduces a closed-loop architecture that combines triggered acquisition, photometric standardization, RT-DETR inference, confidence-aware acceptance logic, actuation and production-event traceability. Second, it gives a formal system model that relates acquisition density, latency, throughput, detection risk and computing cost. Third, it reviews the research path from traditional machine vision to current supervised and anomaly-detection paradigms and the extent to which each meets the requirements for high-speed assembly inspection. Fourth, it provides a reproducible experimental and ablation technique that could be applied to validate the framework without making claims on measurements not yet acquired. This discipline of evidence is important: we use published benchmark statistics as contextual evidence, not as results of the proposed system.

Related Work And Research Trace

From Rule-Based Vision to Learning-Based Inspection

Historically industrial machine vision has been a combination of controlled lighting, optics, triggering, geometric measurement, thresholding, template matching and designed features [1]. These systems work well when the distribution of appearance is limited, but they need to be re-trained when the types of defects or the production conditions change. Examples of the trend of the learnt representations are semantic-segmentation based optical inspection [6] and multiscale feature learning systems for carbon-fibre inspection [7]. Quality inspection with TinyML has also shown that meaningful visual inference can be pushed to the edge for constrained hardware, but model capacity, memory, and sensor variability become significant design constraints [8].

This technique is especially well-suited to the inspection of printed-circuit and electronics because defects can be small compared to the total field of view and conductive or soldered surfaces are highly reflective. Liu and Qu used machine vision to identify the defects of PCB online [9]. DeepPCB is a dataset proposed by Tang et al. It contains 1,500 aligned pairs of images and six common classes of defects. They reported a detector that achieved 98.6% mAP at 62 FPS on their dataset [10]. Kim et al. added a movable side-view line-scan camera, preprocessing, a deep-learning engine and a management platform and claimed 99.5% accuracy for seven fault categories in their experimental environment [5]. These experiments validate the imaging system architecture and highlight that the claimed accuracy depends on the acquisition geometry, dataset and defect taxonomy.

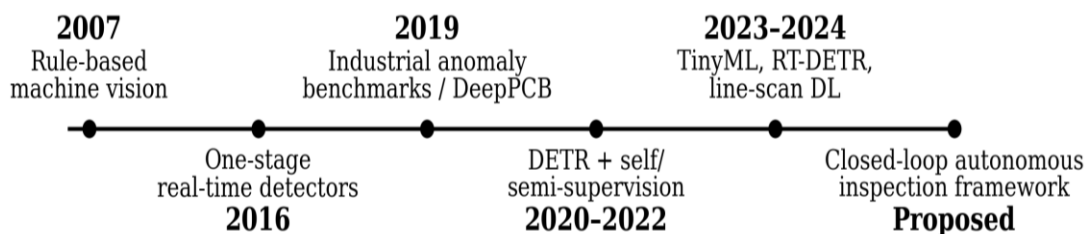


Fig. 1. Research trace motivating the proposed closed-loop inspection framework.

Supervised Detection, Transformers, and Anomaly Methods

The YOLO family [17]–[20]] demonstrated the practical value of one-stage detection for real-time application. For example, Faster R-CNN [21] and SSD [22] are two-stage and single-shot reference points, whereas focal loss deals with the foreground-background imbalance in dense detection [23]. DETR reformulated object detection as set prediction thru bipartite matching [24]. Additional improvements in convergence and multiscale efficiency were achieved thru deformable attention [25], conditional queries [26], denoising [27], and improved anchor-box denoising [28]. RT-DETR is designed mainly to overcome the speed barrier of transformer detection: Zhao et al. get 53.1 AP and 108 FPS with RT-DETR-R50 on a T4 GPU in the COCO test, with end-to-end inference without NMS [29].

While these COCO numbers are not directly comparable to industrial datasets, the architectural property of NMS-free set prediction with variable-depth decoder is relevant for latency-bounded inspection.

Typically, industrial defect data breaks the premise that all defect modes can be labeled at scale. MVTec AD [11] proposed a well-known benchmark for learning from nominal data and MVTec LOCO AD [12] extended the topic to structural and logical anomalies. CutPaste [40], DRAEM [39], PaDiM [41], PatchCore [38] claim that anomaly detection can expose unobserved deviations from normal data or synthetic abnormalities. Their study obtains 96.6 AUC with transfer of ImageNet [40]. For image-level MVTec AD [38], PatchCore achieves 99.6% AUROC. These metrics are not directly comparable to supervised mAP, but the methodologies show a useful design principle: a production system should not be reliant solely on a closed set of defect labels. A future direction could be to merge RT-DETR for known flaws with an anomaly branch for unknown or unusual deviations.

Lack of data also motivates augmentation and semi-supervised learning. In a controlled industrial case (Tabernik et al. [13]) surface defect segmentation can be trained with relatively few defective examples. Jin and Chen talk over small-labeled-data approaches, including augmentation, transfer learning, semi-supervision, and unsupervised techniques [14]. Goodfellow et al. [31] proposed GANs and diffusion probabilistic models provide a stable alternative for generation [32]. General data augmentation can increase invariance [33] and tools such as Albumentations enable reproducible pipelines of transformations [34]. [15] Budhewar et al.: Ensemble and multi-feature deep-learning for robustness across application areas. Their work on semi-supervised decentralized learning further proves the ability to exploit the partially labeled data in the case of unavailability of the full annotation [16]. In this work we use these references specifically to justify robust representation and label-efficiency tactics, not as direct evidence of industrial defect-detection ability.

