Explainable AI for Decision Support in Quality Inspection and Defect Classification Systems

Ayesha D. Keane
Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA.
helloayesha@missouri.edu

Raffaellino Juvara
Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS, USA.
raffaellinojuvara@ku.edu

Abstract

Automated visual inspection increasingly relies on deep neural classifiers to identify surface defects, but high predictive accuracy alone does not provide the evidence that quality engineers need to accept, reject, rework, or escalate a part. The central problem addressed here is how explanation mechanisms can be coupled to cost-sensitive inspection decisions without overstating the reliability of post-hoc interpretability. This manuscript develops a methodological framework for explainable decision support in defect classification. The framework formulates inspection as a cost-sensitive decision problem, distinguishes predictive uncertainty from explanation uncertainty, and integrates local feature attribution, counterfactual explanation, and calibrated probability estimates into a human-in-the-loop workflow. A decision-theoretic formulation is used to show that explanation content should be evaluated against the action it is meant to support, not only against model fidelity. The analysis yields three defensible insights. First, attribution maps and counterfactuals answer different operator questions and should be combined selectively rather than presented as interchangeable evidence. Second, explanation stability and calibration are necessary boundary conditions for threshold-based acceptance rules because unstable explanations can induce inappropriate overrides. Third, decision-aligned evaluation requires joint assessment of predictive performance, explanation fidelity, action cost, and operator workload. The framework provides a principled basis for designing and auditing explainable inspection systems in manufacturing settings where defect costs are asymmetric and traceability is required.

Keywords

explainable artificial intelligence; quality inspection; defect classification; cost-sensitive decision making; counterfactual explanations; uncertainty calibration

1. Introduction

Quality inspection is a decision-intensive activity rather than a pure classification task. A defect classifier may assign a surface image to a scratch, pit, inclusion, or no-defect category, but the downstream action depends on defect severity, location, functional criticality, rework cost, and the risk of escaping a faulty part into assembly. Deep learning has improved the ability to detect subtle visual anomalies in industrial images, yet the most accurate models often provide little explicit evidence for their outputs. This opacity is not merely a technical

inconvenience. It affects whether an operator can distinguish a genuine defect from a lighting artifact, whether a process engineer can trace a recurrent classification error to a upstream cause, and whether an auditor can reconstruct why a part was accepted or rejected. The literature on explainable artificial intelligence has produced a wide range of attribution, example-based, and surrogate methods for making model behavior more intelligible [1], along with conceptual taxonomies that separate transparency, interpretability, and explanation [2]. The unresolved problem for quality inspection is how to convert those explanations into decision support that is technically valid, operationally useful, and robust to the cost asymmetries that characterize manufacturing quality control.

Visual defect detection and classification have been surveyed extensively, with classical image processing, feature engineering, and deep learning approaches applied across material types and inspection regimes [3]. Benchmark datasets and segmentation-based models have advanced the detection of localized surface anomalies [4], [5]. These developments make the predictive component increasingly capable, but they do not resolve the evidentiary gap between a model score and an inspection action. A heatmap that highlights a region of interest can be useful, but it does not by itself indicate whether the highlighted evidence is causally relevant, whether the model would behave similarly under slight changes in illumination, or whether the cost of rejecting a nondefective part exceeds the cost of accepting a defective one. Explainability therefore needs to be embedded in a decision architecture rather than appended as a visualization layer.

This paper makes a methodological contribution to that architecture. It formulates explainable inspection as a cost-sensitive decision problem in which a classifier produces a probability distribution over defect classes, an explanation module produces evidence about the model's input-response behavior, and a human-AI decision policy maps probabilities, uncertainty, explanations, and context into an inspection action. The framework is not a report of a deployed system or a benchmark experiment. It is an analytical and design contribution that specifies the variables, assumptions, explanation mechanisms, evaluation criteria, and failure modes that determine whether explainable decision support is credible in quality inspection. The central argument is that explanations should be selected and evaluated according to the decisions they are intended to support. Attribution is appropriate for localizing evidence, counterfactuals are appropriate for contrasting acceptable and unacceptable states, and calibrated uncertainty is appropriate for deciding when human escalation is warranted. Combining these elements without a decision-theoretic account can produce explanations that are visually plausible but operationally misleading.

