CAD2REPLAY: A TRACEABLE SYNTHETIC-TO-PHYSICAL VALIDATION FRAMEWORK FOR CAD-GROUNDED MOBILE RGB-D INSPECTION

Dmytro Bondar[0009-0003-0548-2467], Yevheniia Basova[0000-0002-8549-4788], Oleksii Vodka[0000-0002-4462-9869]
National Technical University "Kharkiv Polytechnic Institute", 2 Kyrpychova Street, Kharkiv, 61002, Ukraine
Email: yevheniia.basova@khpi.edu.ua

Abstract - Mobile industrial inspection systems combine learned models, RGB-D sensing, geometric processing, temporal logic, and tolerance-based decision rules. Model-level accuracy alone therefore cannot establish whether a deployed system preserves the intended feature identities, measurements, and final inspection outcomes. This paper presents CAD2Replay, a traceable validation framework that links three distinct evidence levels: CAD-controlled synthetic data, deterministic replay of recorded RGB-D observations through the deployed mobile pipeline, and repeated physical captures of the same CAD-grounded cases. The framework keeps these evidence levels analytically separate and uses CAD-derived oracles to verify part identity, feature and defect composition, measurements, failure reasons, and final verdicts. The evaluated corpus contains 11 part families, 275 nominal and defective CAD configurations, and 1,375 labelled synthetic renders. On a fixed 35-image real corpus, Locator versions with nearly identical synthetic validation scores produced real-image accuracies ranging from 0.600 to 0.971, demonstrating that synthetic validation alone was insufficient for model selection. With the selected Locator held fixed, the Inspector that performed best in offline evaluation differed from the one that performed best under complete deployed replay. The strongest replay candidate passed 23 of 34 exact end-to-end cases, with failures concentrated in small edge and corner defects. Two physical acquisition sessions further showed that defect responses were class- and case-dependent rather than uniformly absent. These results do not establish production-level inspection reliability. Instead, they show that CAD2Replay can expose disagreement between evaluation stages, preserve traceable expectations across deployment, and localize failures across preprocessing, inference, feature association, measurement, temporal processing, and verdict generation.

Keywords: Industrial quality control; Mobile spatial inspection; Computer vision; RGB-D sensing; CAD matching; Synthetic data; Ensemble machine learning; Tolerance-based decision-making; Reproducible verification.

1. Introduction

The reliability of industrial defect inspection depends not only on model accuracy but also on the availability of traceable evidence across data generation, software deployment, and physical evaluation. Nominal parts are easy to photograph, whereas controlled defective states are scarce, costly, or undesirable to manufacture. Parametric CAD and synthetic rendering can generate geometry, masks, feature identities, and defect parameters together, but the industrial literature still reports inconsistent evaluation protocols for such data [1]. Synthetic validation alone cannot show whether model conversion, deployed preprocessing, feature association, measurement, temporal logic, and verdict generation preserve the intended CAD-derived expectations.

The problem is sharper in mobile RGB-D inspection. A handset combines colour, depth, confidence, camera intrinsics, and pose, yet consumer LiDAR accuracy changes with range, surface, incidence angle, and acquisition conditions [2]. The final decision may consequently change at localisation, ROI preparation, feature segmentation, CAD association, measurement, temporal stabilisation, or software configuration. Once a live scene disappears, those causes cannot be separated unless the complete observation is retained.

dx.doi.org/10.17683/ijomam/issue25/v2.23

The authors' preceding work established synthetic-to-real instance segmentation and image-based dimensional extraction [3]. A later study introduced mobile 2D/LiDAR inspection [2]. CAD2Replay extends this earlier work from model-level and live-session evaluation to a traceable framework that connects synthetic training evidence, fixed-input replay through the deployed software, and subsequent physical observations while keeping their evaluation units and denominators separate.

Existing studies typically evaluate synthetic-data utility, model accuracy, sensor accuracy, or deployed software testing separately. They do not provide a unified procedure for tracing a CAD-defined case from synthetic generation through deployed RGB-D replay to repeated physical observation. CAD2Replay contributes a traceable CAD-to-training record, full-pipeline fixed-input replay, and post-development paired physical observations.

2. Literature Review

Recent review evidence shows that industrial surface-defect research spans supervised and unsupervised detectors, heterogeneous datasets, and competing accuracy-speed objectives; reliability under the conditions of a specific inspection task remains an open concern [1]. Within synthetic inspection, camera, lighting, and sensor-noise choices materially affect model performance [4]. Procedural pipelines can control defect appearance, geometry, masks, and labels for industrial surfaces [5], while reused-scaffolding inspection demonstrates how synthetic images can support a product-specific quality task [6]. Fulir et al. paired real and synthetic multi-view data for segmentation on a geometrically complex, reflective metal clutch part and compared three segmentation architectures [7]. Antiles and Talathi released synthetic and real stamped-sheet images with hole and defect annotations for an automotive setting [8]. These examples are closer to metal-part inspection than generic image synthesis because surface response, view coverage, and part geometry are part of the dataset design.

Other studies diversify how synthetic evidence is produced and evaluated. Physics-informed thermography uses finite-element simulation to pre-train segmentation beyond ordinary RGB appearance [9]. Monnet et al. compare alternative synthetic-generation methods for scratches on metallic surfaces, including the effort required and the risk of a simulation-to-reality mismatch [10]. Peng et al. generate realistic anomalies through defect-map prediction and evaluate their contribution to downstream anomaly detection [11]. AnomalyDiffusion instead learns few-shot anomaly appearance and spatial conditioning, targeting both authenticity and image-mask alignment [12]. Together, these studies represent geometry/rendering, comparative generation, reconstruction-based anomaly synthesis, and diffusion-based generation. Their principal endpoints are dataset utility, anomaly realism, segmentation, or transfer. CAD2Replay uses synthetic preparation but asks a different question: whether feature identity, defect composition, tolerances, package identity, and verdict semantics remain traceable after conversion into the deployed mobile program and replay of a fixed RGB-D observation.

More generally, iPhone LiDAR point-cloud quality changes with surface material and acquisition conditions [13]. Apple-device indoor/outdoor mapping and an iPad Pro accuracy assessment show that scene structure, range, and scanning protocol influence recovered geometry [14,15]. A mobile inspection record must also preserve the camera-depth relationship: explicit LiDAR-camera calibration relies on valid direct 3D-2D correspondences rather than an RGB frame alone [16].