Table 1. Representative Research Evidence and Its Implication for The Proposed Design

Study	Inspection / benchmark	Reported evidence	Strength	Unresolved system issue
Tang et al. [10]	DeepPCB, 6 PCB classes	98.6% mAP; 62 FPS	PCB-specific real-time detection	Aligned-pair dataset; limited deployment loop
Bergmann et al. [11]	MVTec AD	15 industrial categories	Normal-only anomaly benchmark	No direct actuation / takt constraint
Roth et al. [38]	MVTec AD	Up to 99.6% image AUROC	Strong cold-start anomaly detection	AUROC not equivalent to detector mAP
Zhao et al. [29]	COCO	RT-DETR-R50: 53.1 AP; 108 FPS	NMS-free real-time detection	Generic benchmark, not industrial imaging
Kim et al. [5]	Line-scan PCB inspection	99.5% accuracy; 7 defect types	Integrated imaging + AI + monitoring	Limited defect set; hardware-specific
Albanese et al. [8]	Product quality / TinyML	Edge-oriented inspection study	Resource-aware deployment	Capacity and sensor robustness limits

Table I is intentionally heterogeneous: accuracy, mAP, AP, and AUROC are task-specific and must not be ranked as if they were the same statistic. The table is used to trace design capabilities and gaps, not to manufacture a leaderboard.

Research Gap and System-Level Requirements

Literature shows constant dissociation between sensing, recognition and production control. Detector papers usually assume the input image is available and well-exposed. Optical-inspection papers investigate image formation, but can take advantage of a limited defect taxonomy. Anomaly-detection papers handle cold-start learning, but usually do not model deterministic actuation. Embedded-AI papers focus on resource budgets, but do not necessarily address quality-risk asymmetry. The proposed study looks at the intersection of these issues. Its

originality is thus architectural and methodological: the defect decision is considered the result of a calibrated chain, both the upstream image construction and the downstream actuation of which are observable.

In a high-speed assembly environment, five criteria are non-negotiable. R1 is spatial observability: the minimal flaw of interest must be visible to enough sensor samples after optics and motion are accounted for. R2 is about temporal feasibility. The end-to-end decision shall be made before the physical reject point or before the queue becomes unstable. R3 is defect-data efficiency: the training procedure should still be useful when there are few labeled examples of unusual failure scenarios. R4 is risk-aware decision making: thresholds have to be selected according to false-accept and false-reject cost, not a single global confidence value. R5 is traceability: every final decision can be traced back to the image, model version, threshold configuration, time stamp and actuation consequence.

Table 2. Research-Gap Mapping To Proposed Design Responses

Gap observed in literature	Operational consequence	Proposed response	Validation evidence
Imaging and detector treated independently	Blur/illumination variation enters the model as uncontrolled noise	Encoder-triggered acquisition + versioned preprocessing	Resolution, blur and illumination stress tests
Rare labelled defects	Minority classes dominate miss risk	Conventional + synthetic minority augmentation	Per-class AP/recall; real-only test set
Detector FPS reported without plant timing	Reject signal may arrive too late	End-to-end latency budget including communication/actuator	Median, P95 and worst-case T_E2E
Single global threshold	Unequal defect costs ignored	Class-specific confidence and process rules	False-accept / false-reject curves
Prediction not linked to production record	Difficult audit and drift diagnosis	Event ID, model version and quality database	Trace completeness and replay audit

Problem Formulation and Design Requirements

Inspection Timing and Image Formation

Let a product move at conveyor velocity v (mm/s). For a line-scan camera, let the encoder generate f_t trigger lines per second. The along-motion sampling pitch s (mm/line) is approximately

$$s = \frac{v}{f_t} \quad (1)$$

and should not exceed the spatial sampling required to resolve the smallest defect of interest. If a minimum defect dimension $d_{\min}$ must cover at least k samples after optical projection, a conservative acquisition constraint is $s \leq d_{\min}/k$. This makes camera triggering part of the detection design rather than a separate hardware setting. For area-scan acquisition, an analogous condition constrains exposure time so that motion blur remains below a specified fraction of the projected defect width.

End-to-End Latency and Throughput

The usable inspection latency includes image transfer, preprocessing, neural inference, decision logic, communication, and actuator response. Denoting these terms by T_{acq} , T_{pre} , T_{inf} , T_{dec} , T_{com} , and T_{act} , respectively, the end-to-end latency is

$$T_{E2E} = T_{acq} + T_{pre} + T_{inf} + T_{dec} + T_{com} + T_{act} \quad (2)$$

For a product pitch p (mm) and line speed v , the available takt interval is $T_{takt} = p/v$. A stable one-product-at-a-time pipeline requires $T_{E2E} \leq T_{takt}$ unless acquisition and inference are explicitly pipelined. If a queue is used, the sustainable service rate must exceed the arrival rate. This distinction is critical because a detector can report high FPS while the physical rejection path still misses its timing window.

Decision Risk

For quality control, false acceptance is typically more costly than false rejection. Let C_{FA} and C_{FR} denote their application-specific costs and let P_{FA} and P_{FR} denote observed rates. A simple decision-risk term is

$$R_q = C_{FA}P_{FA} + C_{FR}P_{FR} \quad (3)$$

which allows confidence thresholds to be tuned against manufacturing risk rather than detector accuracy alone. The final design objective can be written as a multi-objective scalarisation

$$J = \lambda_1(1 - mAP) + \lambda_2 T_E 2E + \lambda_3 R_q + \lambda_4 C_{comp} \quad (4)$$

where C_{comp} represents computational cost or energy and λ_i encode deployment priorities. The objective does not imply that all terms share a natural scale; normalisation on validation data is required before optimisation.

Proposed Methodology

Overall Autonomous Inspection Architecture

Figure 2 presents the proposed architecture. The inspection plane moves and creates images deterministically, then perception and production are learnt. The dataset, model lifespan and quality records, and operator analytics are hosted in a parallel learning-and-information plane. This separation mitigates a common failure mode in project prototypes: tightly combining camera control, model code and dashboard logic such that changes in one component destabilize the whole system.