The remainder of the paper develops this argument in six sections. Section 2 formulates the inspection decision problem and describes the system architecture. Section 3 analyzes explanation methods for defect classification, including local attribution, counterfactual explanation, and inherently interpretable models. Section 4 addresses calibrated uncertainty and explanation-conditional decision policies. Section 5 develops fidelity, stability, and human-centered evaluation criteria. Section 6 discusses design trade-offs, boundary conditions, and implications for inspection practice. Section 7 concludes.

2. Problem Formulation and Decision Architecture

2.1 Defect classification and inspection actions

Consider an inspection instance represented by an image or sensor patch $x \in X$. The defect state is a label $y \in \{0,1,...,K\}$, where $y = 0$ denotes a conforming part and $y = k$ for $k \geq$

1 denotes a defect class. A deep classifier fθ with parameters θ produces a predictive distribution over classes. For a standard softmax output, the probability assigned to class y is

$$p\theta(y|x) = \exp\frac{(z_y)}{\Sigma_k}\exp(z_k)$$

where z_y is the logit for class y and the sum runs over all classes. The classifier may be a convolutional network, a transformer, or an ensemble; the formulation is architecture-agnostic. The important assumption is that pθ(y|x) is intended to be a probability, not merely a ranking score. If the model is poorly calibrated, the numerical values may not correspond to empirical frequencies, a point developed in Section 4.

A defect class alone does not determine the inspection action. Let A = {accept, reject, rework, escalate} denote the available actions. For each action a ∈ A and true state y, let C(a,y) denote the cost of taking action a when the true state is y. Costs can include scrap value, rework labor, production delay, safety exposure, and reputational risk. The expected cost of action a given instance x is

$$EC(a|x) = \Sigma_y p\theta(y|x)C(a,y)$$

and the Bayes-optimal action under the model's probabilities is

$$a* = argmin_a\, EC(a|x)$$

This decision rule differs from the conventional argmax class rule. Argmax minimizes expected misclassification when all errors have equal cost, but quality inspection typically has strongly asymmetric costs. Accepting a defective part that would cause a field failure may be far more costly than rejecting a conforming part and sending it to unnecessary review. The expected-cost formulation makes that asymmetry explicit.

For a binary distinction between conforming and defective states with actions accept (A) and reject (R), the threshold structure is transparent. Let y = 1 denote defective and y = 0 denote conforming. Let p = pθ(y = 1|x). The expected cost of rejection is p C(R,1) + (1 − p) C(R,0), and the expected cost of acceptance is p C(A,1) + (1 − p) C(A,0). Rejection is preferred when its expected cost is lower. Solving the inequality yields the threshold

$$\tau = \frac{(C_A0 - C_R0)}{(C_A0 - C_R0 + C_R1 - C_A1)}$$

where C_A0 and C_A1 are the costs of accepting a conforming and defective part, and C_R0 and C_R1 are the costs of rejecting a conforming and defective part. Under the usual normalization in which accepting a conforming part and rejecting a defective part have zero cost, τ reduces to the familiar ratio of false-rejection cost to the sum of false-rejection and escape costs. The practical implication is that a single probability cutoff is justified only when the cost matrix and the calibration of pθ are stable. If defect criticality varies by part or location, the threshold should be conditioned on that context rather than fixed globally.

2.2 Explanation module and inspection workflow

The decision architecture contains four functional components: sensing and preprocessing, defect classification, explanation generation, and human-AI action selection. The explanation module receives the instance x, the model fθ, and the predicted class c, and returns an explanation e = E(x, fθ, c). The explanation can be a feature-attribution map over image regions, a set of counterfactual examples, a rule-based justification, or a combination. The choice of explanation type should be determined by the operator's task. A line inspector who

needs to verify whether a mark is a scratch benefits from localized attribution and a contrastive statement about why the mark is classified as defective rather than as a lighting artifact. A process engineer who needs to reduce false rejections benefits from aggregated feature attributions across flagged batches and from counterfactuals that indicate which measurable defect properties would change the decision.

