

An Investigation of Defect Inspection Performance through Human-Robot Collaboration

Satrio Sanjaya¹, Muhammad Ahsan¹, Alberto Olivares-Alarcos², Hsien-I Lin³, and Guillem Alenyà²

Abstract—In modern manufacturing, visual inspection of metallic components such as solid-state drive (SSD) cases demands high precision and operational flexibility. This paper presents a collaborative human-robot inspection approach that combines high-throughput automated robot inspection with reliable and flexible human oversight. The system uses a robotic arm equipped with an end-effector-mounted 2D camera for SSD surface inspection. When robot prediction confidence exceeds a predefined threshold, the SSD is automatically classified as either *Good* or *Defective*. If confidence falls below this threshold, the system flags the SSD for human inspection. A tray with SSD cases is placed on an array of infrared proximity sensors to monitor human inspection activity. Placing an SSD back to its slot implies acceptance (*Good*), while removal indicates rejection (*Defective*). The shared plan execution is implemented using behavior trees, enabling asynchronous and intention-aware human-robot coordination. A real-time graphical interface displays the inspection state for robot-to-human communication. The approach is evaluated using different robot vision confidence thresholds, simulating more and less challenging scenarios (e.g., changes in lighting conditions). Experimental results demonstrate that under high-confidence settings, the robot alone achieved over 1600 units per hour (UPH). In low-confidence conditions requiring human intervention, the system still maintained a throughput above 900 UPH, ensuring robust defect validation. These findings confirm that the proposed collaborative solution effectively balances the repeatability and speed of robot inspection with the adaptability and reliability of human decision-making, outperforming purely human-only or robot-only baselines.

Index Terms—Human-Robot collaboration, visual inspection, Behavior Tree.

I. INTRODUCTION

Visual inspection remains a critical task in modern manufacturing to ensure product quality and reliability, especially for high-precision components such as metallic solid-state drive (SSD) cases. Traditional inspection methods rely heavily on human operators, offering flexibility and contextual judgment in ambiguous scenarios. However, these manual approaches are susceptible to fatigue, inconsistency, and limited scalability. In contrast, fully automated systems provide high throughput and repeatability but often lack robustness when faced with different lighting conditions or uncertain or subtle defect patterns that fall outside learned distributions. To address these

¹Department of Electrical Engineering and Computer Science International Graduate Program, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: satrio.sanjaya.fv@um.ac.id, muhammadahsanfs.ee12@nycu.edu.tw).

²Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028 Barcelona, Spain (e-mail: {aolivares,galenya}@iri.upc.edu).

³Corresponding author: Institute of Electrical and Control Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: sofin@nycu.edu.tw).

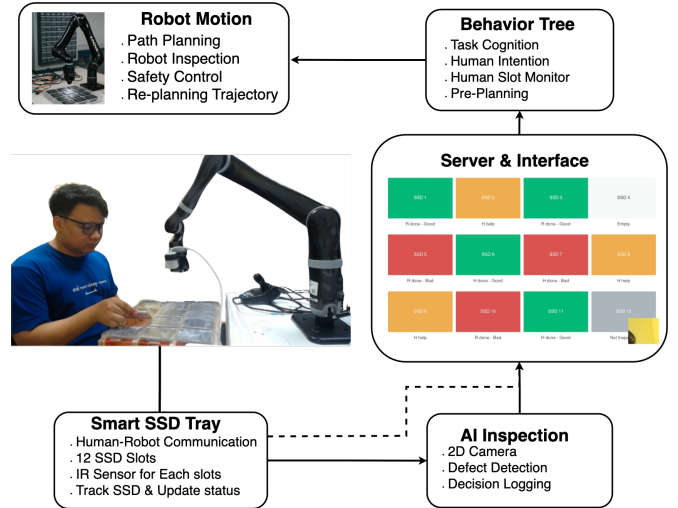


Fig. 1. Overview of the proposed approach for human-robot collaborative inspection of SSD cases. The image shows the pipeline of the human-robot plan execution, from the identification of the human activity through our smart SSD tray, to the robot decision-making process and visual inspection.

limitations, human-robot collaboration (HRC) has emerged as a promising paradigm that integrates the precision of robotic systems with the adaptability of human intelligence [1]–[3]. By leveraging both strengths, HRC systems can maintain high inspection accuracy while gracefully handling uncertain cases via human intervention.