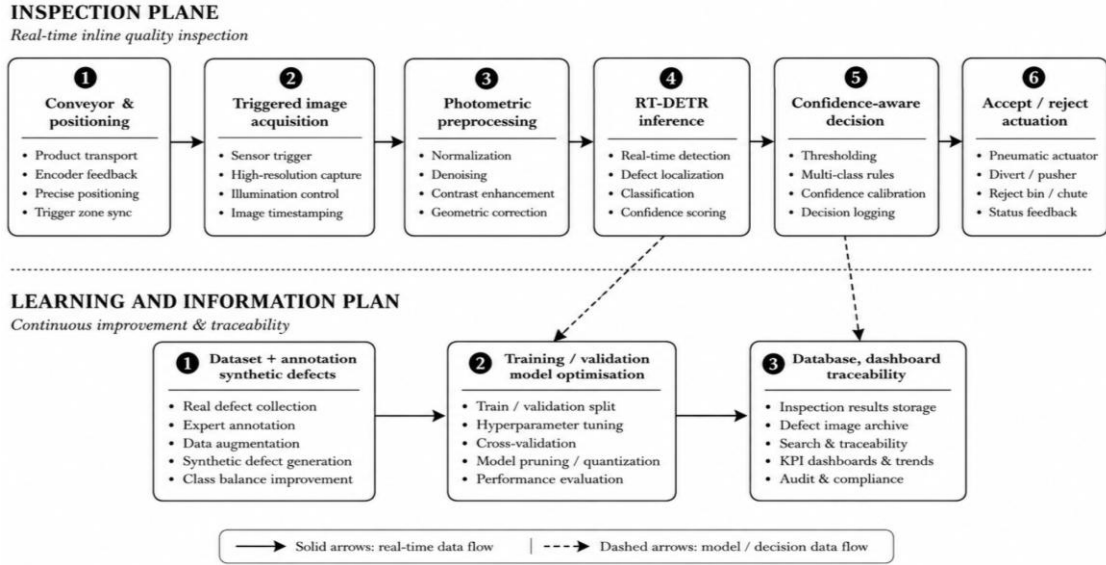


Fig. 2. Proposed autonomous high-speed defect-inspection architecture.

Image Acquisition and Photometric Preprocessing

The acquisition subsystem supports either a line-scan camera for continuous motion or a high-frame-rate area-scan camera where full-frame exposure is preferable. A trigger/encoder signal assigns each product or scan segment to a deterministic spatial position. Mechanical fixturing, working distance, lens selection, and illumination geometry are treated as fixed calibration variables. For reflective solder, metallic leads, and polished surfaces, diffuse or multi-directional illumination should be considered so that defect contrast is not dominated by specular hotspots. Preprocessing is limited to operations that preserve defect evidence: geometric orientation correction, photometric normalisation, denoising were justified by sensor noise, and region-of-interest extraction. Aggressive smoothing is avoided because scratches, cracks, lifted leads, and solder bridges may occupy only a few pixels. Every preprocessing transform is versioned with the model so that the same pipeline is applied during training and inference. Standard calibration and image-formation principles remain applicable [42], while OpenCV provides a suitable implementation layer for deterministic image operations [37].

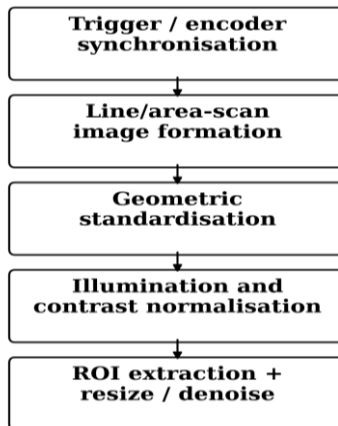


Fig. 3. Acquisition and preprocessing sequence.

Dataset Construction and Synthetic Minority-Defect Augmentation

The dataset schema stores product identifier, acquisition configuration, image or scan identifier, defect class, bounding box, pass/fail status, annotator, and timestamp. A product-aware split is required: images of the same physical unit, lot, or near-duplicate view must not leak across training and test partitions. Where multiple viewing angles are captured, the split should be performed by product instance before views are generated. This protects the evaluation from inflated performance due to correlated samples.

Conventional augmentation includes bounded changes in brightness, contrast, translation, scale, blur, and noise only when they reproduce plausible production variation [33], [34]. For rare defects, synthetic augmentation can be introduced in two stages. First, defect-aware copy-paste or procedural perturbations provide a transparent baseline. Second, GAN or diffusion generation [31], [32] may be used to increase appearance diversity. Synthetic images are retained with provenance labels and are never allowed to dominate the held-out test set. The test set remains real-only so that improvements reflect transfer to actual defects rather than fidelity to the generator.

Table 3. Initial Defect Taxonomy And Annotation Intent

Defect family	Typical visual cue	Preferred label	Process concern
Missing component / ball	Expected object absent	Box + class or logical rule	Electrical/open-circuit failure
Bridge / short	Connected solder or conductive region	Box; optional mask	Short circuit
Double / excess material	Abnormal duplicate or excess deposit	Box + class	Assembly/solder quality
Foreign material	Unexpected local object/texture	Box; anomaly score backup	Contamination
Placement / polarity error	Object shifted, rotated or reversed	Box + geometric rule	Functional assembly error
Damage / crack / scratch	Local discontinuity or surface break	Small box; mask if feasible	Mechanical/reliability risk
Lifted / extra ball or lead	Vertical/side-view separation cue	Box + multi-view association	Poor connection

The taxonomy is deliberately process-oriented rather than model-oriented. Some categories are naturally localised by bounding boxes, whereas placement and missing-object errors can also require geometric or logical constraints. If a class cannot be reliably represented by a box, the annotation schema should be extended instead of forcing the detector to learn an ambiguous target.

