

Real-Time Vision And Digital Twins For Defect Detection And Process Optimization

Xiaoling Li^{1,2} and Kai Li^{3*}

¹ College of Big Data and Artificial Intelligence, Zhengzhou University of Economics and Business, Zhengzhou, 451191, China

² Henan Province Engineering Research Center of Multimodal Perception and Intelligent Interaction Technology, Zhengzhou, 451191, China

³ Zhengzhou Technical College, Zhengzhou, 450100, China

* Corresponding author. E-mail: lk20210533@126.com

Received: Aug. 22, 2026; Accepted: Sep. 08, 2026

This study presents an online closed-loop quality-control method that integrates real-time computer vision with a digital twin. Rather than treating inspection and control as separate tasks, a lightweight multiscale detector converts bounding boxes into timestamped quality states used for next-cycle risk prediction and constrained parameter search. On the NEU test set, the detector reaches 94.1% mAP@0.5 and 63.8% mAP@0.5:0.95 at 78.6 FPS. The online-corrected twin predictor obtains an RMSE of 0.018 and R^2 of 0.923. Under paired simulated disturbances, the complete loop reduces average defect rate from 7.82% to 2.94%, raises first-pass yield to 96.9%, and runs with an average latency of 34.8 ms. Representative images support the reported localization behavior for small, low-contrast defects. These results establish controlled-environment feasibility, not plant-wide generalization.

Keywords: Multiscale feature fusion; quality prediction; physical-virtual synchronization; online decision-making; closed-loop control

© The Author(s). This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are cited.

http://dx.doi.org/10.6180/jase.202701_36.004

1. Introduction

Smart manufacturing is moving beyond isolated machine automation toward production systems that can perceive conditions, interpret them, and act autonomously. In continuous and quasi-continuous processes such as rolling, stamping, coating, injection molding, and precision assembly, surface defects often arise from interactions among equipment condition, process setpoints, and localized disturbances. Because offline sampling occurs only after defects have formed, it introduces a substantial information delay and allows abnormal conditions to persist across several production cycles. Industrial systems therefore need rapid visual inspection, state prediction, and parameter adjustment in one workflow so that quality information can directly inform process control.

Recent industrial detectors show that single-stage designs can provide useful speed and accuracy [1]. Lightweight cross-level fusion [2], efficient steel-strip feature aggregation [3], and architectures designed for tiny defects [4–6] improve sensitivity without the cost of a large two-stage model. Transformer-based context modeling has also been applied to PCB and hot-rolled-strip inspection [7, 8]. These studies advance real-time recognition, but their outputs usually end at class, box, and confidence; they do not define a timestamped quality state that can be consumed and verified by a process-control loop.

Digital-twin studies have linked quality prediction with assembly diagnosis [9], roughness estimation and parameter adjustment [10, 11], real-time process-variable prediction [12], coupled physical-data modeling [13], and virtual data generation [14]. The broader machining literature

identifies state synchronization, model evolution, and decision feedback as the functions that separate an operational twin from a static digital representation [15]. However, these approaches seldom use continuously refreshed visual defect states as both the predictor input and the post-adjustment verification signal.

Three integration gaps motivate this study: visual detections are rarely converted into compact process-ready states; twin predictors often lack continuously refreshed inspection evidence; and prediction, optimization, execution, and verification are not consistently governed by one cycle-level latency and safety policy. The contributions are therefore bounded to three points. First, the method builds a timestamped quality interface from multiscale detections. Second, it couples a lightweight temporal surrogate with constrained candidate evaluation, guarded execution, and next-cycle verification. Third, it evaluates detector accuracy, risk prediction, process outcomes, robustness, and latency in one reproducible workflow. The lightweight detector and GRU are implementation components; the principal contribution is their governed closed-loop coupling.