Reference-controlled studies clarify the accuracy boundary. Tamimi compared iPhone 13 Pro collection with surveying controls to assess relative accuracy rather than assuming nominal sensor performance [17]. Suleymanoglu et al. assessed iPhone 14 Pro road-infrastructure point clouds against RTK-GNSS and other mapping platforms [18]. Those works concern reconstruction or mapping, not part-level industrial verdicts, yet they demonstrate why device, pose, reference system, and acquisition state must accompany the observation. CAD2Replay adds frozen CAD-derived oracles, deployed-stage replay, and model lineage while keeping replay evidence separate from later physical observations.

Repeated inspection must distinguish consistency from correctness, as formalised by attribute-agreement analysis [19]. Multisensor reviews separate acquisition-, feature-, and decision-level fusion [20], while active-learning and confidence-calibration work shows that uncertainty management affects automated visual-inspection outcomes [21]. These approaches motivate repeated and uncertainty-aware decisions, but they do not create a replayable chain from CAD expectation to deployed verdict.

ML-system studies show that projects require tests beyond isolated model accuracy and face practical problems in test-data collection, execution, and result analysis [22, 23]. Data Cards formalise dataset origin, intended use, annotation, and evaluation context [24]. End-to-end provenance connects datasets, pipeline runs, code versions, parameters, metrics, and model artefacts [25]. These mechanisms can document inputs and executions, but generic ML testing does not define the industrial oracle itself. CAD2Replay therefore binds a CAD state to a fixed RGB-D record, runs that record through

deployed stages, and compares the resulting trace with a separate CAD-derived expectation. This connection distinguishes replay assurance from a synthetic-data benchmark, a mobile mapping assessment, or a model registry alone.

Four contrasts delimit the contribution. Model-only evaluation stops before conversion, preprocessing, association, measurement, temporal state, and verdict logic. Replay without a CAD oracle can reproduce execution but cannot determine whether the returned feature identity and composition are correct. Replay that bypasses deployed preprocessing cannot expose conversion, rotation, cropping, or decoding faults. Unpaired physical testing may describe two sessions but cannot preserve case-level transitions. Digital-twin testing research likewise treats oracle definition as a central cyber-physical-system problem [26], while the robotic-system testing literature distinguishes verification against specifications from validation against user needs and reports limited evidence of industrial applicability [27]. CAD2Replay combines these concerns by retaining an independent CAD-derived oracle, executing the deployed preprocessing path, and keeping physical identities paired, while still treating replay conformance and physical validity as different evidence levels.

3. Methodology

CAD2Replay separates three controlled units. Level 1 is a parameterised CAD configuration rendered into labelled training examples. Level 2 is a fixed RGB-D fixture replayed through the deployed software for alignment and regression protection. Level 3 is a CAD-grounded physical case acquired manually after development. Figure 1 separates offline preparation from online inspection. CAD and defect configuration drive rendering, annotation, training, and frozen CoreML packages. Online, RGB-D, pose, and confidence pass through Locator, Inspector, CAD association, measurement, stabilisation, and Judge. Recorded fixtures re-enter these deployed stages rather than a simplified surrogate.

For configuration $_i$, the synthetic record is:

$$s_i = \langle g_i, d_i, c_i, l_i \rangle,$$
$$x_i^{\text{synth}} = \langle \Omega(s_i), \Psi(s_i), m_i \rangle. \quad (1)$$

Here g_i is part/feature geometry in the part-local CAD frame (millimetres), d_i is the defect contract, c_i contains intrinsics, pose, orientation, and the 0.20–0.40 m camera range, and l_i contains material, illumination, and environment state. The render operator Ω returns RGB pixels; Ψ returns masks, classes, feature IDs, and geometric annotations. The immutable metadata m_i stores seeds, configuration identity, software/model identity, and content hashes. Equation (1) is the provenance root used by later fixture/oracle identities.

A defect is admitted only as the typed contract

$$d_i = \langle q_i, h_i, \theta_i, \delta_i, \Gamma_i \rangle. \quad (2)$$

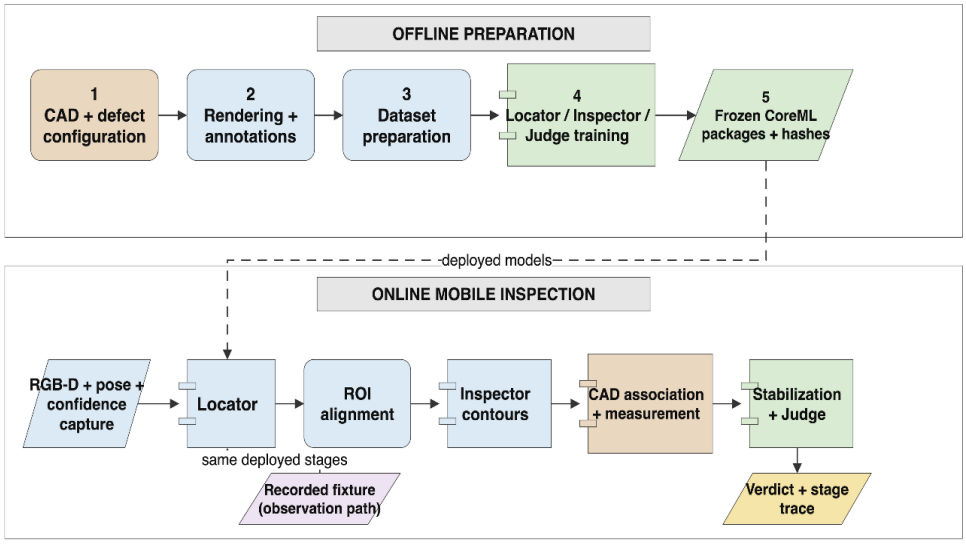


Figure 1: End-to-end production architecture. CAD-controlled offline preparation supplies frozen deployed packages; live and recorded observations enter the same mobile inspection stages

In (2), q_i is the manufacturing meaning, h_i is a host feature ID, surface, or part region, θ_i contains defect-specific parameters with units, δ_i is the imposed geometric deviation (millimetres, angle, or area fraction as appropriate), and Γ_i is a set of placement, overlap, coverage, and interaction predicates. The implemented corpus has 275 configurations across 11 families: 11 nominal, 22

misposition, 22 wrong-tool, 173 single mechanical-defect, and 47 combined-defect states. Five render variants yield 1,375 labelled image records. These counts establish generation coverage, not physical accuracy.

The preceding study used one YOLO11 segmentation over the complete image to segment features and support 2D measurement [3]. CAD2Replay retains CAD-driven synthetic/real preparation but separates scene-level and feature-level tasks. The Locator identifies the part and ROI in the full frame; the Inspector receives a rotated, scale-normalised crop and preserves more pixels for small holes, slots, and defect instances. This is an architectural response to mobile scale variation and local-detail requirements, not a direct claim that the cascade outperforms the earlier model.