The workflow should not present the explanation as a substitute for the model probability. It should present the probability, the calibrated uncertainty, the recommended action under the cost model, and the explanation in a coordinated interface. The operator can then accept the recommendation, override it with a recorded reason, or escalate to a higher-tier review. Logging the explanation, the model score, the operator action, and the final disposition is essential for audit and for detecting systematic misalignment between model evidence and human judgment. In real-time inspection, explanation generation must fit within the line cycle. That constraint favors methods with predictable computational cost, such as region-level perturbation with a fixed number of samples, gradient-based attribution, or a small set of precomputed counterfactual templates. Expensive model-agnostic searches may be appropriate for offline root-cause analysis but not for inline decisions.

3. Explainability Methods for Defect Classification

3.1 Local feature attribution

Local attribution methods explain an individual prediction by assigning importance to input features. For image inspection, features are usually defined as superpixels, patches, or segmented regions rather than raw pixels, because region-level attribution is more stable and more intelligible to operators. LIME approximates a complex model locally with an interpretable surrogate around the instance of interest [6]. SHAP assigns importance values derived from cooperative game theory and has an additive property that supports local accuracy [7]. For a set of features N and a coalition S of features, the Shapley value for feature j is

$$\varphi_j \;=\; \Sigma_\{S \subseteq N, j \notin S\}\, |S|!\,(|N| - |S| - 1)!/|N|!\;\; [v(S \cup \{j\}) - v(S)]$$

where v(S) is the model output when only the features in S are present, and the sum is over all subsets of features that exclude j. The value φ_j represents the average marginal contribution of feature j to the model output across all possible feature coalitions. The local accuracy property states that the model output at the instance is the sum of the base value and the feature attributions. This property is valuable for inspection because it ties the explanation to the actual model prediction rather than to a separate surrogate.

Attribution maps are useful for answering a specific question: which regions of the image most influenced the model's score? In defect classification, that question is often the operator's first diagnostic step. If the attribution is concentrated on a scratch-like region, the operator can compare the region with the defect specification. If the attribution is dominated by the image border, a shadow, or a background texture, the operator has reason to suspect a spurious correlation. However, attribution should not be treated as a causal explanation. Correlated features, such as a scratch and its associated shadow, may share importance, and the Shapley decomposition may distribute credit in a way that does not match the physical defect mechanism. Attribution is also sensitive to the feature partition; changing the superpixel size can change the map. These limitations do not make attribution useless, but they mean that attribution should be presented as evidence about model behavior, not as a direct statement about the manufacturing process.

3.2 Counterfactual and contrastive explanations

Counterfactual explanations answer a different question: what minimal change to the input would alter the model's decision? For an instance x predicted as class c, a counterfactual seeks a feasible alternative x′ classified as a different class c′. A standard formulation is

$$x' = argmin_\{z \in F\}\, d(x, z)\ such\ that\ f\theta(z) = c'$$

where d(·,·) is a distance metric, F is the set of feasible inputs, and c′ is the target class. The feasibility constraint is critical in inspection. A counterfactual that moves a defect to a different location or introduces an impossible surface texture is not actionable. For a scratch classified as severe, an actionable counterfactual might describe a shallower scratch of the same length or a shorter scratch of the same depth, depending on the defect specification. For a pit classified as a porosity defect, the counterfactual might identify the depth threshold at which the classifier would switch to a conforming label. Such explanations support rework decisions and process adjustment because they identify the boundary of the model's decision region in physically meaningful terms.

Counterfactual explanations are contrastive rather than merely descriptive. They tell the operator why the current instance is not classified as the alternative class. This is often more useful in inspection than a full feature attribution because quality decisions are frequently framed as comparisons: is this mark a defect or an acceptable cosmetic variation? Is this pit within tolerance or outside it? The main risks are computational and semantic. The optimization in the counterfactual formulation can be nonconvex, and the distance metric encodes a judgment about which changes are small and which are large. If the metric is not aligned with the defect specification, the explanation may be technically valid but operationally irrelevant. Counterfactual explanations should therefore be constrained by domain knowledge, such as allowed repair actions, material continuity, and tolerance limits.

