Bamboos Consulting
www.bamboos-consulting.com
info@bamboos-consulting.com

Industrial Computer Vision: Evaluation Protocol

TECHNICAL NOTE | 04 | v1.2 review | 1 November 2026

Document type: Technical evaluation protocol and illustrative scoring implementation.
Evidence status: Executable synthetic metric checks; no trained model or industrial image data evaluated.
Application boundary: Fixed-camera visual inspection, advisory defect classification, and human review.

1 Executive summary

Visual inspection projects are frequently described by a single accuracy figure. That measure can conceal the cost of missed defects, differences between product batches, and failures caused by illumination or camera changes. This note defines an evidence chain from data provenance to a release decision. It specifies class-sensitive metrics, batch-aware validation, shifted-condition tests, confidence reporting, and an executable synthetic scoring example. No production-readiness claim is made.

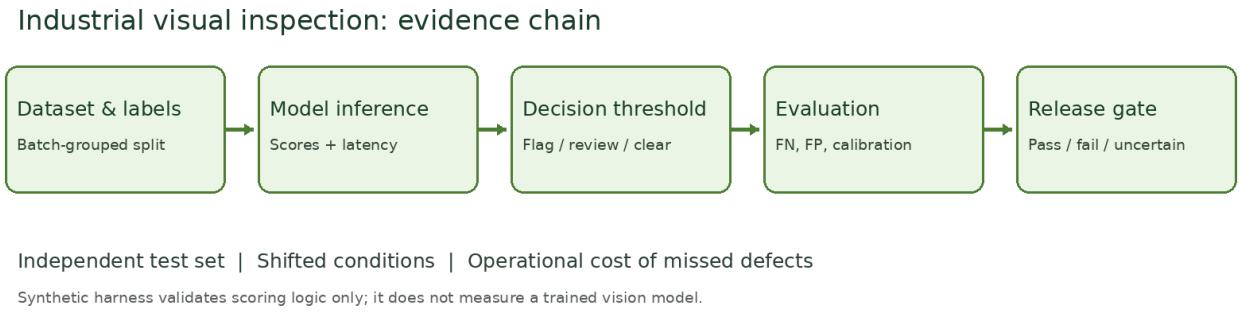


Figure 1. Proposed inspection evaluation workflow. The companion executable validates scoring logic, not visual inference.

2 Decision problem and operating assumptions

Consider a manufacturing line where a camera captures one image of each product and a model proposes *defect* or *no defect*. A missed defect can reach downstream processing; a false alarm sends an acceptable item to manual review. These errors have different costs. Define the actual label $y \in \{0, 1\}$ and the predicted label $\hat{y} \in \{0, 1\}$, where 1 denotes a defect. The threshold converting model scores to decisions must be declared before final testing. A low-confidence output may instead trigger review. The operational design must specify camera placement, cycle time, defect taxonomy, and review capacity.

3 Dataset design and leakage prevention

Record image identifier, acquisition timestamp, product family, manufacturing batch, camera, illumination regime, defect label, annotation procedure, and any adjudication. Partition data by batch, acquisition session, or physical item where relevant. Random image-level splitting may place near-duplicate views of the same item in training and test sets, leading to an optimistic estimate. Preserve a final held-out set, record its provenance, and do not repeatedly tune thresholds on it. Evaluate defect subclasses separately, especially rare but high-cost defects. Annotation disagreements should be retained and reported rather than silently resolved.

4 Evaluation metrics

Let TP be correctly flagged defects, FN missed defects, FP acceptable items incorrectly flagged, and TN correctly cleared items. Report

$$\text{Recall}_{\text{defect}} = \frac{TP}{TP+FN}, \qquad \text{FNR} = \frac{FN}{TP+FN}, \qquad \text{FPR} = \frac{FP}{FP+TN}.$$

Precision $TP/(TP+FP)$ depends on the frequency of defects in the evaluated sample; report prevalence and avoid extrapolating precision from artificially balanced data. Where probabilistic outputs are available, assess calibration and score distributions. For the system, measure end-to-end latency, dropped frames, review queue load, and trace completeness. Report uncertainty intervals using a sampling unit consistent with data dependence; frames from the same item or batch are not necessarily independent.