RT-DETR Detection Core

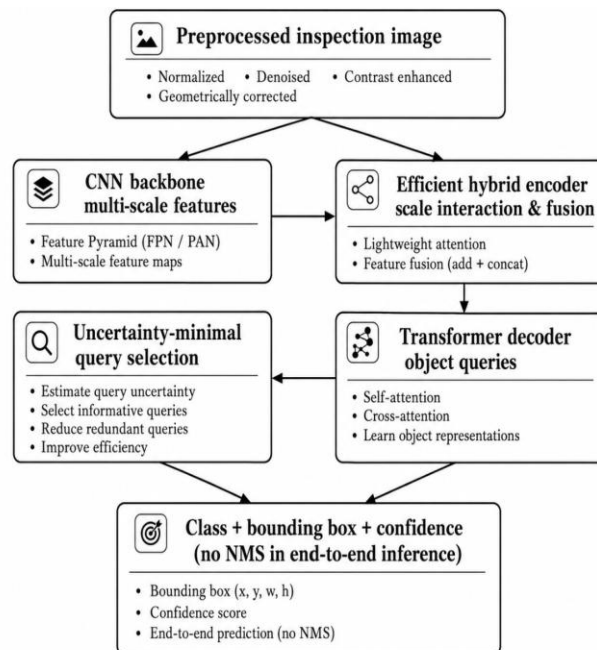


Fig. 4. RT-DETR-oriented detection module used in the proposed framework.

RT-DETR is selected because it provides end-to-end set prediction and removes NMS from the real-time path [29]. The backbone extracts multiscale feature maps and may use residual feature extraction principles [30]; an efficient hybrid encoder performs intra-scale interaction and cross-scale fusion; uncertainty-minimal query selection seeds high-quality object queries; and the decoder predicts class probabilities and bounding boxes. This design is particularly attractive where the inference budget is strict because decoder depth can be adjusted to trade accuracy for speed without rebuilding the acquisition and control layers.

For an image with ground-truth set G and prediction set P , training follows DETR-style bipartite matching [24]. A simplified detection loss is

$$L = \alpha L_{cls} + \beta L_{L1} + \gamma L_{GIoU} \quad (5)$$

where L_{cls} is the classification term, L_{L1} is the box-coordinate loss, and L_{GIoU} penalises poor overlap. Class weighting or focal-style modulation [23] can be considered when defect prevalence is highly imbalanced. The exact loss formulation should follow the selected RT-DETR implementation to preserve reproducibility.

Training, Calibration, and Model Selection

Training begins from an officially released pretrained checkpoint rather than from random initialisation unless the dataset is unusually large. Candidate input resolutions are selected from the smallest-defect pixel footprint and GPU memory. The optimiser, learning-rate schedule, weight decay, warm-up policy, gradient clipping, number of epochs, early-stopping rule, random seed, and mixed-precision setting are recorded in a run manifest. At least three independent seeds are recommended for the final comparison so that a single favourable initialisation does not determine the reported result. Hyperparameters are selected on the validation set only.

Model selection uses a constrained criterion rather than maximum mAP alone. A candidate is eligible only if its P95 end-to-end latency fits the target takt budget; among eligible models, the validation objective in (4) is minimised. This rule prevents a slow high-accuracy model from being selected for a line on which it cannot physically act. Confidence calibration is performed after the detector weights are fixed. Class-specific thresholds are estimated from precision-recall curves and the quality cost matrix, with final thresholds frozen before test evaluation.

Algorithm 1. Deployment-oriented training and selection procedure

- 1: Split data by physical product/IOT into train, validation, and real-only test partitions.
- 2: Fit deterministic preprocessing on training conditions; freeze its configuration.
- 3: Train baseline detector with conventional augmentation; record all run metadata.
- 4: Add synthetic minority-defect samples only to the training partition and retrain.
- 5: Profile candidate models at batch size 1 on target hardware; reject those violating P95 latency.
- 6: Select the eligible model minimising validation objective J in (4).
- 7: Calibrate class-specific confidence thresholds using validation precision-recall and cost curves.
- 8: Lock weights, thresholds, preprocessing, and software versions; evaluate once on the test set.
- 9: Report per-class metrics, latency distribution, resource use, and representative failure cases.

Confidence-Aware Decision and Automatic Rejection

Only after these class-specific confidence thresholds are applied does the detector output convert to a manufacturing decision. If any of the mandatory-reject classes exceed their thresholds, or if a configurable rule based on count, location or severity is satisfied, the product is marked defective. This decouples perception from process policy: a model update does not implicitly change acceptance criteria. When the decision message is to carry a product/event identifier to the PLC/microcontroller which schedules the reject actuator at the downstream location, the physical prototype should be available. If the health checks on the camera, inference, or comms fail, then the watchdog logic must revert to a safe state.

Edge Deployment and Traceability

The deployment target is an industrial PC or an edge device and is selected after profiling the final model. Model compression may include pruning and quantization following the standard compression principles [35], [36], but only if the accuracy and calibration are revalidated. The database stores image references, predictions, thresholds, decision outcomes, latency, and model version. This allows a later quality audit to reconstruct why a product was rejected and supports drift analysis across lots, shifts, or illumination changes.

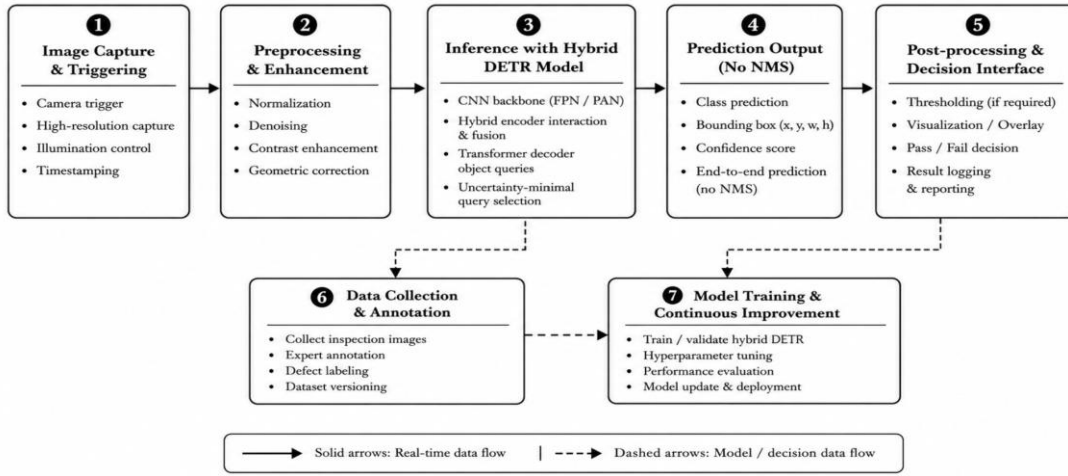


Fig. 5. Closed-loop decision, actuation, and quality-trace layer.