3.3 Inherently interpretable models and hybrid decision layers

Post-hoc explanation is not the only route to interpretable decision support. For high-stakes or regulated inspection, an inherently interpretable model may be preferable when it can achieve adequate accuracy. Rule-based classifiers, generalized additive models, and sparse decision lists can provide transparent decision boundaries without a separate explanation step [9], [10]. In quality inspection, an interpretable decision layer can be placed on top of a deep feature extractor: the deep model detects and segments candidate defect regions, and a rule-based or additive layer combines measurable features such as defect area, contrast, elongation, and location relative to a critical surface. The decision layer can be audited directly, while the deep model remains responsible for the perceptual task of finding anomalies.

The trade-off is between flexibility and transparency. A fully interpretable model may fail to capture complex texture patterns or rare defect morphologies that a deep network can learn. A hybrid architecture can preserve perceptual accuracy while making the final action rule transparent, but it introduces a new dependency: the interpretable layer is only as good as the features it receives. If the segmentation model misses a defect, the rule layer will make a transparent but wrong decision. The appropriate choice depends on the cost of errors, the availability of labeled defect examples, and the regulatory requirement for traceability. The framework does not assume that post-hoc explanation is always superior to interpretable modeling. It treats both as design alternatives that should be evaluated against the decision task.

4. Decision Support Under Uncertainty and Cost Asymmetry

4.1 Calibration and uncertainty

Cost-sensitive decision rules require probabilities that are meaningfully calibrated. Modern neural networks can be overconfident, and the softmax score is not guaranteed to equal the empirical frequency of correctness [11]. Calibration can be assessed with the expected calibration error over probability bins:

$$ECE = \Sigma_\{m = 1\}\^M (|B_m|/n) |acc(B_m) - conf(B_m)|$$

where B_m is the set of instances in bin m, n is the number of instances, acc(B_m) is the observed accuracy in the bin, and conf(B_m) is the average predicted confidence. A low ECE indicates that predicted probabilities align with observed frequencies, but it does not guarantee that the model is well calibrated for every defect class or every operating condition. Calibration should be checked by defect type, material, and lighting condition because a model may be calibrated on average while being systematically overconfident for a rare and costly defect.

Uncertainty has two relevant components in inspection. Aleatoric uncertainty reflects ambiguity in the input, such as a faint mark that is difficult to distinguish from a surface texture. Epistemic uncertainty reflects model ignorance, such as a defect morphology underrepresented in training data. Bayesian approximation methods can estimate predictive uncertainty, but their computational cost may be incompatible with high-throughput inspection [12]. A practical alternative is to use a combination of calibrated confidence, ensemble disagreement, and out-of-distribution detection. The decision policy can then use uncertainty as an escalation signal. If the calibrated probability is near the cost-based threshold or if ensemble disagreement is high, the instance is routed to human review rather than automatically accepted or rejected. This use of uncertainty is decision-relevant because it identifies the cases in which the expected-cost calculation is most sensitive to probability error.

4.2 Explanation-conditional decision policies

A human-AI decision policy maps the model probability, uncertainty, explanation, and context into an action. Let g(pθ, u, e, s) → a denote that policy, where u is an uncertainty summary and s is context such as part identifier, operator role, and station. The explanation does not replace the probability in the cost calculation. Instead, it supports the operator's assessment of whether the model's evidence is admissible. For example, if the model predicts a defect with high probability but the attribution is dominated by a known reflective artifact, the operator may override the automatic reject and request re-imaging. If the model predicts a conforming part with moderate probability but a counterfactual indicates that a small increase in defect depth would cross the rejection threshold, the operator may choose additional measurement rather than immediate acceptance.