Figure 2 turns (2) into a seven-gate extension workflow spanning meaning, parameters, CAD/annotation output, training lineage, runtime semantics, replay oracle, and physical evidence. Feedback may revise implementation or future data without changing an observed exact oracle.

A replay case separates observation O from oracle E:

$$R = \langle O, E \rangle, \quad O = \langle I, D, C, K, T_{WC}, H_{VI}, A \rangle, \quad E = \langle E_c, E_f \rangle. \tag{3}$$

$I \in \{0, \ldots, 255\}^{1440\times1920\times3}$ is RGB in row-major image order. $D \in \mathbb{R}^{192\times256}$ is row-major Float32 depth in metres; non-finite or non-positive samples are invalid. $C \in \{0,1,2\}^{192\times256}$ stores low, medium, and high confidence. $K \in \mathbb{R}^{3\times3}$ is the colour-camera intrinsic matrix in pixel units at the stored image resolution and is serialized column-major. $T_{WC} \in SE(3)$ is the right-handed ARKit camera-to-world transform with translation in metres. H_{VI} maps viewport coordinates to the normalized captured-image domain, and A stores optional anchor transform, last distance, UUID, and validity. E_c and E_f are frozen, hashed CAD-derived configuration and feature oracles; E_f uses the part-local CAD frame in millimetres. Figure 3 separates deployed RGB-D inputs from expected JSON used only for assertions and stage comparisons.

Figure 4 illustrates the multimodal observation retained for replay. The RGB image is stored at the colour-camera resolution, whereas the ARKit depth has a lower spatial resolution. During replay, the stored viewport transform and camera intrinsics preserve the mapping between these representations rather than estimating it again from the displayed images.

For viewport point $p_V = (x_V, y_V, 1)^T$ in stored orientation, the implementation first maps to normalized captured-image coordinates, then indexes depth, rescales to the colour image, unprojects using colour intrinsics, and applies the camera pose:

$$\begin{aligned} p_n &= H_{VI} p_V = (x_n, y_n, 1)^T, \\ (u_D, v_D) &= (\lfloor x_n W_D \rfloor, \lfloor y_n H_D \rfloor), \quad z = D[v_D, u_D], \\ & \quad C[v_D, u_D] \geq 2, \\ (u_I, v_I) &= (x_n W_I, y_n H_I), \\ P_C &= \left(z\frac{u_I - c_x}{f_x}, -z\frac{v_I - c_y}{f_y}, -z, 1 \right)^T, \\ \tilde{P}_W &= T_{WC} P_C, \qquad P_W = \tilde{P}_{W,1:3} / \tilde{P}_{W,4} \quad [\text{m}]. \end{aligned} \tag{4}$$

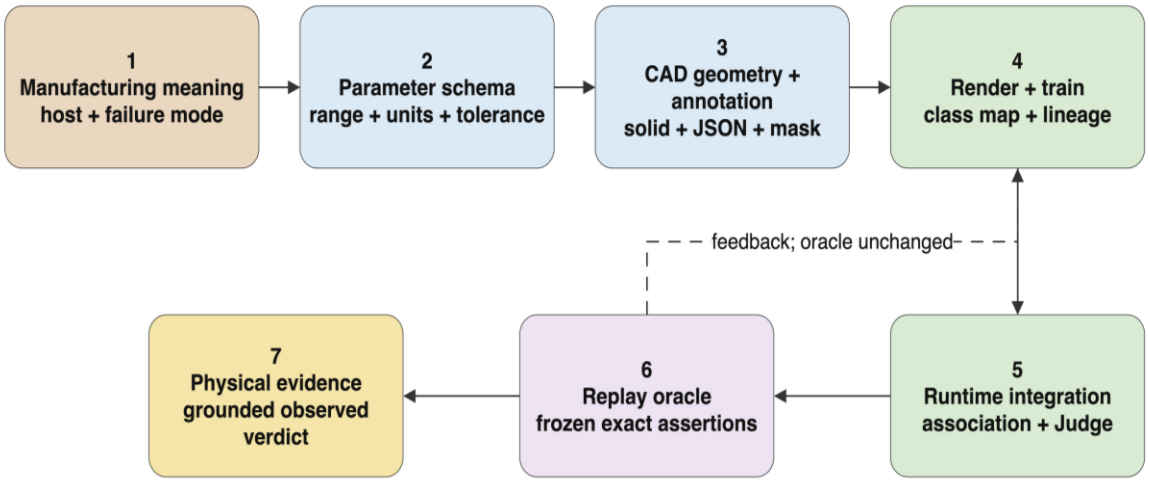


Figure 2: Seven-gate workflow for extending CAD2Replay with a feature or defect type. Dashed feedback can revise implementation or future evidence without relaxing the frozen oracle