Evaluation Methodology

Detection and Classification Metrics

Evaluation is performed on a real-only held-out test partition. For a fixed IoU threshold, precision, recall, and F1 score are

$$Precision = TP / (TP + FP) \quad (6)$$

$$Recall = TP / (TP + FN) \quad (7)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (8)$$

Bounding-box overlap is measured by

$$IoU = \frac{|B_p \cap B_g|}{|B_p \cup B_g|} \quad (9)$$

and detection quality is reported using AP per class, mAP@0.5, and mAP@0.5:0.95. Because missing a true defect can have higher business cost than rejecting a good unit, confusion statistics are also converted into false-accept and false-reject rates at the product level. Confidence calibration should be inspected using reliability plots or expected calibration error if decisions are threshold-sensitive.

Real-Time and Resource Metrics

Runtime reporting must name the CPU/GPU or accelerator, input resolution, batch size, numerical precision, software stack, and whether data-transfer time is included. Average network inference alone is less informative than the median, 95th-percentile and worst-case T_{E2E}. Throughput under the current acquisition plan is reported as products per second. For comparison, we report deployment and memory footprint, model parameters and approximate FLOPs or MACs.

Baselines and Ablation Plan

The main supervised baselines should be a current YOLO configuration and the suggested RT-DETR configuration trained on the same split, resolution, augmentation policy and compute budget. A two-stage detector such as Faster R-CNN [21] can provide an accuracy-oriented reference when feasible. For scarce-label conditions, an anomaly detector such as PatchCore [38] may be evaluated as a complementary baseline, but its AUROC must not be merged with mAP. The planned ablations isolate: (i) raw images versus photometric preprocessing; (ii) no augmentation versus conventional augmentation versus synthetic minority-defect augmentation; (iii) standard versus reduced decoder depth; (iv) multiple input resolutions; and (v) global versus class-specific confidence thresholds.

Table 4. Fair-Comparison Matrix For Model Families

Model family	Primary role	Post-processing	Expected advantage	Fair-comparison condition
YOLO-class one-stage	Known-defect baseline	Typically, NMS	Mature real-time ecosystem	Same data, resolution and augmentation
Faster R-CNN	Accuracy-oriented baseline	Proposal/NMS pipeline	Strong two-stage reference	Same backbone scale where feasible

RT-DETR	Proposed known-defect core	End-to-end; no NMS	Flexible decoder-depth speed tuning	Same hardware and batch=1 timing
PatchCore / anomaly	Unknown/rare anomaly complement	Anomaly threshold	Can learn from nominal data	Report AUROC separately from mAP

Statistical Validity and Failure Analysis

The final study should report uncertainty rather than only a single best checkpoint. For seed-based repeats, mean and standard deviation of validation and test metrics are reported, while latency is summarised with empirical quantiles. Bootstrap confidence intervals can be used for product-level false-accept and false-reject rates when the number of defective products is limited. If two detectors are evaluated on the same test products, paired comparisons are preferred because the samples are not independent across models. Statistical testing should be secondary to effect size and engineering relevance; a small change in mAP may be unimportant if it materially worsens tail latency or false acceptance.

Failure analysis is mandatory for safety-relevant quality control. False negatives are grouped by defect size, class, illumination condition, location, and product variant. False positives are grouped by reflectance, benign texture, fixture occlusion, and annotation ambiguity. At least one representative image from each major failure mode should be shown in the final experimental paper. This analysis determines whether the next research step should change the model, optics, annotation policy, or process threshold.

Table 5. Minimum Reproducibility Checklist For The Experimental Phase

Category	Required disclosure	Reason
Data	Class counts; product-aware train/val/test split; annotation protocol	Prevents leakage and enables replication
Imaging	Camera type, lens, illumination, exposure/line rate, working distance	Links performance to image formation
Training	Model variant, resolution, epochs, optimiser, learning rate, seed, augmentations	Separates algorithm effects from tuning
Runtime	Hardware, precision, batch=1 latency, end-to-end latency distribution	Tests takt-time feasibility
Decision	IoU and confidence thresholds; reject policy; cost assumptions	Connects ML metrics to manufacturing risk
Reporting	Per-class AP/recall, false accept/reject, failure cases	Exposes minority-class weaknesses

Analytical Validation And Research-Trace Discussion

Why the Framework Is More Than a Detector Swap

The research trace in Section II shows that each generation solves a different part of the inspection problem. Traditional machine vision provides deterministic image formation but is sensitive to appearance change [1]. Real-time detectors based on CNN alleviate the complexity of feature engineering [17]–[20] but are vulnerable to NMS and class imbalance. DETR family detectors simplify end-to-end prediction [24]–[29]. Industrial anomaly approaches minimize the dependence on fault labeling [38]–[41]. Line-scan systems meet the spatial sampling requirements of continuous motion [5]. The suggested design purposefully composes these lessons at their right abstraction levels : optics and triggering stabilize data, RT-DETR deals with known-object localization, augmentation deals with unusual classes, and traceability ends the production cycle.