2. Methods

2.1. Real-time Vision-Digital Twin Closed-Loop Framework

The system comprises a physical production layer, a visual perception layer, a digital-twin layer, and a decision and execution layer, as shown in Fig. 1. The physical layer contains the equipment, camera, PLC, and sensors; the visual layer outputs location, class, confidence, and severity for each cycle. In this work, the twin is not the GRU alone. It includes a timestamped state cache, a normalized process-equipment-quality schema, the temporal response surrogate, the feasible operating envelope, a candidate evaluator, and a feedback updater. Candidate settings are evaluated inside the twin, whereas the decision and execution layer retains command authority, repeats the final constraint check, and sends only an admissible setting to the PLC.

The initial state of production cycle t comprises the visual image, process parameters, equipment condition, and quality state:

$$S_t = \{I_t, P_t, E_t, Q_t\} \quad (1)$$

where S_t denotes the system state in cycle t ; I_t denotes the current product image; $P_t = [p_{t,1}, \dots, p_{t,m}]$ denotes m -dimensional adjustable process parameters; E_t denotes the equipment operating state; and Q_t denotes the quality state derived from the visual model. All four data types are written to the twin-state cache under a common timestamp. To

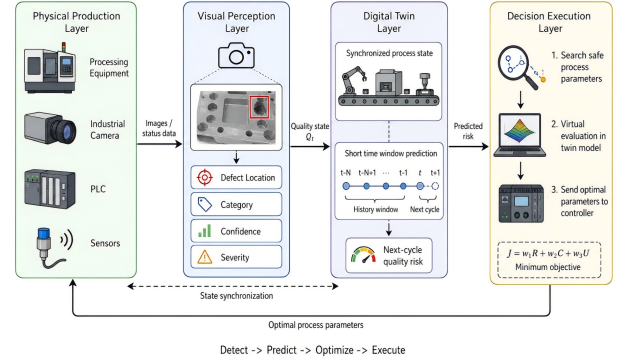


Fig. 1. Overall architecture of the real-time vision-digital-twin closed loop

prevent false associations due to asynchronous sampling, process and equipment measurements are aligned to the image-exposure time by nearest-neighbor registration. The maximum permitted time offset is set to 20% of one visual sampling period; samples beyond this threshold are excluded from online correction. In the experiments, the camera event defines a 1.0 s cycle, process and equipment signals are sampled at 50 Hz, and the absolute alignment limit is 200 ms. One missing record permits the last validated state to be held for one cycle; two consecutive invalid alignments inhibit search and retain the last safe setting.

The closed-loop sequence consists of visual inference, state synchronization, quality prediction, parameter optimization, and command execution. Total latency is defined as:

$$T_{\text{loop}} = T_{\text{vis}} + T_{\text{sync}} + T_{\text{pred}} + T_{\text{opt}} + T_{\text{comm}} \quad (2)$$

Here, T_{loop} denotes the end-to-end closed-loop delay; T_{vis} denotes the visual inference delay; T_{sync} denotes the data-synchronization delay; T_{pred} denotes the twin-prediction delay; T_{opt} denotes the parameter-search delay; and T_{comm} denotes the control-command communication delay. When $T_{\text{loop}} < T_{\text{cycle}}$, the system can complete a quality adjustment before the next production cycle begins, where T_{cycle} is the actual production-cycle time. The measured 34.8 ms is an average within the 1.0 s experimental cycle and is not treated as a certified worst-case deadline.

2.2. Lightweight Multi-Scale Defect Detection Network

The vision module uses a single-stage detector composed of a lightweight backbone, multiscale feature layers, an adaptive fusion module, and a detection head, as shown in Fig. 2. Input images are resized to 640×640 pixels. The stem uses strided convolutions for initial downsampling,

and the remaining backbone uses depthwise separable convolutions to reduce computational cost. The P3, P4, and P5 feature maps retain spatial resolutions of 80×80 , 40×40 , and 20×20 , respectively, to represent small, localized, and larger defects. Instead of directly stacking features from multiple levels, the adaptive fusion module derives weights from the effective response at each level, which limits interference from strongly textured backgrounds when detecting small defects.