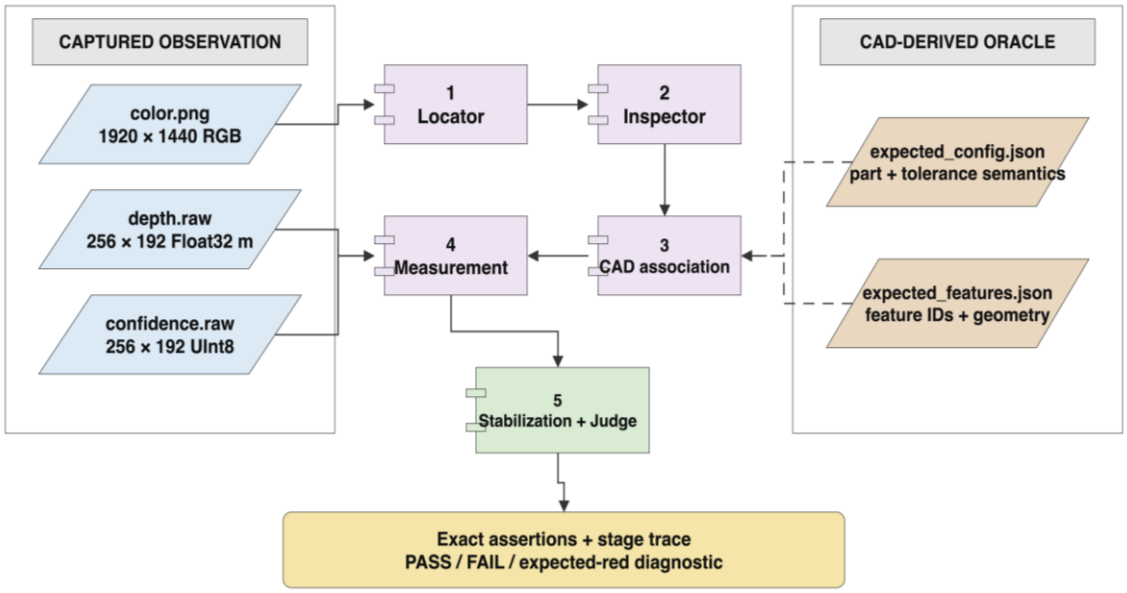
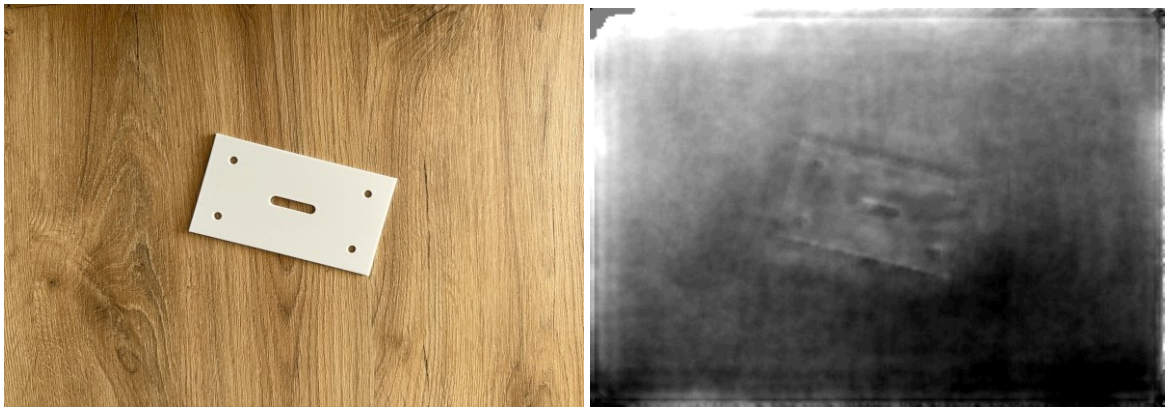


Figure 3: Replay observation and oracle flow. Captured RGB-D files and CAD-derived expectations enter through separate paths before exact assertions compare the deployed trace.



(a) (b)

Figure 4: Example of a stored mobile RGB-D observation used for deterministic replay: (a) RGB frame at the colour-camera resolution; (b) corresponding ARKit depth map, where intensity represents measured distance

$W_D = 256$ and $H_D = 192$ are depth-grid sample counts; $W_I = 1920$ and $H_I = 1440$ are colour dimensions in pixels. Indices are clamped to valid bounds. The pixel convention has u right and v down; ARKit camera coordinates are right-handed with x right, y up, and viewing direction $-z$. The signs in (4) therefore invert image y and depth-forward z. Orientation and viewport transform are captured rather than inferred during replay.

For an Inspector region M, the confidence-gated valid set and robust depth are:

$$\mathcal{Z}(M) = \{D[v,u]: (u,v) \in M,\ C[v,u] \geq 2,\ 0 < D[v,u] < \infty\},\quad \hat{z} = \text{median}\mathcal{Z}(M). \tag{5}$$

The implementation may use its surface estimator before the median fallback; the reported trace records valid fraction $|\mathcal{Z}|/|M|$. Equation (5) defines the confidence and finite-depth contract, not a claim that the resulting handset measurement is CMM-grade.

Let detections j and expected CAD features k have class, centre, size, and aspect descriptors. After class and feasibility gates, assignment is

$$X^* = \underset{X}{\text{argmin}} \sum_{j,k} c_{jk} X_{jk},\quad c_{jk} = w_p \frac{\| p_j - p_k \|_2}{L} + w_s \frac{|s_j - s_k|}{s_k} + w_a |a_j - a_k| w_c \mathbf{1}[t_{j \neq t_k}], \tag{6}$$

subject to one-to-one binary assignment constraints. p and s are in a common normalized/registered

frame, L is the part scale, a is aspect ratio, and t is feature class. Infeasible class or tolerance pairs receive prohibitive cost; the Hungarian solution supplies feature identity and match diagnostics. The weights and class gates are configuration inputs and belong to the hashed runtime state.

Registration is trusted only under the following rule:

$$\tau_R = \mathbf{1}[n_m \geq 3]\mathbf{1}[\text{finite}(r)]\mathbf{1}[r \leq 0.02\ \text{m}],$$
$$\varepsilon_{\text{eff}} = \varepsilon_{\text{CAD}} + \tau_R r. \quad (7)$$

n_m is the matched-feature count, r is registration RMS in metres, and ε_{CAD} is the configured tolerance in the same unit after conversion. When $\tau_R = 0$, an untrusted transform cannot manufacture a position failure. A CAD-based fallback is used only if its feature match is otherwise trusted; the trace records that fallback. Equation (7) separates registration uncertainty from nominal CAD tolerance.

For per-frame failure probability p_t, the smoothed value and persistence count are

$$q_t = \alpha p_t + (1-\alpha) q_{t-1},$$
$$h_t = \begin{cases} h_{t-1} + 1, & q_t \geq \tau_q, \\ 0, & q_t < \tau_q. \end{cases} \quad (8)$$

The replay protocol submits the same observation for five monotonically advancing 30 Hz-equivalent steps so the stored median/history and EMA/persistence paths execute. Five steps are a protocol condition, not proof of convergence under arbitrary live motion.

The verdict aggregation is:

$$V_t = \begin{cases} \text{WAIT}, & e_t < e_{\min}, \\ \text{FAIL}, & h_t \geq h_{\min} \wedge \left(\bigvee_k b_{k,t}\right), \\ \text{PASS}, & h_t \geq h_{\min} \wedge \neg\left(\bigvee_k b_{k,t}\right) \end{cases} \quad (9)$$

where e_t is the trusted-evidence state and $b_{k,t}$ are registered position/size, missing/extra feature, or persistent defect predicates. The exact reason remains in the stage trace. Equations (6)–(9) correspond to CAD matching, registration trust/fallback, Judge history, and feature aggregation rather than an abstract classifier alone.

Every end-to-end XCTest loads one fixture, constructs the deployed Locator, Inspector, CAD library, measurement, history, and Judge components, and asserts the expected prefix, exact feature/defect composition, verdict, and—for misposition cases—the positionOutOfTolerance reason. Empty-scene replay asserts the absence of part, verdict, and instances.