This paper presents a human-robot collaborative system for the surface inspection of metallic SSD cases. As illustrated in Fig. 1, the proposed system integrates a 6-DOF robotic arm equipped with a 2D camera for visual data acquisition, an AI-based defect detection model for real-time classification, and a user interface for seamless human-robot communication. Each inspection cycle processes a tray containing 12 SSD units. The robotic arm autonomously captures high-resolution images of each unit, which are analyzed by a deep learning model trained to detect surface anomalies. When the AI model produces high-confidence predictions (e.g., above a certain threshold), the result is autonomously recorded. For low-confidence cases, the system defers to the human operator for review and decision-making.

To enable robust coordination between the robot and the human, we employ a smart SSD tray with embedded infrared sensors under each slot. These sensors detect object presence and transitions, enabling the system to infer human input through passive observation. The state information is propagated to a behavior tree that governs the inspection task

plan execution, coping with human intention recognition, and role switching logic. A graphical user interface (GUI) provides real-time visualization of SSD states, enabling transparent monitoring of inspection results and decision provenance.

The main contributions of this work are:

- An integrated HRC-based inspection system, incorporating robotic motion, vision-based defect detection, and sensor-assisted human communication through a smart SSD tray;
- An evaluation of the proposed method in a laboratory mock-up of real-world industrial setting, demonstrating a good balance between high-throughput and a flexible and reliable performance.

II. RELATED WORKS

A. Automated Visual Inspection and Anomaly Detection

Deep learning-based visual inspection systems have shown remarkable performance in detecting surface defects in industrial applications. Roth et al. [4] proposed PatchCore, a memory-based anomaly detection method that achieves high accuracy by using sparse patch representations and nearest neighbor comparisons. Despite its strong performance on benchmark datasets, PatchCore involves memory-intensive operations and lacks real-time inference capability, which limits its suitability for high-throughput production lines.

EfficientDet, introduced by Tan et al. [5], presents a scalable object detection architecture using compound model scaling. Although it offers a favorable trade-off between accuracy and efficiency for general object detection tasks, it is not specifically tailored to detect subtle surface anomalies common in manufacturing, such as scratches or dents on metallic surfaces.

Sanjaya et al. [6] developed an automated inspection system for metallic flywheel components using a deep learning model integrated with custom hardware, achieving 95% accuracy and 40 fps while detecting defects as small as 0.5 mm. Lin et al. [7] proposed a defect detection approach for metal laptop cases that combines up-sampling and down-sampling with dilation convolution to enhance high-semantic feature maps, reaching 81% accuracy and outperforming other conventional deep learning methods. Both studies highlight the effectiveness of customized automated inspection solutions in handling domain-specific challenges, such as reflective surfaces, fine-grained textures, and the need for rapid, high-accuracy defect identification in industrial environments.

B. Human-Robot Collaboration in Industrial Environments

Human-robot collaboration (HRC) has gained significant attention as a means to enhance task flexibility, safety, and productivity in industrial settings. Castro et al. [8] reviewed recent advancements in HRC, including handover mechanisms, learning-based adaptation, and evaluation metrics. However, their survey mainly addresses generic collaboration workflows, with limited focus on inspection tasks requiring high precision and role-switching capabilities.

Rodriguez-Guerra et al. [9] conducted a comprehensive review of human-robot interaction in modern industrial environments, highlighting challenges such as safe coexistence,

intuitive communication, and system adaptability. Although their work underscores the importance of real-time interaction, it does not provide concrete solutions for integrating vision-based decision-making with collaborative inspection routines.

In the context of motion behavior, Gäbert et al. [10] proposed a method to generate human-like robot arm motions using sampling-based motion planning. While this approach enhances predictability and acceptance during collaboration, it does not incorporate task-specific feedback from visual inspection systems, which is essential for dynamic role assignment in inspection scenarios.

C. Trust and Interaction in Human-AI Systems

Establishing user trust is essential for effective human-AI collaboration, particularly in inspection tasks where erroneous decisions may have significant consequences. Bach et al. [11] conducted a systematic literature review on user trust in AI-enabled systems from an HCI perspective. Their study identifies explainability, reliability, and transparency as critical factors influencing trust. However, most of the discussed systems are outside industrial domains and do not address the real-time interaction requirements of visual inspection workflows.

Prati et al. [12] advocated for the use of interaction design methodologies to improve human-robot collaboration in industrial applications. Their work emphasizes the importance of ergonomic and intuitive interfaces, yet lacks integration with AI-based perception systems. Moreover, it does not address how inspection outcomes and decision uncertainties can be effectively communicated to human operators during task execution.