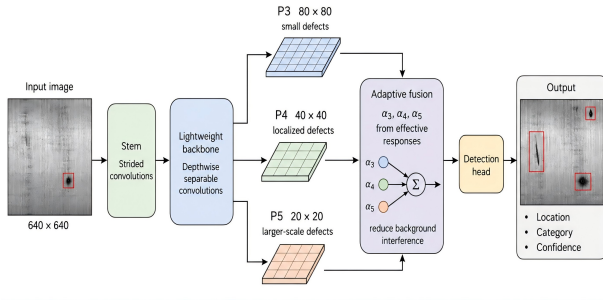


Fig. 2. Architecture of the lightweight multiscale defect detector

Let F_i denote the input feature at scale i . Channel mapping with a 1×1 convolution, followed by spatial alignment, produces a feature representation with a common dimension.

$$\hat{F}_i = \phi_i(F_i), \quad i \in \{1, 2, 3\} \quad (3)$$

where $\phi_i(\cdot)$ denotes the channel-mapping and interpolation operation at scale i ; $\hat{F}_i$ is the aligned feature. Global average pooling and two lightweight mapping layers are then applied at each scale to obtain the scale descriptor g_i . A Softmax operation converts these descriptors into fusion weights. All three inputs are projected to 128 channels and bilinearly aligned to the 80×80 P3 grid. Each descriptor produces one scalar per scale and per image, and Softmax is evaluated only across the three scales.

$$\alpha_i = \frac{\exp(g_i)}{\sum_{j=1}^3 \exp(g_j)} \quad (4)$$

Here, α_i is the weight assigned to scale i , subject to $\sum_i \alpha_i = 1$; g_i is the response descriptor at that scale. The fused feature is defined as:

$$F_f = \sum_{i=1}^3 \alpha_i \hat{F}_i \quad (5)$$

where F_f denotes the fused feature passed to the detection head. The training objective combines classification, bounding-box regression, and object-confidence losses:

$$\mathcal{L}_{det} = \lambda_c \mathcal{L}_{cls} + \lambda_b \mathcal{L}_{box} + \lambda_o \mathcal{L}_{obj} \quad (6)$$

Here, $\mathcal{L}_{det}$ denotes the total detection loss; $\mathcal{L}_{cls}$, $\mathcal{L}_{box}$ and $\mathcal{L}_{obj}$ are the classification, bounding-box regression, and object-confidence losses, respectively; λ_c , λ_b and λ_o are the corresponding weights. The three weights remain fixed so that every comparison model is trained under the same conditions.

The detections are then converted into a quality vector that the digital-twin predictor can use. If N_t defects are detected in the current cycle, the vector is defined as:

$$Q_t = [n_t, r_t, \bar{c}_t, z_t] \quad (7)$$

where $n_t = N_t$ is the number of defects; r_t is the ratio of the defect-box area to the effective product area; $\bar{c}_t$ is the mean detection confidence; and z_t is the overall severity.

Severity combines the defect-class weight, area ratio, and confidence. The resulting vector preserves the number and severity of defects without imposing the online computational cost of feeding high-dimensional image features directly into the twin model. Class weights are fixed before closed-loop testing: scratches and crazing use 1.0, pitted surfaces and rolled-in scale 0.8, and inclusions and patches 0.6. The four elements are scaled with training-set limits and combined into the observed risk using coefficients 0.20, 0.30, 0.10, and 0.40 for count, area, confidence, and severity; the result is clipped to $[0, 1]$. Defect rate denotes the fraction of defective products in an evaluation window and is not interchangeable with this normalized risk.

2.3. Digital Twin State Synchronization and Quality Prediction

The digital-twin layer maintains a dynamic mapping of process, equipment, and quality states. The twin input for cycle t is denoted by $X_t = [P_t, E_t, Q_t]$. A window contains L consecutive cycles:

$$X_t = [X_{t-L+1}, X_{t-L+2}, \dots, X_t] \quad (8)$$

where X_t is a state sequence of length L . A two-layer GRU serves as the lightweight temporal predictor and outputs the defect risk $\hat{q}_{t+1}$ for the next cycle. The GRU captures short-term state dependence at a lower computational cost than a large attention model and therefore fits the closed-loop latency requirement. Training minimizes mean squared error between the predicted risk and the visual observation. During online operation, a sliding window is used to update the final part of the predictor. The common input window is $L = 12$ cycles, and continuous process

and equipment variables are standardized with training-set statistics.