The 34-fixture corpus contains 12 nominal/reference, 11 edge/corner, 4 wrong-tool, 6 misposition, and 1 empty case. Each Inspector ran every fixture as a separate deployed XCTest evaluation; full-corpus exact pass count determines the ordering. No candidate passed the complete suite, so the ordering identifies only the best candidate observed so far.

4. Results and Discussion

Results are reported in six steps so completeness gates are not pooled with comparative model evidence. Step 1 audits the current CAD/configuration and asset corpus. Step 2 verifies rendering, dataset composition, and class coverage.

Step 3 uses the separate fixed real-evaluation corpus to select the Locator and test whether synthetic validation transfers to real images. Step 4 screens the four Inspectors and compares on fixed corpora. Step 5 follows the selected deployment lineage into exact replay and localises remaining failures. Step 6 reports the paired physical observations. The real-evaluation corpus in Step 3 is an offline transfer test; it is not either of the two physical acquisition sessions in the last Step.

The current evaluated corpus covers 11 part families with lengths of 60–140 mm, widths of 60–100 mm, and thicknesses of 2–5 mm. Its nominal library spans hole diameters of 3–55 mm and slots 8–40 mm long by 4–6 mm wide. Every feature has a stable ID, millimetre-valued geometry, a CAD-relative position, and a host part identity, so the same entity can be followed into annotations and runtime expectations.

Configuration generation starts from a nominal family record and applies a seeded, geometry-constrained transformation. A wrong-tool state selects a hole or slot and changes its diameter, width, or length while retaining the feature centre, respecting part boundaries, and preserving the slot constraint $l \geq w$; the evaluated absolute change is 4.00–7.84 mm. A misposition state translates one selected feature within the part boundary; evaluated resultant offsets are 6.35–13.88 mm.

Hole breakouts are attached to identified host holes and scale to the available rim, with notch radii of 1.05–5.98 mm. Slot bridges retain the source-slot ID and use tabs 3.6–5.4 mm wide and 4.0–9.97 mm long. Edge and corner primitives are generated along named boundaries with bounded depth, thickness penetration, roughness, cause, and contour geometry. The constructor then emits the altered solid, triangulated mesh, configuration record, and feature/annotation record from the same state.

This procedure produced 275 evaluated configurations: 11 nominal, 22 wrong-tool, 22 misposition, 173 single mechanical-defect, and 47 combined-defect states.

Five variants per configuration produced 1,375/1,375 matched image-label pairs. Each configuration was rendered once with each of five metal materials and five HDRI environments, with

pose and distance variation. Adding 41 real records produced 1,416 records for both task views. Table 1 reports the fixed split. Every one of the 11 Locator families and six Inspector classes occurs in every split; the test set is synthetic-only and therefore cannot support a mixed-domain test claim.

The correspondence between synthetic training images and manually annotated phone photographs was evaluated in the authors' earlier Synthetic-to-Real study [3] using one FeatureDetector and five-frame 2D MAE/MAPE comparisons. The present work does not reinterpret those measurements as evidence for the expanded architecture; instead, Step 3 performs a separate fixed real-image evaluation for Locator selection and Inspector count transfer.

Table 1. Compact dataset composition

Split	N	S/R, n	Coverage, L/I
Train	1,126	1,089/37	11/11; 6/6
Validation	147	143/4	11/11; 6/6
Test	143	143/0	11/11; 6/6
Total	1,416	1,375/41	11/11; 6/6

S/R denotes synthetic/real records; coverage denotes Locator families/Inspector classes.

Figure 5 provides a concrete example of the data representations linked by CAD2Replay for the eeb2 family. The nominal part geometry and feature classes is a CAD-rendered synthetic image.

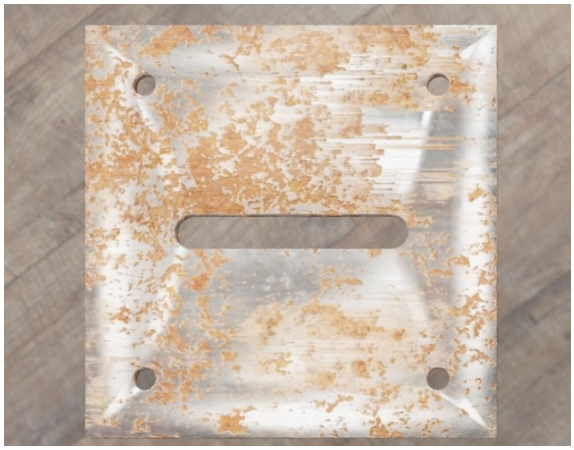


Figure 5: Example synthetic render for the eeb2 family.

The Locator model versions form a configuration sequence: v1 used YOLO11n-seg with strong mosaic and scale/translation augmentation, v2 retained YOLO11n-seg but added ±180° rotation and vertical flipping while reducing those augmentations, and v3 moved to YOLO11s-seg and disabled mosaic and rotation while retaining vertical flipping.

Synthetic validation did not distinguish the Locator model versions strongly: box mAP50–95 was 0.995 for v1, v2, and v3, while mask mAP50–95 was 0.995, 0.993, and 0.994. The fixed 35-sample real evaluation did. Locator accuracy progressed from 0.600 for v1 to 0.886 for v2 and 0.971 for v3; part-type accuracy on detected ground truth progressed from 0.588 to 0.882 and 0.971. Locator v3 was therefore selected from real-evaluation localization evidence and then held fixed for every Inspector comparison.

This disagreement answers the synthetic-to-real drift question at the version-selection level. Three essentially tied synthetic validation results mapped to real-image accuracies spanning 0.600–0.971, so synthetic validation alone would not have selected the observed real-domain leader. For v3, 34/35 samples were localized correctly; the family breakdown isolates the remaining signal to the U-channel group, with 3/4 correct. The result shows that v3 reduces the observed transfer gap on this fixed corpus, not that domain drift has been eliminated or that 35 samples estimate population performance.

Step 4 uses the four Inspector model versions as probes of whether CAD2Replay preserves disagreements between evidence levels. The Inspector model versions followed a parallel progression: v1 used the YOLO11s-scale segmentation architecture with strong mosaic augmentation, v2 retained that architecture while adding ±180° rotation and vertical flipping and reducing mosaic, v3 moved to the larger YOLO11m-scale architecture with mosaic and rotation disabled, and v4 retained the v3 architecture and no-mosaic policy while restoring ±180° rotation, increasing mask resolution through a mask ratio of 1 instead of 4, and reducing the batch size from 32 to 8. Table 2 screens every model version once: synthetic validation favours v1, fixed offline evaluation favours v2, and deployed replay favours v4. The purpose is not to declare a final Locator–Inspector pair, but to expose which conclusion each evaluation surface supports.