The policy should be designed to avoid two failure modes. The first is overreliance, in which the operator accepts the model recommendation without inspecting the evidence because the explanation is visually persuasive. The second is underreliance, in which the operator ignores useful model output because an earlier explanation was unstable or incomprehensible. Both failure modes are failures of trust calibration. The interface should therefore present explanation confidence alongside explanation content. Attribution maps can be accompanied by stability estimates, and counterfactuals can be accompanied by feasibility

constraints. When explanation stability is low, the policy should treat the explanation as diagnostic rather than decisive, and the default action should favor escalation for high-cost defect classes.

5. Explanation Fidelity, Stability, and Evaluation

5.1 Fidelity to model behavior

An explanation is faithful if it reflects the model's actual input-response behavior. Faithfulness is not the same as plausibility. A heatmap can look convincing to a human while failing a perturbation test in which removing the highlighted region does not reduce the model's prediction. Evaluation should therefore include objective tests. For attribution methods, one can measure the change in model output when high-attribution regions are masked or when low-attribution regions are retained. For counterfactual methods, one can verify that the proposed alternative is classified as intended and that it satisfies the feasibility constraints. Visualization evaluation in neural networks has emphasized that explanation quality must be assessed with respect to the model, not only with respect to human aesthetic judgment [13].

Fidelity is necessary but not sufficient for decision support. A faithful explanation of a spurious model will reveal that the model relies on an irrelevant artifact, but it will not make the model suitable for inspection. In that case, the explanation is valuable for model debugging and for rejecting the model, not for persuading an operator to accept its decisions. The evaluation should distinguish explanatory fidelity from model validity. A high-fidelity explanation of a bad model is a diagnostic tool, not a justification for deployment.

5.2 Stability and robustness

Explanation stability refers to the consistency of an explanation under small, decision-irrelevant perturbations of the input. If a slight change in illumination, compression, or sensor noise produces a radically different attribution map, the operator cannot rely on the map to identify the defect evidence. Let E(x) denote the explanation for input x and let δ be a bounded perturbation. A stability measure can be written as

$$S(E) \;=\; E_\{x \sim D\}\,[D_E(E(x), E(x+\delta))]$$

where D_E is a distance between explanations and the expectation is over the inspection distribution. The definition of D_E depends on the explanation type. For attribution maps, D_E may be a rank correlation or a top-k overlap; for counterfactuals, it may be the distance between the proposed alternatives or the change in the recommended action. Stability should be evaluated on perturbations that are realistic for the inspection environment, such as lighting variation, camera noise, and minor part positioning changes.

There is a tension between stability and sensitivity. An explanation that is perfectly stable may be insensitive to genuine defect changes, while an explanation that is highly sensitive may be unstable under noise. The appropriate balance depends on the decision. For accept/reject decisions with high escape costs, stability is especially important because unstable evidence can cause inconsistent dispositions across identical parts. For root-cause analysis, sensitivity to subtle defect features may be more valuable, and instability can be managed by aggregating explanations across a batch.

5.3 Human-centered evaluation

Human-centered evaluation asks whether the explanation supports the operator's task without increasing cognitive burden beyond the available cycle time. This is not reducible to a single metric. Relevant dimensions include comprehension, actionability, trust calibration, and workload. Comprehension concerns whether the operator can correctly interpret the explanation in terms of the defect specification. Actionability concerns whether the explanation suggests a feasible next step, such as re-imaging, measurement, rework, or escalation. Trust calibration concerns whether the operator's reliance on the model matches the model's actual reliability. Workload concerns whether the explanation can be processed within the inspection cycle.

The distinction between explainability and causability is useful here. Explainability refers to the properties of the model and explanation method; causability refers to the extent to which a human can understand and use the explanation in a given domain [15]. A system can be explainable in a technical sense while having low causability for inspectors because the explanation uses features that do not align with their defect ontology. Human-centered evaluation should therefore involve domain experts and should test whether explanations improve decision quality under realistic time pressure, not merely whether they are rated as useful in a static survey.