When the visual result for a new cycle becomes available, the prediction error is defined as:

$$e_t = q_t - \hat{q}_t \quad (9)$$

Here, q_t is the observed defect risk derived from the visual quality vector, $\hat{q}_t$ is the risk predicted by the twin model, and e_t is the virtual-to-physical deviation. If the mean absolute error over K consecutive cycles exceeds the threshold δ , the system triggers a lightweight online update. The update rule is:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} \mathcal{L}_t \quad (10)$$

where θ_t denotes the current predictor parameters; η is the online learning rate; and $\mathcal{L}_t$ is the prediction loss over the most recent window. Online correction updates only the output layer and the final GRU layer, limiting parameter drift and computational overhead. This restricted update allows the twin to adapt to slowly varying shifts associated with material batches, equipment heating, and load changes. The trigger uses $K = 5$, $\delta = 0.025$, and $\eta = 0.0001$. One gradient step is taken with norm clipping at 1.0. A 24-cycle shadow window rejects an update if loss increases or a predicted risk leaves $[0,1]$, in which case the preceding parameters are restored.

2.4. Constrained Process Parameter Optimization

The digital-twin predictor also evaluates candidate parameter settings. Let the adjustable variables be speed v , temperature T , pressure p , and feed rate f , so that $P = [v, T, p, f]$. Minimizing defect rate alone could reduce throughput or increase energy use. We therefore define a joint objective that accounts for quality risk, efficiency, energy consumption, and the magnitude of parameter changes:

$$J(P) = w_1 R_{\text{def}}(P) + w_2 C_{\text{ene}}(P) + w_3 C_{\text{time}}(P) + w_4 \|P - P_t\|_2^2 \quad (11)$$

Here, $J(P)$ is the objective value for a candidate setting; $R_{\text{def}}(P)$ is the defect risk predicted by the twin model; $C_{\text{ene}}(P)$ is normalized energy consumption; $C_{\text{time}}(P)$ is the normalized production-cycle penalty; P_t is the current parameter vector; and w_1 through w_4 are non-negative weights that sum to 1. The quality-risk weight is set highest so that the search gives priority to product quality. After training-range normalization, the experiments use weights 0.55, 0.15, 0.15, and 0.15 for risk, energy, cycle time, and adjustment magnitude, respectively.

Parameter search combines small-scale Latin hypercube sampling with local-neighborhood refinement. In

each closed-loop cycle, M candidate settings are generated within the safe region Ω . The twin model predicts the risk for each candidate, after which the best 20% undergo local search. The final setting is: We use $M = 120$; the best 24 candidates each receive ten bounded neighborhood trials.

$$P_{t+1}^* = \operatorname{argmin}_{P \in \Omega} J(P) \quad (12)$$

where P_{t+1}^* is the setting executed in the next cycle; Ω is the feasible region defined by equipment limits, parameter rate-of-change limits, and process rules. To prevent unnecessary adjustments, the current setting is retained when $|J(P_t) - J(P_{t+1}^*)| < \epsilon$, where ϵ is the minimum effective improvement threshold. After an optimized setting is issued, the next cycle's visual result evaluates the effect of the adjustment and is fed into the online correction in Eq. (10), thereby closing the loop. Each normalized variable is rate-limited to 0.08 per cycle and $\epsilon = 0.005$. Commands carry the source-cycle identifier and timestamp, are rounded before a second feasibility check, and require PLC acknowledgement within 100 ms. A stale, rejected, or unacknowledged command is cancelled, the last acknowledged safe setting remains active, and no model update is accepted from an unconfirmed actuation. These interlocks support the experiment but do not constitute functional-safety certification.