III. PROPOSED METHOD

The proposed system is composed of four main functional modules: a robotic manipulator for motion execution and trajectory planning, an AI-based visual inspection module, a smart SSD tray equipped with embedded sensors, and a behavior tree-based task execution model. These components work in harmony to support asynchronous collaboration between a human inspector and a robot. The server-side graphical user interface (GUI), integrated within the behavior tree framework, provides real-time visualization of task progress and inspection results. The complete system pipeline is illustrated in Fig. 1.

A. Robot Motion: Trajectory Planning

In this work, we use the Kinova Jaco platform. The inspection sequence follows a structured zigzag trajectory across the 12 SSD slots, as illustrated in Fig. 2. This pattern ensures spatial continuity, minimizing end-effector travel distance and idle time between inspections. The left-to-right, row-wise traversal enables predictable motion, which supports visual tracking and enhances safety in human-robot collaborative scenarios. Although the sequence is predefined for the current setup, the approach can be easily adapted for different tray configurations or dynamic planning strategies in future implementations.

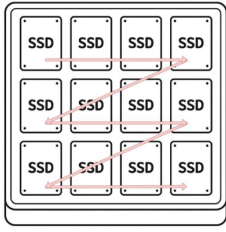


Fig. 2. Trajectory that the robot would follow based on a zigzag predefined SSD cases inspection sequence.

B. AI-Based Defect Detection

Our defect detection module builds on the foundation of patch-based anomaly detection, with enhancements for improved feature adaptability and robust discrimination. As illustrated in Fig. 3, the system begins by extracting spatial features from the input image using a convolutional backbone. These local features are then transformed by a feature adaptor that projects them into a latent space better aligned with anomaly representations. In parallel, synthetic anomalous features are generated to introduce pseudo-defective patterns, which are crucial for training the model in a supervised contrastive manner.

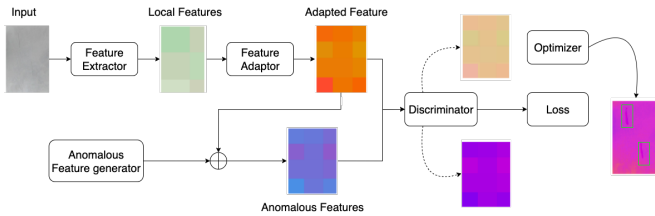


Fig. 3. Overview of the defect detection model

A discriminator is trained to separate normal from anomalous representations using a loss function that encourages feature separability. The resulting model produces an anomaly heatmap that highlights defective regions, along with a scalar confidence score used for classification. SSDs with scores above a certain threshold are labeled as *Good* or *Defective*, while uncertain cases are flagged for human review. Different threshold values are used during the evaluation in Sec. IV. Our model is inspired by SimpleNet [13], a baseline anomaly detection framework known for its simplicity and speed. However, unlike SimpleNet, which relies on nearest-neighbor distances in feature space, our model introduces feature adaptation and discriminator-based learning for improved generalization in SSD surface inspection.

C. Smart SSD Tray with Embedded Sensors

To enable passive and intuitive human-robot interaction, we designed a smart SSD tray consisting of 12 individual slots, each equipped with an infrared (IR) proximity sensor, as shown in Fig. 4. These sensors detect the presence or absence of SSD units by measuring reflected IR signals. A microcontroller continuously polls the sensor values and publishes the occupancy state of each slot in real-time to a dedicated

ROS topic. This state information is then integrated with the behavior tree system and user interface, enabling seamless synchronization between physical actions and software-level task execution.



Fig. 4. Smart Tray for 12 SSD.

Each SSD slot transitions between predefined states based on sensor input and interaction context. These transitions are used to monitor the inspection process, detect agent actions, and coordinate collaborative inspection workflows. For example, when a slot transitions from state 0 (Not Inspected) to 6 (Empty), it indicates that a human has picked up the SSD for manual inspection. If the slot later returns to state 1 (Human Done – Good) or 3 (Human Done – Bad), this is interpreted as the completion of human inspection. If the vision system produces a low-confidence result, the state 5 (Human Help) is triggered, prompting the robot to route the SSD for human validation. State 6 (Empty) is a system-level status to indicate that the SSD slot is unoccupied and ready for reset.

Table I describes the complete mapping of states used in the system, including their semantic meanings. These states are used by both the GUI and behavior tree logic to support transparent and asynchronous collaboration.