Table 2. Compact four-Inspector screen

Inspector	Mask mAP	Offline exact/MAE	Vector exact	Replay
v1	0.533	0.552/0.971	4/35	7/34
v2	0.463	0.657/0.681	6/35	16/34
v3	0.416	0.576/0.943	2/35	8/34
v4	0.447	0.648/0.710	6/35	23/34

The four model versions screen comprises 136 separately executed replay tests. The primary comparison retains v2, the offline leader, and v4, the best deployed candidate observed so far. Locator v3 and both corpora are fixed: the comparison contains 34 replay executions, without pooling their evidence units. Neither candidate reaches 34/34, so v4 remains a provisional deployed candidate rather than a final Inspector.

The real-evaluation breakdown identifies where the next improvement is required. For all 11 edge/corner samples, the fixed Locator localized the part, yet v4 achieved 0/11 full count-vector exact matches and a macro feature MAE of 1.424 count/sample. Across all 35 samples, the hole class contained 190 reference instances but only 130 predictions. Locator v3 is therefore adequate as the fixed upstream anchor for this bounded comparison, while the residual real-domain error lies in local feature and defect counting. The small corpus localises this development target but cannot quantify its production prevalence.

Step 5 tests whether the completed replay framework can expose and localise an incomplete deployed candidate without weakening the CAD oracle. It can: Inspector v4 passed the 23 non-edge fixtures but failed all 11 exact edge/corner compositions. Across the 11 fixtures, v4 returned 1/23 required corner instances, 15/33 edge instances, 22/28 hole-breakout instances, and 18/19 slot-bridge instances.

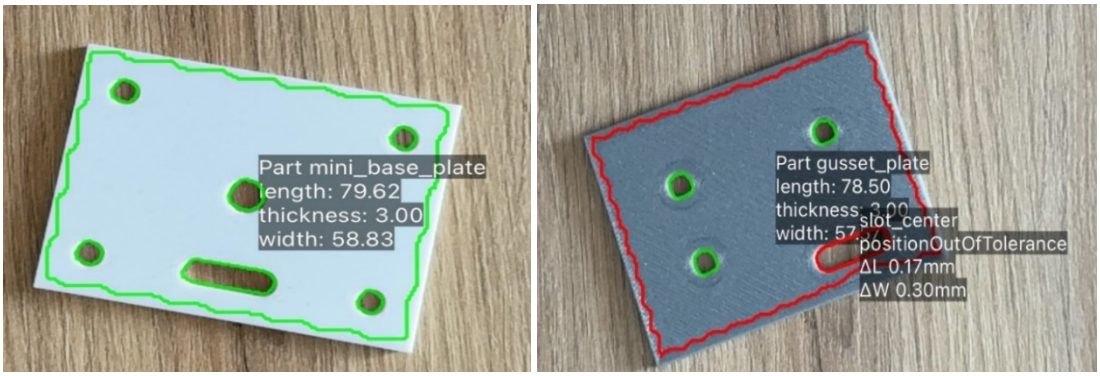
Corner under-counting occurred in every failed fixture: the sheet plate returned 1/3, and each of the other ten returned 0/2. Edge counts were exact only for mmp and vdct2; they were partial for bbp2 (1/3), eeb2 (2/3), gus (1/3), lbrk (2/3), sgp2 (1/3), and the sheet plate (2/3), and absent for bcp2, mflg, and the U-channel. Breakout deficits were concentrated in bbp2 (1/2), eeb2 (1/2), sgp2 (5/6), and the sheet plate (1/4). Bridge composition was nearly complete, with the only deficit in the U-channel (2/3). Replay therefore explains specifically why the candidate does not pass: a defective verdict can be produced from partial evidence while the required class-and-count composition remains incomplete.

The dimensional comparison explains why this residual is harder than counting ordinary hosts. Nominal holes span 3–55 mm in diameter and slots span 8–40 mm by 4–6 mm. In contrast, evaluated corner contours are only 0.33–4.75 mm across their shorter axis and 1.73–6.80 mm across their longer axis. Edge contours are elongated but thin, 1.20–5.96 mm across and 9.53–36.20 mm long. Breakout contours overlap a hole rim and span 1.04–11.24 mm by 1.76–11.96 mm, while bridge tabs occupy 3.5–5.3 mm by 4.05–9.97 mm—approximately the same transverse scale as the slot that contains them. Unlike a closed hole or slot, these primitives share a boundary with the host or part silhouette, occupy fewer pixels, and can merge with cropping, mask suppression, or one another. Exact multi-instance composition is therefore more demanding than detecting that the part is defective.

Those repairs improved the verification surface but did not manufacture missing model candidates, preserving the separation between framework correction and future Inspector improvement.

The physical block retains 31 identities: 12 nominal and 19 defective. Session 1 was correct on 26/31 (7/12 nominal; 19/19 defective), and Session 2 on 28/31 (9/12; 19/19).

Figure 6 shows representative outputs of the deployed mobile interface for an accepted physical case and a rejected misposition case. The visualization retains the predicted part identity, estimated dimensions, contour-level verdict, and explicit out-of-tolerance reason. These examples illustrate the form of the deployed output; the complete quantitative evidence is reported at case and frame level in Table 3.



(a) (b)

Figure 6: Representative deployed mobile inspection outputs on printed physical cases: (a) accepted part with a green contour; and (b) rejected part with a red contour and an explicit positionOutOfTolerance reason.

Table combines frame and case coverage by defect type. For wrong-tool and misposition cases, the numerator is the number of matched target-feature rows carrying the appropriate size- or position-out-of-tolerance flag.

For mechanical classes, it is the number of acquisition frames containing that explicit Inspector class in a case whose CAD metadata requires it. A case is positive when at least one applicable row or frame is positive.

Table 3. Raw-derived defect evidence grouped by type

Defect signal	Applicable cases	Session 1 frames; cases	Session 2 frames; cases
Wrong-tool size	4	96/140; 4/4	143/184; 4/4
Misposition	4	110/139; 4/4	112/171; 4/4
Hole breakout	1 1	256/362; 9/11	380/429; 11/11
Slot bridge	8	203/255; 7/8	291/302; 8/8
Edge defect	1 1	100/362; 5/11	260/429; 10/11
Corner defect	1 1	0/362; 0/11	0/429; 0/11

Frames within a case are correlated repeated observations, not independent part trials.