3. results and discussion

3.1. Experimental Setup

The vision experiments use the public NEU surface-defect dataset as the primary detection dataset, stratified by class into training, validation, and test subsets. For the digital-twin experiments, a nonlinear quality-response model is constructed from four adjustable process parameters, three equipment-state variables, and the visual quality variables. The simulation includes batch drift, random noise, and two abrupt disturbance types. Using a fixed random seed of 20260820, we generate 12,000 production cycles: 7,200 for twin-model training, 1,800 for validation, and 3,000 for closed-loop testing. All comparison methods use the same state sequences and disturbance times. Detection is evaluated using precision, recall, mAP@0.5, and mAP@0.5:0.95; real-time performance using FPS, per-frame latency, and parameter count; quality prediction using MAE, RMSE, and R^2 ; and closed-loop performance using average defect rate, first-pass yield (FPY), relative energy consumption per cycle, and T_{loop} . Each closed-loop result is based on 20 independent disturbance realizations. Tables report the mean; for the main indicators, standard deviations are below 5.1% of the corresponding means. NEU denotes the Northeast-

ern University surface-defect dataset and contains 1,800 images, 300 per class; stratified counts are 1,080/360/360 for training/validation/test. All detectors are retrained at 640×640 for 200 epochs with batch size 16, AdamW, initial learning rate 0.001, and 20-epoch early stopping. Training-only augmentation uses horizontal/vertical flips, 0.8-1.2 scaling, brightness variation within 15%, and mild blur. Tests use an Intel i7-12700, RTX 306012 GB, PyTorch 2.1, and CUDA 12.1. FPS uses batch size 1 after 100 warm-up frames over 1,000 frames. Predictor hyperparameters are selected on validation data only. The 20 process trials vary noise and disturbance sequences, with each sequence paired across controllers; reported variation is descriptive simulation variability, not independent plant replication.

3.2. Defect Detection Performance

Table 1 compares detectors retrained and timed under Section 3.1. Faster R-CNN and RT-DETR-R18 trade more latency and parameters for context, while the YOLO variants provide higher throughput. The proposed detector reaches 94.1% mAP@0.5 and 63.8% mAP@0.5:0.95 with 6.8 M parameters at 78.6 FPS. Its lower speed than YOLOv8n reflects the retained P3 path and three-scale alignment, yielding an accuracy-speed compromise.

Fig. 3 compares the five detectors on scratches, inclusions, and patches. Before inspecting model outputs, one test image was chosen from each of three morphology groups as the sample nearest its class median in annotated area and local contrast: elongated, compact, and diffuse defects, respectively. The three categories fit the available page width and do not replace class-wide evaluation. Faster R-CNN produces wider boxes, YOLOv5s loses parts of elongated or weak regions, and YOLOv8n and RT-DETR-R18 retain small boundary offsets in complex texture. The proposed boxes more closely follow the visible regions. These images illustrate localization behavior only; aggregate conclusions rely on Table 1.

The gain is consistent with prior evidence that high-resolution multiscale features support small-defect inspection [2–7]. Here, P3 preserves fine edges, while the learned scale weights limit responses to coarse texture. Classwise AP@0.5 ranges from 91.2% to 96.8%; crazing is lowest because very thin, low-contrast traces lose detail after resizing. Remaining false positives occur mainly where rolled texture resembles short scratches. Thus, the confidence component can reduce the influence of uncertain detections on the twin input, but it cannot remove visual failure, especially under brightness change or blur.

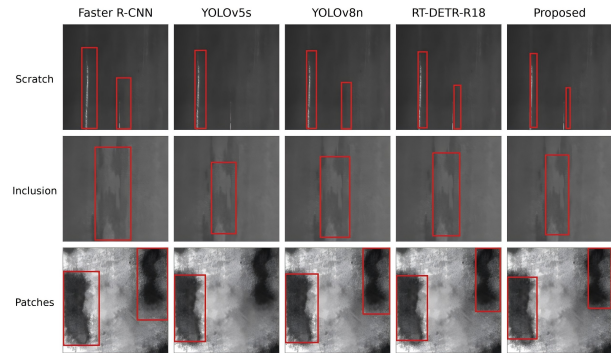


Fig. 3. Qualitative comparison of defect-localization behavior on representative NEU test images

3.3. Digital Twin Quality Prediction

Table 2 compares linear regression, random forest, SVR, LSTM, GRU, and the proposed online-corrected predictor. The linear model cannot adequately represent the coupled effects of temperature, speed, and pressure and yields an RMSE of 0.041. Random forest and SVR capture more of the nonlinearity but respond slowly to sustained disturbances. LSTM and GRU use state changes within the time window and reduce the RMSE to 0.026 and 0.024, respectively. Adding the visual quality state and online correction further lowers the RMSE to 0.018 and raises R^2 to 0.923.