TABLE I
SLOT STATUS MAPPING IN HUMAN-ROBOT COLLABORATIVE INSPECTION

State	Label	Description	Color
0	Not Inspected	Initial state before any inspection activity	gray
1	Human Done – Good	Human reinspected SSD indicating it is good	green
2	Robot Done – Bad	Robot classified SSD as defective	red
3	Human Done – Bad	Human removed SSD indicating it is defective	red
4	Robot Done – Good	Robot classified SSD indicating as good	green
5	Human Help	Robot asked for human assistance (low confidence)	orange
6	Empty	Slot is empty	light gray

This smart tray thus serves a dual role: as a physical interface for placing and removing SSDs and as an implicit communication channel between human and robot. By embedding these status transitions into the system’s shared memory and decision logic, the robot can adaptively respond to inspection progress, route uncertain cases to the human, and track task completion without requiring any explicit input device or manual command.

D. Behavior tree for reactive human-robot collaborative visual inspection

Task plan coordination and execution is governed by a behavior tree framework. This framework interprets robot inspection decisions, sensor feedback, and operator actions to

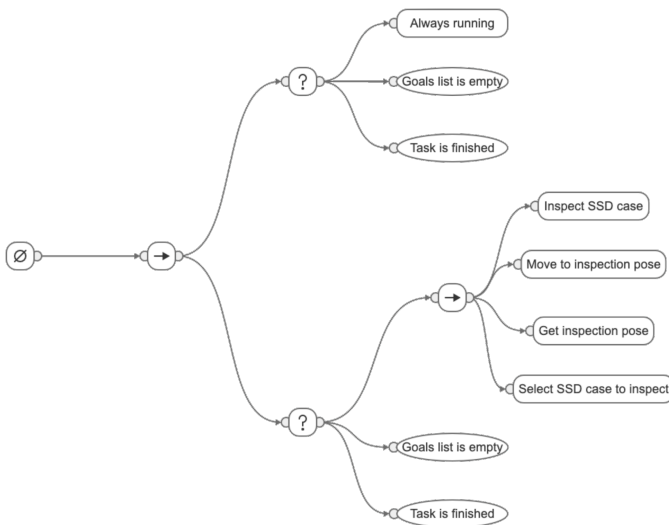


Fig. 5. Overview of the implemented behavior tree for robot SSD case inspection. The tree starts with a sequence of two fallbacks. The first one checks the conditions of whether the task has been finished (i.e. all the cases are inspected) and whether the list of robot goals is empty (i.e. the robot has done its part of the task, which may be only half of the tray). When those conditions return false, the tree continues with a sequence of actions: selecting the next case, getting its inspection pose, moving there and inspecting it. If all the actions succeed, the second fallback ensures that the tree keeps running until the task is finished or the robot’s goal list is empty. Note that this is a simplified version of the actual behavior tree. For instance, some of the action nodes depicted here are indeed implemented as more complex sub-trees (e.g. move to inspection pose).

manage robot behavior and inspection flow (see Fig. 5). It supports asynchronous execution by distinguishing between robot-led and human-led tasks. Initially, the robot is commanded to inspect all the SSD cases, but it is able to adapt and skip those that are being inspected, or already inspected by the human.

The server collects all inspection outcomes and displays them on a graphical user interface. Real-time status for each SSD as shown in Table I is clearly indicated using a color-coded layout, as shown earlier in Fig. 1. The graphical interface also shows the agent (i.e. a robot or human) that has inspected a SSD case. This interface aims to enhance transparency, reduce human cognitive load, and facilitate an efficient collaborative decision making.

IV. VALIDATION IN A LAB MOCK-UP OF A REALISTIC INDUSTRIAL INSPECTION TASK

A. Experimental Setup

The experimental validation was conducted in a laboratory environment that replicates an industrial SSD inspection task. The setup includes a 6-degree-of-freedom Kinova Jaco2 robotic arm equipped with a wrist-mounted 2D RGB camera. Each slot can hold a single metallic SSD case. Infrared (IR) sensors are embedded beneath each slot to detect SSD picking and placing (i.e. human interaction with the tray). The robot operates under a Behavior Tree (BT)-based decision framework implemented in ROS Noetic. Rather than directly controlling the robot, the Behavior Tree issues high-level inspection commands, such as which SSD slot to inspect. These commands are then executed by the robot’s low-level

motion and perception modules, enabling coordinated visual inspection and seamless human collaboration.¹²

A deep learning-based defect detection model, as shown in Figure 3, is used to perform surface defect classification in real time. The robot captures an image of each SSD and sends it to the model for inspection. If the model’s confidence exceeds a certain threshold, the SSD is automatically labeled as *Good* or *Defective* and the robot proceeds to the next slot. If the confidence is below the threshold, the system flags the SSD for human help. During the evaluation, different thresholds were used aiming at simulating different robot profiles (i.e. more and less capable robot). A custom GUI displays the inspection status and confidence scores, allowing the human operator to intervene when necessary. The robot avoids interfering with any slot currently being handled by a human.