The case records show both strong response and improvement targets hidden by final verdicts. Wrong-tool size evidence reached 40/40 bcp2 rows in Session 1 and 42/45 vdct2 rows in Session 2, whereas mflg was the weakest in both sessions at 8/33 and 19/45. Misposition reached 30/30 for sgp2 in Session 1, while mmp decreased from 22/33 to 11/45. Breakout was present in all 29/29 gus frames in Session 1 and all 34/34 bcp2 frames in Session 2; the Session 1 zero cases bbp2 and eeb2 subsequently produced 30/34 and 11/29. Bridge similarly changed from 0/39 to 55/59 for the sheet plate. Edge evidence reached 40/44 for vdct2 and 29/29 for sgp2, but remained 0/34 for the U-channel in both sessions. Corner remained 0 in every physical case.

Across the 11 mechanical cases, at least one applicable explicit class appeared in 339/362 frames in Session 1 and 409/429 in Session 2. The class-specific Table 3 totals show why this combined signal cannot be interpreted as complete composition: breakout covered 9/11 then 11/11 cases, bridge 7/8 then 8/8, edge 5/11 then 10/11, and corner 0/11 in both sessions. Wrong-tool and misposition nevertheless produced target-specific evidence in all 4/4 corresponding identities in both sessions. The deployed candidate is therefore not generally blind to defective evidence; its remaining limitation is class- and case-dependent completeness and persistence.

The fixed real evaluation supplies the missing class-level corner observation: v4 emitted two corner predictions, both on images with positive corner ground truth. Combined with the session evidence, every supported defect semantic therefore produced a positive signal somewhere in the real-evidence chain. This does not establish instance-level recall. The raw app log records class counts per frame, but not the predicted mask or instance identifier needed to prove a one-to-one match between every output and every CAD defect primitive.

The central result is that the end-to-end verification framework is operational even though the evaluated Locator–Inspector pair is not final. Synthetic validation measures optimization on rendered labels. The fixed offline real evaluation is the explicit transfer-drift check: it separates Locator versions that synthetic validation leaves tied, establishes v3 as an adequate fixed anchor for this corpus, and then exposes residual Inspector count error that favours v2. Deployed replay adds CoreML conversion, ROI rotation/cropping, proposal decoding, post-processing, CAD association, RGB-D measurement, history, and Judge semantics; on that complete cascade v4 is the best candidate observed so far.

The disagreement is the framework's intended assurance output, not evidence that v4 is production-ready. v2 remains the offline leader, and no controlled experiment attributes v4's replay advantage to model size or augmentation. Replay preserves the 34/34 gate and turns v4's 11 edge/corner failures into class-level improvement targets: 1/23 corner and 15/33 edge instances indicate the largest gaps, while 22/28 breakouts and 18/19 bridges show that the cascade is not uniformly blind to mechanical defects. The framework also separated missing candidates from software defects that could be repaired without relaxing expected JSON. A future Inspector or cascade revision can therefore be compared on the identical fixtures and oracle.

The two physical sessions reinforce this development diagnosis. Mechanical-defect output occurred in 339/362 and 409/429 frames, and wrong-tool and misposition signals reached every corresponding identity. Breakout, bridge, and edge outputs also appeared repeatedly, but their case coverage changed between sessions and corner output remained absent. Together with the two ground-truth-positive corner predictions in fixed real evaluation, every supported semantic produced a real-data signal somewhere in the evidence chain. This supports the narrow conclusion that v4 is not class-blind; it does not imply reliable per-instance composition. Repeated frames are correlated observations of 31 selected cases, not hundreds of independent specimens.

Compared with the earlier Synthetic-to-Real study, CAD2Replay changes the assured object from one FeatureDetector and short 2D error sequences to the whole chain: typed CAD states, generated assets, two task-specific models, deployment lineage, deterministic replay, and paired observations. The contribution is the ability to identify where evidence is gained, transformed, or lost and to carry the resulting defect back into the next development

iteration. Results therefore do not establish other-part, device, site, safety, process-capability, gauge R&R, or CMM-grade generalisation.

5. Conclusions

CAD2Replay establishes a traceable verification chain from typed CAD configurations and labelled synthetic data to deployed RGB-D replay and paired physical observations. Completeness gates passed for 275 configurations and 1,375 labelled renders. On the fixed 35-image real corpus, Locator versions with nearly identical synthetic validation scores produced substantially different real-image results. Locator v3 achieved 0.971 accuracy and was therefore held fixed as the upstream model for the Inspector comparisons. The remaining observed transfer error was localised downstream to feature and defect counting.

Inspector selection remains unresolved: v2 performs best in fixed offline evaluation, whereas v4 performs best under complete deployed replay. Its 23/34 replay result and exact 1/23 corner, 15/33 edge, 22/28 breakout, and 18/19 bridge counts demonstrate that the framework detects and localises why the provisional candidate cannot yet pass. The two physical sessions independently show that v4 produces wrong-tool, misposition, breakout, bridge, and edge evidence, sometimes at high frame coverage, but not physical corner output. Although two corner predictions occurred in the fixed real-image evaluation, corner defects were not detected in either physical session. The available evidence therefore supports only class-level responsiveness under some conditions, not reliable instance-level detection. These results do not establish field reliability because the evaluation contains only 35 fixed real images, 34 replay fixtures, 31 selected physical cases, two acquisition sessions, and correlated frame-level observations. The main contribution is therefore the verification framework rather than the accuracy of the current Locator–Inspector pair. CAD2Replay preserves case identities, model lineage, exact CAD-derived oracles, and stage-level failure traces, allowing future model and software revisions to be evaluated against unchanged inputs and expectations. Future work should expand the number of physical specimens, devices, acquisition conditions, and operators; retain predicted masks and instance identifiers in physical logs; and evaluate measurement repeatability and reproducibility using reference metrology.

References

[1] Ma, Y., Yin, J., Huang, F., and Li, Q. Surface defect inspection of industrial products with object detection deep networks: a systematic review. Artificial Intelligence Review, 57, 333, 2024. https://doi.org/10.1007/s10462-024-10956-3.

[2] Bondar, D., Basova, Y., Vodka, O., and Machado, J. Mobile-focused spatial inspection of industrial parts using 2D image processing and LiDAR. Measurement, 266, 120522, 2026. https://doi.org/10.1016/j.measurement.2026.120522.

[3] Bondar, D., Basova, Y., and Vodka, O. Synthetic-to-real domain adaptation in computer vision systems: towards high-precision industrial applications. International Journal of Mechatronics and Applied Mechanics, I(21), 315–322, 2025. https://doi.org/10.17683/ijomam/issue21.29.