3.4. Closed-Loop Process Optimization Effect

Fig. 4 shows the response to a simulated material-batch change at 200 s and machine-load disturbance at 420 s under the common nonlinear response model. Fixed parameters recover most slowly; rule-based control lowers the peak but is delayed by threshold triggering and fixed increments. The twin-only strategy shortens recovery, whereas the full loop returns risk to the stable range within several cycles by combining prediction, constrained action, and visual verification. This controlled result does not show that higher visual mAP alone causes fewer defects; rather, improved inspection supports a more reliable state interface within the complete simulated loop.

3.5. Ablation and Real-Time Performance Analysis

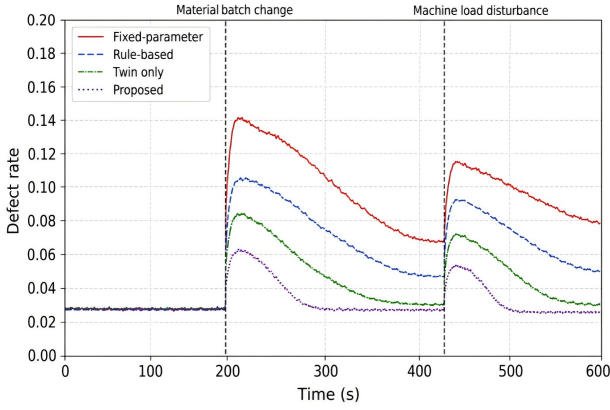
Table 3 begins with YOLOv5s and an uncalibrated GRU, then adds the multiscale detector, visual quality state, online calibration, and constrained optimization. Multiscale redesign increases mAP@0.5 from 92.7% to 94.1%; later twin components leave detector accuracy unchanged. With the same backbone and head, equal-weight fusion obtains 93.4%, compared with 94.1% for adaptive weighting. The visual state reduces twin RMSE from 0.028 to 0.024, online calibration reduces it to 0.018, and optimization lowers

Table 1. Comparison of real-time defect-detection models.

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Params (M)	FPS
Faster R-CNN	90.3	88.1	92.0	60.7	41.2	24.5
YOLOv5s	90.8	89.6	92.7	59.8	7.2	91.4
YOLOv8n	91.7	90.2	93.2	61.1	3.2	96.8
RT-DETR-R18	92.4	90.9	93.7	62.5	20.0	51.2
Proposed	93.1	91.8	94.1	63.8	6.8	78.6

Table 2. Quality-risk prediction performance of the digital twin.

Predictor	MAE	RMSE	R ²	Inference time (ms)
Linear regression	0.032	0.041	0.652	0.18
Random forest	0.024	0.033	0.774	2.86
SVR	0.022	0.031	0.801	1.72
LSTM	0.019	0.026	0.857	3.94
GRU	0.017	0.024	0.876	3.31
Proposed twin predictor	0.012	0.018	0.923	3.68

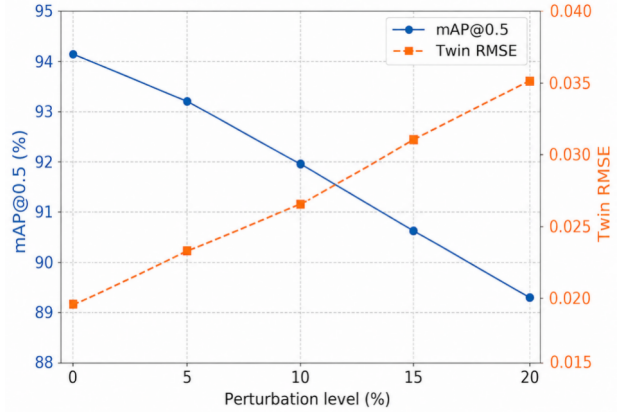
**Fig. 4.** Dynamic defect-rate response under two process disturbances

defect rate from 4.41% to 2.94%. Mean latency comprises 12.7 ms visual inference, 2.4 ms synchronization, 3.7 ms prediction, 12.8 ms search, and 3.2 ms communication, totaling 34.8 ms. This is an average within the 1.0 s experimental cycle, not a certified worst-case deadline.