Evidence-Based Performance Expectations

Feasibility is supported by published evidence, but not performance on the requested data set. In the experimental setting of DeepPCB, PCB fault localization is shown at >60 FPS [10]. RT-DETR shows that end-to-end transformer detection may achieve real-time performance on a T4 GPU at generic-object scale [29]. Kim et al. demonstrate the integration of line-scan acquisition, deep learning and monitoring in an industrially focused PCB inspection platform [5]. High anomaly-detection performance enables training with only nominal data, as shown in PatchCore and other anomaly algorithms [38]–[41]. The results justify the architectural decisions but the suggested system needs to be evaluated under its own optics, defect distribution, product speed and hardware.

Accuracy–Latency–Risk Trade-Off

The main design trade-off isn't only precision vs FPS. Higher input resolution improves visibility of small errors but also increases T_{inf} and memory. The larger the augmentation, the stronger, but also more unrealistic, the textures.

Lowering the confidence levels will increase the recall and possibly the false reject rate. RT-DETR achieves faster inference speed. But the detection quality may be decreased by using smaller number of decoder layers. Therefore, Equation (4) provides a rigorous way to select an operating point: parameters are tuned on validation data and the final test set is used only once for unbiased reporting.

Known and Unknown Defects

Closed-set detectors are most effective when the kinds of fault are operationally defined and relevant instances are widely available. Manufacturing, however, regularly confronts new combinations of contamination, mechanical damage, placement error, and process drift. Thus, an anomaly branch is a plausible future extension. MVTec LOCO AD precisely distinguishes structural anomalies, e.g., dents or pollution, from logical anomalies, e.g., missing or misplaced items [12]. This distinction maps readily to assembly inspection.

A hybrid system can route high-confidence known defects thru RT-DETR with an anomaly score serving as a secondary safeguard for unmodelled events. Such a branch should be examined independently to avoid mixing supervised and unsupervised metrics

Traceability as a Research Variable

Traceability is generally seen as a feature of software, but it modifies what can be learned from deployment. The system builds a longitudinal dataset to detect drift and recurrent process errors by storing product ID, lot, acquisition parameters, model version, thresholds, prediction set, decision, actuation acknowledgment, and operator override. The trace can for example show if an increase in false rejects is related to a certain lighting setup, shift, supplier lot or model release. This converts inspection from a binary endpoint into a measurement instrument for manufacturing improvement.

The trace layer also enables post-deployment evaluation of calibration. Predictions can be stratified by confidence decile and compared with later audit outcomes. A model whose 0.9-confidence detections are correct only 70% of the time should not be treated as trustworthy simply because its aggregate mAP is high. Accordingly, the proposed framework treats model monitoring and trace completeness as acceptance criteria alongside vision performance.

Research Questions and Acceptance Criteria

The framework can be converted into testable research questions rather than evaluated as a demonstration. RQ1 asks whether controlled preprocessing improves minority-defect recall without a disproportionate increase in false positives. RQ2 asks whether RT-DETR provides a more favourable end-to-end accuracy-latency operating point than a matched modern one-stage baseline when both are trained under the same data and compute conditions. RQ3 asks whether synthetic minority-defect augmentation improves real-only test performance for scarce classes. RQ4 asks whether class-specific, cost-aware thresholds reduce product-level false acceptance at an acceptable false-reject penalty. RQ5 asks whether the complete pipeline, including image transfer and actuation communication, satisfies the production takt requirement with adequate tail-latency margin.

Each research problem has an a priori criterion for acceptance. For RQ1 and RQ3 the improvement should be on the unchanged and consistent real test set, not on augmented validation data. For RQ2, we consider a detector deployment-competitive if it satisfies the P95 end-to-end latency budget. We do not think that breaking the takt time, which increases accuracy, is a successful consequence. When answering RQ4, the selected operating point should be proportional to the cost of false acceptance and false rejection in particular. RQ5 - the delay is measured over long enough production-like sequences to capture jitter from scheduling, memory transfer and communication.

The question-based method also avoids a common methodological trap: picking a story based on the data. Log hypotheses, splits, metrics and main comparisons in project experiment log before final test evaluation Exploratory analyzes are still useful but are presented as exploratory and are not included in the confirmatory comparison. Thus, the final paper can make a limited but defensible claim e.g. that synthetic augmentation improved rare-class recall under a fixed latency budget rather than a broad claim that the whole system is uniformly better.

Industrial Implementation, Limitations, And Future Work

Implementation Path

A phased adoption is recommended. Stage 1 adjusts the imaging geometry and collects representative normal and faulty products. Stage 2: Build the product-aware dataset and create a replicable baseline. Stage 3 combines RT-DETR inference with the decision service on an industrial PC. Stage 4 links event identifiers to the actuator and database. In Stage 5, once the base system is stable, we introduce model monitoring, synthetic augmentation, and optional anomaly detection. This sequence avoids disguising hardware or annotation issues with progressively sophisticated models.

Operational Acceptance Test Plan

An operational acceptance sequence should be followed before an autonomous reject mechanism is enabled. The first phase is the metrology validation. A calibration target or known geometry is scanned over the usable field to evaluate the spatial scale, distortion, focus and repeatability. Line-scan operation is verified with encoder-to-line synchronization at various conveyor speeds.

The motion blur testing is done for strobe timing and exposure for area scan operation. The minimum pixel footprint used in the model design needs to be smaller than the smallest significant fault. Before corrections, the comparison should be to a camera that cannot reproduce the required signal. When comparing detectors, as understood.

The second phase is the repeatability and reproducibility. A sample group of products checked over time, upon restart and where safe after small changes in operator configuration. Confidence Position of bounding box Distribution of choice outcome A lot of volatility, and just feeling out of whack even if the average is good. This phase should include normal parts that contain reflecting or textured areas known to produce false positives. Stable false positive behavior is essential for practical adoption.

The third phase is the validation of timing. A product stream with realistic pitch is processed for a sustained interval while T_{acq} , T_{pre} , T_{inf} , T_{dec} , T_{com} and actuator acknowledgment are independently timestamped. The tracing then shows if the latency spikes are in the neural network, the operating-system scheduling, the storage, the networking, or the mechanical reaction. The line must contain safety buffer δ such that $P95(T_{E2E}) + \delta \leq T_{takt}$. The buffer is not a universal constant, but is derived from the observed jitter and the effects of a missed reject window. Queue depth and dropped-frame counters are kept together.