[4] Schraml, D., and Notni, G. Synthetic training data in AI-driven quality inspection: the significance of camera, lighting, and noise parameters. Sensors, 24(2), 649, 2024. https://doi.org/10.3390/s24020649.

[5] Schmedemann, O., Baaß, M., Schoepflin, D., and Schüppstuhl, T. Procedural synthetic training data generation for AI-based defect detection in industrial surface inspection. Procedia CIRP, 107, 1101–1106, 2022. https://doi.org/10.1016/j.procir.2022.05.115.

[6] Kim, A., Lee, K., Lee, S., Song, J., Kwon, S., and Chung, S.-W. Synthetic data and computer-vision-based automated quality inspection system for reused scaffolding. Applied Sciences, 12(19), 10097, 2022. https://doi.org/10.3390/app121910097.

[7] Fulir, J., Bosnar, L., Hagen, H., and Gospodnetić, P. Synthetic data for defect segmentation on complex metal surfaces. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 4424–4434, 2023. https://doi.org/10.1109/CVPRW59228.2023.00465.

[8] Antiles, S., and Talathi, S. S. Open Stamped Parts Dataset. 2024 IEEE 20th International Conference on Automation Science and Engineering, 1319–1324, 2024. https://doi.org/10.1109/CASE59546.2024.10711790.

[9] Pareek, K. A., May, D., Meszmer, P., Ras, M. A., and Wunderle, B. Synthetic data generation using finite element method to pre-train an image segmentation model for defect detection using infrared thermography. Journal of Intelligent Manufacturing, 36(3), 1879–1905, 2025. https://doi.org/10.1007/s10845-024-02326-1.

[10] Monnet, J., Petrovic, O., and Herfs, W. Investigating the generation of synthetic data for surface defect detection: a comparative analysis. Procedia CIRP, 130, 767–773, 2024. https://doi.org/10.1016/j.procir.2024.10.162.

[11] Peng, T., Zheng, Y., Zhao, L., and Zheng, E. Industrial product surface anomaly detection with realistic synthetic anomalies based on defect

map prediction. Sensors, 24(1), 264, 2024. https://doi.org/10.3390/s24010264.

[12] Hu, T., Zhang, J., Yi, R., Du, Y., Chen, X., Liu, L., Wang, Y., and Wang, C. AnomalyDiffusion: few-shot anomaly image generation with diffusion model. Proceedings of the AAAI Conference on Artificial Intelligence, 38(8), 8526–8534, 2024. https://doi.org/10.1609/aaai.v38i8.28696.

[13] Rizali, M., Idris, A., Rizali, M., & Syafuan, W. Quality assessment of 3D point clouds on the different surface materials generated from iPhone LiDAR sensor. International Journal of Geoinformatics, 18(4), 2022. https://doi.org/10.52939/ijg.v18i4.2259.

[14] Diaz-Vilarino, L., Tran, H., Frias, E., Balado, J., and Khoshelham, K. 3D mapping of indoor and outdoor environments using Apple smart devices. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B4-2022, 303–307, 2022. https://doi.org/10.5194/isprs-archives-xliii-b4-2022-303-2022.

[15] Teo, T.-A., and Yang, C.-C. Evaluating the accuracy and quality of an iPad Pro's built-in LiDAR for 3D indoor mapping. Developments in the Built Environment, 14, 100169, 2023. https://doi.org/10.1016/j.dibe.2023.100169.

[16] Yi, H., Liu, B., Zhao, B., and Liu, E. Extrinsic calibration for LiDAR-camera systems using direct 3D-2D correspondences. Remote Sensing, 14(23), 6082, 2022. https://doi.org/10.3390/rs14236082.

[17] Tamimi, R. Relative accuracy found within iPhone data collection. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2022, 303–308, 2022. https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-303-2022.

[18] Suleymanoglu, B., Tamimi, R., Yilmaz, Y., Soycan, M., and Toth, C. Road infrastructure mapping by using iPhone 14 Pro: an accuracy assessment. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-M-1-2023, 347–353, 2023. https://doi.org/10.5194/isprs-archives-XLVIII-M-1-2023-347-2023.

[19] Aust, J., and Pons, D. Assessment of aircraft engine blade inspection performance using attribute agreement analysis. Safety, 8(2), 23, 2022. https://doi.org/10.3390/safety8020023.

[20] Tsanousa, A., Bektsis, E., Kyriakopoulos, C., Gómez González, A., Leturiondo, U., Gialampoukidis, I., Karakostas, A., Vrochidis, S., and Kompatsiaris, I. A review of multisensor data fusion solutions in smart manufacturing: systems and trends. Sensors, 22(5), 1734, 2022. https://doi.org/10.3390/s22051734.

[21] Rozanec, J. M., Bizjak, L., Trajkova, E., Zajec, P., Keizer, J., Fortuna, B., and Mladenic, D. Active learning and novel model calibration measurements for automated visual inspection in manufacturing. Journal of Intelligent Manufacturing, 35, 1963–1984, 2024. https://doi.org/10.1007/s10845-023-02098-0.

[22] Openja, M., Khomh, F., Foundjem, A. T., Jiang, Z. M., Abidi, M., and Hassan, A. E. An empirical study of testing machine learning in the wild. ACM Transactions on Software Engineering and Methodology, 34(1), Article 7, 2024. https://doi.org/10.1145/3680463.

[23] Li, S., Guo, J., Lou, J.-G., Fan, M., Liu, T., and Zhang, D. Testing machine learning systems in industry: an empirical study. Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice, 263–272, 2022. https://doi.org/10.1145/3510457.3513036.

[24] Pushkarna, M., Zaldivar, A., and Kjartansson, O. Data Cards: purposeful and transparent dataset documentation for responsible AI. Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, 1776–1826, 2022. https://doi.org/10.1145/3531146.3533231.

[25] Schlegel, M., and Sattler, K.-U. Capturing end-to-end provenance for machine learning pipelines. Information Systems, 132, 102495, 2025. https://doi.org/10.1016/j.is.2024.102495.

[26] Somers, R. J., Douthwaite, J. A., Wagg, D. J., Walkinshaw, N., and Hierons, R. M. Digital-twin-based testing for cyber-physical systems: a systematic literature review. Information and Software Technology, 156, 107145, 2023. https://doi.org/10.1016/j.infsof.2022.107145.

[27] Araujo, H., Mousavi, M. R., and Varshosaz, M. Testing, validation, and verification of robotic and autonomous systems: a systematic review. *ACM Transactions on Software Engineering and Methodology*, 32(2), Article 51, 2023. https://doi.org/10.1145/3542945.