3.6. Robustness Analysis

The combined stress test increases brightness shift, Gaussian noise, motion blur, and process-state noise from 0% to 20%. At 20%, mAP@0.5 falls from 94.1% to 89.4% and twin RMSE rises from 0.018 to 0.035. Two one-factor checks separate the mechanisms: visual perturbation alone produces 89.6% mAP and 0.021 RMSE, whereas state noise alone leaves mAP at 94.1% but raises RMSE to 0.032. The

combined result therefore reflects direct visual loss, state-measurement error, and propagation of imperfect visual states. Degradation remains gradual, but these controlled perturbations do not cover all field lighting or communication failures.

**Fig. 5.** Robustness under increasing visual and process-state perturbations

4. conclusion

This study develops and evaluates a closed-loop quality-control method that links real-time defect detection, synchronized risk prediction, constrained parameter search, and next-cycle verification. The detector achieves 94.1% mAP@0.5 and 63.8% mAP@0.5:0.95 with 6.8 M parameters at 78.6 FPS; the online-corrected twin predictor reaches

Table 3. Ablation study of the proposed closed-loop framework.

Configuration	mAP@0.5 (%)	Twin RMSE	Defect rate(%)	Loop latency (ms)
Baseline	92.7	0.029	6.84	19.6
+ Multi-scale fusion	94.1	0.028	6.21	23.4
+ Visual quality state	94.1	0.024	5.12	25.8
+ Online calibration	94.1	0.018	4.41	28.7
+ Constrained optimization	94.1	0.018	2.94	34.8

0.018 RMSE. Under the defined simulated disturbances, the complete loop reduces average defect rate from 7.82% to 2.94%, raises first-pass yield to 96.9%, and operates with 34.8 ms mean latency. These findings are limited to one public image dataset, a controlled response model, and the specified timing and safety rules. They do not establish transfer across equipment, camera configurations, material batches, or operator interventions. Future work will use multi-line production data to evaluate cross-equipment transfer, long-term drift, multi-station coordination, and safety-governed adaptation before broader deployment claims are made.

Acknowledgements

The work was supported by the Henan Province Science and Technology Research and Development Project for 2025 (No. 252102210062)