5 Experimental matrix

Condition	Required observation
Nominal acquisition	Class-wise confusion matrix, score calibration, throughput and timing.
Illumination shift	Performance by lighting regime and compared with nominal baseline.
Blur and occlusion	Performance by predefined severity and defect category.
New batch or camera	Generalization across batches/devices; document configuration differences.
Timeout and invalid frame	Explicit rejection/escalation; no silent default to acceptable product.
Low-confidence case	Review routing, operator workload and final disposition trace.

The test matrix should specify dataset counts per condition, model and preprocessing versions, thresholds, hardware, and execution environment. If the application uses OpenVINO, retain the model artifact digest and runtime/device configuration. Hardware benchmarking must use actual inference, not synthetic timing values.

6 Worked example: synthetic scoring harness

The companion standard-library Python script reads 48 deliberately constructed cases in a CSV file, validates their schema, computes TP/TN/FP/FN and rates, and stratifies counts by nominal versus shifted lighting. It is useful for testing the measurement pipeline and for teaching error asymmetry. The labels and predictions are invented for this illustration. The script always returns `INCONCLUSIVE_NO_REAL_MODEL_EVIDENCE`, even when all synthetic predictions are correct. This is intentional: a metric implementation cannot certify an unevaluated classifier.

7 Release decision and uncertainty

Define minimum acceptable defect recall, maximum false-positive review load, latency budget, and mandatory robustness conditions from the business process and its costs. A **fail** occurs when a predeclared critical criterion is violated; an **inconclusive** result occurs when sample coverage, model execution, trace completeness, or integration evidence is insufficient; a **conditional pass** requires all predeclared criteria to be met on an independent evaluation set. Zero observed misses does not imply zero miss probability. If n independent defective items are tested with zero misses, a rough one-sided 95% upper bound on the miss probability is $1 - 0.05^{1/n}$; dependence and sampling bias can invalidate this calculation. Report the number of independent defect-bearing items and the scope of the bound.

8 Relation to academic research and industry practice

The MVTec AD dataset [1] provides a recognized anomaly-detection benchmark, but its laboratory acquisition does not establish performance on a particular production line. Heckler-Kram et al. [2] introduce MVTec AD 2 with challenging conditions, useful when discussing external evaluation difficulty. MVTec AD 2 data are distributed under a non-commercial license; this publication does not reproduce the dataset, and a commercial proof of concept would require an appropriately licensed dataset. The NIST AI Risk Management Framework [3] motivates documented measurement and monitoring, not an industrial inspection certification. Intel OpenVINO documentation [4] specifies inference tooling; actual accuracy and timing depend on selected model and hardware. McKinsey's 2025 AI survey [5] and Deloitte's 2026 Manufacturing Industry Outlook [6] provide general adoption and manufacturing-investment context, respectively, but cannot validate the present classifier. These sources support distinct claims and should not be interpreted as endorsements.

9 Industrial relevance, limits, and next experiment

A bounded feasibility study can establish whether the defect taxonomy, image acquisition, model performance, and review capacity justify further integration. The next experiment should select a dataset with permissions covering the intended experimental and commercial use, run a fixed baseline model, compare nominal and shifted conditions, and publish a manifest, code version, confusion matrices, timing distributions, and limitations. The present release contains no trained model, camera measurements, or physical production-line results.

References

[1] P. Bergmann et al., "MVTec AD: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," *CVPR* (2019). doi:10.1109/CVPR.2019.00982.
[2] L. Heckler-Kram, J.-H. Neudeck, U. Scheler, R. Koenig and C. Steger, "The MVTec AD 2 Dataset: Advanced Scenarios for Unsupervised Anomaly Detection," *International Journal of Computer Vision* 134, article 175 (2026). DOI. Dataset license information.
[3] NIST, *AI Risk Management Framework 1.0* (2023). doi:10.6028/NIST.AI.100-1.
[4] Intel, *OpenVINO Documentation*. Official documentation.
[5] McKinsey & Company (2025), *The State of AI in 2025: Agents, Innovation, and Transformation*. Survey.
[6] Deloitte (2025), *2026 Manufacturing Industry Outlook*, 13 November 2025. Industry report.