5.4 Decision-aligned evaluation

Decision-aligned evaluation connects explanations to inspection outcomes. The relevant question is not only whether an explanation is faithful or stable, but whether it improves the expected cost of the human-AI team. A complete evaluation would compare the team's decisions with and without explanations at the same level of automation, using the same cost matrix and the same defect population. It would also measure whether explanations reduce the rate of high-cost errors, such as escaped defects, without increasing false rejections beyond an acceptable limit. This requires logging the explanation, the operator's interpretation, the action, and the final ground truth.

Real-time integration of recognition and decision making is a practical constraint that affects evaluation. In production settings, defect recognition and optimized decision making must be coupled closely enough to support line-speed actions; work on structural timber jointing has demonstrated a real-time workflow in which defect recognition informs manufacturing decisions [16]. That type of integration highlights a general requirement: explanation generation cannot be treated as an offline add-on if the decision is made inline. The explanation must be available at the point of decision or the decision policy must be designed to proceed without it. Explanations in artificial intelligence are also selective and contrastive in human communication, meaning that the most useful explanation is often the one that answers a specific why-not question rather than the one that describes every feature [17]. Decision-aligned evaluation should therefore specify the question the explanation is intended to answer and test whether the operator can answer it correctly and quickly.

6. Technical Discussion and Implications

6.1 Explanation as decision evidence rather than model decoration

The framework developed here treats explanations as evidence in a decision process. That framing has several consequences. First, explanation type should be matched to the operator's question. Attribution is suited to localization and spurious-correlation detection. Counterfactuals are suited to boundary cases and rework decisions. Rule-based justifications are suited to audit and regulatory review. Presenting all three simultaneously can increase

cognitive load without improving decisions. Second, explanation quality should be evaluated against the action it supports. A faithful attribution that does not change or inform an action has low decision value, while a less detailed explanation that reliably routes a high-cost defect to escalation can have high decision value. Third, explanation should be accompanied by a statement of its limitations. An attribution map does not prove that a feature is causal, and a counterfactual does not prove that the alternative is physically achievable. The interface should communicate those limits without overwhelming the operator.

The distinction between interpretability and explanation is relevant to inspection practice. Interpretability is a property of a model or representation, whereas explanation is an activity that answers a question for a recipient [18]. An inspection system can provide a model-level interpretable decision rule, a local post-hoc explanation, or both. The choice depends on the recipient. A quality engineer may need model-level rules for process control, while a line operator may need a local contrastive explanation for a single part. A regulatory auditor may need a traceable decision record that combines the model score, the calibrated probability, the cost-based recommendation, the explanation, and the override reason.

6.2 Design principles and trade-offs

Several design principles follow from the analysis. First, calibrate before thresholding. Cost-based thresholds assume that probabilities are meaningful. If calibration is poor, the threshold should be adjusted using a held-out calibration set or the decision should be escalated. Second, separate detection from disposition. A deep model can be used for perceptual detection and segmentation, while an interpretable or rule-based layer handles disposition. This separation allows the explanation of the final action to be auditable even when the perceptual model is opaque. Third, use uncertainty as a routing signal. Instances near the decision threshold or with high epistemic uncertainty should be routed to human review rather than forced into an automatic decision. Fourth, evaluate explanations for stability under realistic perturbations. An explanation that changes dramatically under minor lighting variation is not reliable evidence. Fifth, log explanations and overrides. The log supports post-hoc analysis of whether the explanation was useful and whether the operator's overrides were justified.

The trade-offs are not eliminable. More faithful explanations may be less comprehensible; more stable explanations may be less sensitive to subtle defects; more interpretable models may have lower accuracy on complex textures. The appropriate balance is determined by the cost matrix and the regulatory context. In a high-volume cosmetic inspection line with low escape cost, a fast attribution map and a simple threshold may be sufficient. In a safety-critical inspection line with high escape cost, a conservative escalation policy and an interpretable decision layer may be justified even at the cost of additional false rejections. The framework does not prescribe a single architecture. It specifies the variables and evaluations that should inform the choice.