References

- [1] H. F. Le, L. J. Zhang, and Y. X. Liu, (2022) "Surface Defect Detection of Industrial Parts Based on YOLOv5" **IEEE Access** **10**: 130784–130794. DOI: [10.1109 / ACCESS.2022.3228687](https://doi.org/10.1109/ACCESS.2022.3228687).
- [2] X. Qian, X. Wang, S. Yang, and J. Lei, (2022) "LFF-YOLO: A YOLO Algorithm with Lightweight Feature Fusion Network for Multi-Scale Defect Detection" **IEEE Access** **10**: 130339–130349. DOI: [10.1109 / ACCESS.2022.3227205](https://doi.org/10.1109/ACCESS.2022.3227205).
- [3] Y. Li, S. Xu, Z. Zhu, P. Wang, K. Li, Q. He, and Q. Zheng, (2023) "EFC-YOLO: An Efficient Surface-Defect-Detection Algorithm for Steel Strips" **Sensors** **23**(17): 7619. DOI: [10.3390/s23177619](https://doi.org/10.3390/s23177619).
- [4] Z. Yu, Y. Wu, B. Wei, Z. Ding, and F. Luo, (2023) "A Lightweight and Efficient Model for Surface Tiny Defect Detection" **Applied Intelligence** **53**(6): 6344–6353. DOI: [10.1007/s10489-022-03633-x](https://doi.org/10.1007/s10489-022-03633-x).
- [5] J. Lu, M. Yu, and J. Liu, (2024) "Lightweight Strip Steel Defect Detection Algorithm Based on Improved YOLOv7" **Scientific Reports** **14**(1): 13267. DOI: [10.1038/s41598-024-64080-x](https://doi.org/10.1038/s41598-024-64080-x).
- [6] S. Gao, M. Chu, and L. Zhang, (2024) "A Detection Network for Small Defects of Steel Surface Based on YOLOv7" **Digital Signal Processing** **149**: 104484. DOI: [10.1016/j.dsp.2024.104484](https://doi.org/10.1016/j.dsp.2024.104484).
- [7] W. Chen, Z. Huang, Q. Mu, and Y. Sun, (2022) "PCB Defect Detection Method Based on Transformer-YOLO" **IEEE Access** **10**: 129480–129489. DOI: [10.1109 / ACCESS.2022.3228206](https://doi.org/10.1109/ACCESS.2022.3228206).
- [8] Q. Wang, H. Dong, and H. Huang, (2024) "Swin-Transformer-YOLOv5 for Lightweight Hot-Rolled Steel Strips Surface Defect Detection Algorithm" **PLOS ONE** **19**(1): e0292082. DOI: [10.1371/journal.pone.0292082](https://doi.org/10.1371/journal.pone.0292082).
- [9] C. Zhuang, Z. Liu, J. Liu, H. Ma, S. Zhai, and Y. Wu, (2022) "Digital Twin-Based Quality Management Method for the Assembly Process of Aerospace Products with the Grey-Markov Model and Apriori Algorithm" **Chinese Journal of Mechanical Engineering** **35**(1): 105. DOI: [10.1186/s10033-022-00763-8](https://doi.org/10.1186/s10033-022-00763-8).
- [10] L. Liu, X. Zhang, X. Wan, S. Zhou, and Z. Gao, (2022) "Digital Twin-Driven Surface Roughness Prediction and Process Parameter Adaptive Optimization" **Advanced Engineering Informatics** **51**: 101470. DOI: [10.1016/j.aei.2021.101470](https://doi.org/10.1016/j.aei.2021.101470).
- [11] Z. Zhao, S. Wang, Z. Wang, S. Wang, C. Ma, and B. Yang, (2022) "Surface Roughness Stabilization Method Based on Digital Twin-Driven Machining Parameters Self-Adaption Adjustment: A Case Study in Five-Axis Machining" **Journal of Intelligent Manufacturing** **33**(4): 943–952. DOI: [10.1007/s10845-020-01698-4](https://doi.org/10.1007/s10845-020-01698-4).
- [12] M. Perno, L. Hvam, and A. Haug, (2023) "A Machine Learning Digital Twin Approach for Critical Process Parameter Prediction in a Catalyst Manufacturing Line"

- Computers in Industry** 151: 103987. DOI: [10.1016/j.compind.2023.103987](https://doi.org/10.1016/j.compind.2023.103987).
- [13] Y. Wang, F. Tao, Y. Zuo, M. Zhang, and Q. Qi, (2023) “Digital-Twin-Enhanced Quality Prediction for the Composite Materials” **Engineering** 22: 23–33. DOI: [10.1016/j.eng.2022.08.019](https://doi.org/10.1016/j.eng.2022.08.019).
- [14] Y. Zuo, H. You, X. Zou, W. Ji, and F. Tao, (2024) “Digital Twin Enhanced Quality Prediction Method of Powder Compaction Process” **Robotics and Computer-Integrated Manufacturing** 89: 102762. DOI: [10.1016/j.rcim.2024.102762](https://doi.org/10.1016/j.rcim.2024.102762).
- [15] S. Liu, J. Bao, and P. Zheng, (2023) “A Review of Digital Twin-Driven Machining: From Digitization to Intellectualization” **Journal of Manufacturing Systems** 67: 361–378. DOI: [10.1016/j.jmsy.2023.02.010](https://doi.org/10.1016/j.jmsy.2023.02.010).