The fourth phase is validation of the decision by controlled combination of good and defective items. The final product level result is recorded at the physical reject station instead of just reporting the image level detection. A false accept is a non-conforming unit that is passed by the station. A false reject is a conforming unit that is rejected. The test should be stratified by defect class and defect size. 2. In problematic cases, if the manufacturing process allows it, a blind second inspection or destructive verification can be used as an independent reference.

The fifth phase is traceability and recovery validation. For randomly selected inspection events, an auditor should be able to reconstruct the image, preprocessing version, model hash or version, confidence threshold, predicted boxes, final decision, actuator acknowledgement, and operator override if present.

Then the system is intentionally put into recoverable faults such as camera disconnection, model-service restart, database unavailability or delayed PLC acknowledgment. The test specifies the required fail-safe behavior. If a communication failure can silently turn an uncertain event into an accepted product, then autonomous rejection should not be enabled. **Table 6. Proposed Factory Acceptance Tests Before Autonomous Rejection**

Test	Primary measurement	Pass concept	Failure implies
Optical/metrology	Pixel scale, focus, distortion, repeatability	Critical defect remains observable across field	Fix optics/illumination before ML tuning
Acquisition timing	Trigger jitter, blur, dropped lines/frames	Stable capture at min/max planned speed	Revise encoder/exposure/strobe
Inference timing	Median/P95/worst T_{E2E} , queue depth	Tail latency fits takt with safety margin	Optimise model/pipeline or hardware
Decision quality	Per-class FA/FR and physical reject outcome	Risk targets met at frozen thresholds	Recalibrate, retrain or modify policy
Trace audit	Fraction of events fully reconstructable	All sampled events replayable	Repair data model/versioning
Fault recovery	Fail-safe response and restart integrity	No silent accept on critical service failure	Add watchdog/interlock logic

Data Lifecycle and Drift Monitoring

The industrial data after deployment are no longer identically distributed. New suppliers, solder chemistry, surface finish, maintenance, lens replacement, lighting aging, camera gain, and product variances can all change the image distribution. The suggested trace layer enables drift monitoring thru inexpensive summary statistics on picture brightness, contrast, prevalence of defect classes, confidence distributions and feature embeddings. A change in these data does not necessarily indicate model failure, but does tell us where to do focused manual sampling.

Instead, we use a regulated feedback procedure rather than continuous on-line retraining. Suspect and low confidence events are sampled for review; new version of the dataset is created with revised labels and retraining is done offline. The candidate model is then tested against a new challenge set and a fixed regression set of previously accepted production conditions. We only allow deployment if the new model improves the desired failure mode without violating the stated latency and regression criteria. The previous model package is retained for roll back purposes.

Data should be kept in a risk-based way. We cannot keep all the full resolution photos forever, but the system can keep all the defective and uncertain cases, a statistically significant sample of normal cases, and compact metadata for each event. If the production photography contains proprietary geometry or customer data then access control and retention policies are part of the study implementation. These criteria also make a subsequent conference extension credible, since the claimed dataset can be documented without revealing sensitive industrial material.

Limitations

The structure and analysis provided here are not based on a completed industrial trial. The most basic disadvantage is the absence of project specific measurements. The precise fault taxonomy may also be different, depending on the selected assembly product. In addition, the line-scan hardware and the reject-actuation process have to be physically integrated and validated.

Synthetic generation may lead to artifacts that can hurt performance in the real world. Deployment risk continues from domain transition, camera replacement, aged lighting, product finish and line speed.

Future Research

Future work should first produce the real dataset and execute the evaluation protocol in Section V. Once a reproducible supervised baseline exists, three extensions are technically justified: unknown-defect screening using anomaly detection [38]–[41], semi-supervised learning to exploit unlabelled production data [16], and edge optimisation through quantisation or structured compression [35], [36]. A digital-twin or adaptive scheduling layer may later be considered for multi-line resource allocation, but such functionality should be added only if it addresses an observed production bottleneck rather than for architectural novelty alone.

Reproducibility, Data Governance, and Publication Discipline

For conference submission, the manuscript should distinguish three evidence classes: measurements produced by the project, numbers reported by prior literature, and planned evaluation targets. This distinction is maintained throughout the present paper. Dataset versions and train/validation/test manifests should be archived; preprocessing and annotation guidelines should be fixed before final testing; code dependencies and random seeds should be recorded; and any images that contain proprietary manufacturing information should be anonymised or released only under an appropriate data-sharing agreement.

References are included only where they provide substantive methodological or contextual support. Ensemble-learning and semi-supervised strategies are cited in the specific contexts of representation robustness and label efficiency [15], [16], while the remainder of the bibliography is focused on industrial vision, detection, anomaly modelling, augmentation, and edge deployment.

Conclusion

This paper reformulates high-speed assembly-line defect detection as a closed-loop manufacturing problem rather than a stand-alone computer-vision benchmark. The proposed framework combines synchronised acquisition, conservative preprocessing, product-aware data management, synthetic minority-defect augmentation, RT-DETR end-to-end detection, confidence-aware acceptance logic, automatic rejection interfaces, and event-level traceability. A research-trace analysis shows how the design draws complementary lessons from classical machine vision, one-stage CNN detectors, DETR-family methods, industrial anomaly detection, and recent line-scan systems. Formal limitations include sampling, takt time, latency, false-accept risk and computational cost. The design can be verified with a large experimental procedure without over-promising unmeasured performance. So the immediate research goal is not to add more model components but to perform a leakage-free hardware-aware evaluation reporting per-class detection quality and end-to-end production latency in real operational conditions.