6.3 Boundary conditions and failure modes

The framework relies on several boundary conditions. The cost matrix must be reasonably stable and known. If costs vary by part, customer, or production regime, the decision threshold should be conditioned on those variables. The model probabilities must be calibrated for the relevant defect classes and operating conditions. If the inspection population shifts, calibration should be rechecked. The explanation method must be computationally

compatible with the line cycle. If it is not, the decision policy should either use a faster explanation or proceed without inline explanation and rely on offline audit.

Failure modes include spurious attribution, unstable counterfactuals, and miscalibrated confidence. Spurious attribution occurs when the model relies on a correlate of the defect rather than the defect itself; the explanation can reveal this, but it cannot correct it. Unstable counterfactuals occur when the optimization finds alternatives that are not physically feasible; feasibility constraints and domain review are necessary. Miscalibrated confidence occurs when the model's probabilities are systematically too high or too low; cost-based thresholds then produce incorrect actions. These failure modes are not independent. A miscalibrated model can produce unstable explanations because small probability changes near the threshold can change the explanation. The evaluation should therefore test the system as a whole, not each component in isolation.

7. Conclusion

This paper developed a methodological framework for explainable AI in quality inspection and defect classification, with decision support as the organizing objective. The framework formulates inspection as a cost-sensitive decision problem, distinguishes the roles of local attribution, counterfactual explanation, and inherently interpretable decision layers, and integrates calibrated uncertainty into human-AI action selection. The analysis shows that explanations are not interchangeable visual aids. Attribution answers localization questions, counterfactuals answer boundary and rework questions, and interpretable decision layers answer audit and process-control questions. Their value depends on whether they improve the expected cost of the human-AI team under realistic operating constraints.

The main implication for practice is that explainable inspection systems should be designed around the decision, not around the model alone. Calibration, stability, feasibility, and operator workload are not secondary concerns; they determine whether an explanation can be trusted as evidence. The framework also clarifies why decision-aligned evaluation is necessary. A system that produces faithful explanations but does not reduce high-cost errors, or that reduces errors at the expense of unmanageable false rejections, has not demonstrated decision support. By making the decision variables, explanation mechanisms, and evaluation criteria explicit, the framework provides a basis for designing explainable inspection systems that are technically defensible, auditable, and operationally useful in manufacturing quality control.

References

1. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access, 6, 52138–52160.

2. Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115.

3. Czimmermann, T., Ciuti, G., Milazzo, M., Chiurazzi, M., Roccella, S., Oddo, C. M., & Dario, P. (2020). Visual-based defect detection and classification approaches for industrial applications—A survey. Sensors, 20(5), 1459.

4. Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019). MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection. In Proceedings of

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 9592–9600).

5. Tabernik, D., Šela, S., Skvarč, J., & Skočaj, D. (2020). Segmentation-based deep-learning approach for surface-defect detection. Journal of Intelligent Manufacturing, 31(3), 759–776.

6. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144).

7. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems 30 (pp. 4765–4774).

8. Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Harvard Journal of Law & Technology, 31(2), 841–887.

9. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206–215.

10. Letham, B., Rudin, C., McCormick, T. H., & Madigan, D. (2015). Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. Annals of Applied Statistics, 9(3), 1350–1371.

11. Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning (pp. 1321–1330).

12. Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In Advances in Neural Information Processing Systems 30 (pp. 5574–5584).

13. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., & Müller, K.-R. (2017). Evaluating the visualization of what a deep neural network has learned. IEEE Transactions on Neural Networks and Learning Systems, 28(11), 2660–2673.

14. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

15. Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9(4), e1312.

16. Luo, D., Gattas, J. M., & Tan, P. S. S. (2020, July). Real-time defect recognition and optimized decision making for structural timber jointing. In The International Conference on Computational Design and Robotic Fabrication (pp. 36-45). Singapore: Springer Singapore.

17. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267, 1–38.

18. Lipton, Z. C. (2018). The mythos of model interpretability. Communications of the ACM, 61(10), 36–43.

19. Molnar, C. (2022). Interpretable machine learning (2nd ed.). Independently published.

20. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., & Yang, G.-Z. (2019). XAI—Explainable artificial intelligence. Science Robotics, 4(37), eaay7120.