References

1. H. Golnabi and A. Asadpour, “Design and application of industrial machine vision systems,” *Robot. Comput.-Integr. Manuf.*, vol. 23, no. 6, pp. 630–637, 2007.
2. J. Lee, B. Bagheri, and H.-A. Kao, “A cyber-physical systems architecture for Industry 4.0-based manufacturing systems,” *Manuf. Lett.*, vol. 3, pp. 18–23, 2015.
3. M. Prunella et al., “Deep learning for automatic vision-based recognition of industrial surface defects: A survey,” *IEEE Access*, vol. 11, pp. 43370–43423, 2023, doi: 10.1109/ACCESS.2023.3271748.
4. R. Khanam, M. Hussain, R. Hill, and P. Allen, “A comprehensive review of convolutional neural networks for defect detection in industrial applications,” *IEEE Access*, vol. 12, pp. 94250–94295, 2024, doi: 10.1109/ACCESS.2024.3425166.
5. B. Kim, M. Shin, and S. Hwang, “Design and development of a precision defect detection system based on a line scan camera using deep learning,” *Appl. Sci.*, vol. 14, no. 24, Art. no. 12054, 2024, doi: 10.3390/app142412054.
6. J. Karangwa, L. Kong, D. Yi, and J. Zheng, “Automatic optical inspection platform for real-time surface defects detection on plane optical components based on semantic segmentation,” *Appl. Opt.*, vol. 60, no. 19, pp. 5496–5506, 2021, doi: 10.1364/AO.424547.

7. D. Li, S. Hua, Z. Li, X. Gong, and J. Wang, "Automatic vision-based online inspection system for broken-filament of carbon fiber with multiscale feature learning," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022, doi: 10.1109/TIM.2022.3180423.
8. Albanese, M. Nardello, G. Fiacco, and D. Brunelli, "Tiny machine learning for high accuracy product quality inspection," *IEEE Sens. J.*, vol. 23, no. 2, pp. 1575–1583, 2023, doi: 10.1109/JSEN.2022.3225227.
9. Z. Liu and B. Qu, "Machine vision based online detection of PCB defect," *Microprocess. Microsyst.*, vol. 82, Art. no. 103807, 2021.
10. S. Tang, F. He, X. Huang, and J. Yang, "Online PCB defect detector on a new PCB defect dataset," *arXiv:1902.06197*, 2019.
11. 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, pp. 9592–9600.
12. P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization," *Int. J. Comput. Vis.*, vol. 130, pp. 947–969, 2022, doi: 10.1007/s11263-022-01578-9.
13. D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *J. Intell. Manuf.*, vol. 31, pp. 759–776, 2020, doi: 10.1007/s10845-019-01476-x.
14. Q. Jin and L. Chen, "A survey of surface defect detection of industrial products based on a small number of labeled data," *arXiv:2203.05733*, 2022.
15. S. Budhewar, P. G. Patil, R. Bodkhe, S. Ghuge, D. Bhavar, and G. Shirsath, "A comprehensive survey on ensemble multi featured deep learning models: Applications, challenges, and future directions," *Open Access Int. J. Sci. Eng.*, vol. 7, no. 7, pp. 23–30, 2024.
16. S. Budhewar, P. G. Patil, and S. M. Kale, "Neighbour-aware cooperation for semi-supervised decentralized machine learning," *Educ. Adm.: Theory Pract.*, vol. 30, no. 5, pp. 2239–2247, 2024, doi: 10.53555/kuey.v30i5.3271.
17. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE CVPR*, 2016, pp. 779–788.
18. J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE CVPR*, 2017, pp. 7263–7271.
19. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv:2004.10934*, 2020.
20. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF CVPR*, 2023, pp. 7464–7475.
21. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.
22. W. Liu et al., "SSD: Single shot multibox detector," in *Proc. ECCV*, 2016, pp. 21–37.
23. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE ICCV*, 2017, pp. 2980–2988.
24. N. Carion et al., "End-to-end object detection with transformers," in *Proc. ECCV*, 2020, pp. 213–229.
25. X. Zhu et al., "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. ICLR*, 2021.
26. D. Meng et al., "Conditional DETR for fast training convergence," in *Proc. IEEE/CVF ICCV*, 2021, pp. 3651–3660.
27. [27] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate DETR training by introducing query denoising," in *Proc. IEEE/CVF CVPR*, 2022, pp. 13619–13627.
28. H. Zhang et al., "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," in *Proc. ICLR*, 2023.
29. Y. Zhao et al., "DETRs beat YOLOs on real-time object detection," in *Proc. IEEE/CVF CVPR*, 2024, pp. 16965–16974, doi: 10.1109/CVPR52733.2024.01605.
30. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.
31. Goodfellow et al., "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.
32. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6840–6851, 2020.
33. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, Art. no. 60, 2019.
34. A. Buslaev et al., "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, Art. no. 125, 2020.
35. S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. ICLR*, 2016.
36. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF CVPR*, 2018, pp. 2704–2713.
37. G. Bradski, "The OpenCV library," *Dr. Dobb's J. Softw. Tools*, vol. 25, no. 11, pp. 120–126, 2000.
38. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proc. IEEE/CVF CVPR*, 2022, pp. 14318–14328, doi: 10.1109/CVPR52688.2022.01392.

39. V. Zavrtanik, M. Kristan, and D. Skočaj, “DRAEM—A discriminatively trained reconstruction embedding for surface anomaly detection,” in Proc. IEEE/CVF ICCV, 2021, pp. 8330–8339.
40. C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, “CutPaste: Self-supervised learning for anomaly detection and localization,” in Proc. IEEE/CVF CVPR, 2021, pp. 9664–9674.
41. T. Defard, A. Setkov, A. Loesch, and R. Audigier, “PaDiM: A patch distribution modeling framework for anomaly detection and localization,” in Proc. ICPR Workshops, 2021, pp. 475–489, doi: 10.1007/978-3-030-68799-1_35.
42. R. Szeliski, Computer Vision: Algorithms and Applications, 2nd ed. Cham, Switzerland: Springer, 2022.