B. Experimental Design: Metrics and Parameters

To assess the performance of the proposed human-robot collaborative inspection system, several evaluation metrics were defined. The *Completion Time* represents the total duration from the start of the inspection process until all tasks are finished and the robot returns to its home position. *Human Workload* and *Robot Workload* indicate the number of SSD slots inspected by the human and robot, respectively. The *Total Time* for each agent refers to the cumulative time spent inspecting assigned slots, calculated independently and not accounting for parallel execution.

The *Low Confidence Count* quantifies the number of times that the AI vision model produced a prediction with confidence below the defined threshold, prompting human verification. *Units Per Hour (UPH)* is used as a throughput metric, estimating the number of SSDs inspected per hour. It is computed as:

$$UPH = \frac{12}{T_c} \times 3600 \quad (1)$$

where T_c denotes the Completion Time in seconds.

Additionally, *Robot Autonomy (%)* expresses the proportion of SSDs inspected autonomously by the robot. Experiments were conducted under three confidence thresholds: default (no filtering), 90%, and 95%. When the model’s confidence exceeded the threshold, the prediction was accepted automatically; otherwise, the SSD was marked as doubtful and passed to the human. All trials were performed with 12 SSDs per tray, under consistent lighting and timing conditions, with execution managed by a ROS-based Behavior Tree framework.

C. Experimental Results

Table II summarizes the performance of the human-robot collaborative inspection system across 18 trials under three confidence thresholds: default (HRC), 90%, and 95%. Each trial involved inspecting 12 SSDs in a standardized tray layout. The robot inspected high-confidence slots, while those below the threshold were delegated to the human operator (*Low Conf.* column).

¹https://github.com/ahsanfs/ssd_inspection_action_function

²https://github.com/albertoOA/iri_ssd_case_inspection_robot_behavior

TABLE II
EXPERIMENTAL RESULTS OF HUMAN-ROBOT COLLABORATIVE, HUMAN-ONLY, AND ROBOT-ONLY INSPECTION

Trial Name	Workload (slots)		Total Time (s)		Low Conf.	Completion Time (s)	UPH ^a	Robot (%)
	Human	Robot	Human	Robot				
TR1_1_HRC	5	7	18.00	21.00	0	26.90	1605.75	58.3
TR2_1_HRC	6	6	12.00	18.00	0	24.99	1728.18	50.0
TR1_2_HRC	5	7	19.00	19.00	0	26.97	1601.89	58.3
TR2_2_HRC	4	8	22.00	24.00	0	28.95	1492.17	66.7
TR1_3_HRC	5	7	20.00	22.00	0	26.94	1603.65	58.3
TR2_3_HRC	4	8	11.00	23.00	0	28.99	1489.63	66.7
TR1_1_HRC_90	5	9	21.00	25.00	2	39.97	1081.16	75.0
TR2_1_HRC_90	4	8	23.00	22.00	0	32.92	1310.90	66.7
TR1_2_HRC_90	6	9	24.00	27.00	3	40.97	1054.02	75.0
TR2_2_HRC_90	3	9	8.00	26.00	0	30.95	1394.77	75.0
TR1_3_HRC_90	6	9	22.00	28.00	3	43.91	983.06	75.0
TR2_3_HRC_90	4	8	11.00	24.00	0	28.94	1492.06	66.7
TR1_1_HRC_95	7	9	31.00	28.00	4	45.91	941.03	75.0
TR2_1_HRC_95	6	9	23.00	24.00	3	40.95	1055.01	75.0
TR1_2_HRC_95	9	7	36.00	21.00	4	43.97	982.34	58.3
TR2_2_HRC_95	5	9	20.00	25.00	2	36.99	1167.60	75.0
TR1_3_HRC_95	9	8	39.00	23.00	5	47.96	900.66	66.7
TR2_3_HRC_95	8	8	28.00	21.00	4	42.99	1005.36	58.3
T1_1_Human	12	0	54.50	0.00	0	54.50	792.66	0.0
T2_1_Human	12	0	50.51	0.00	0	50.51	855.27	0.0
T1_2_Human	12	0	51.52	0.00	0	51.52	838.51	0.0
T2_2_Human	12	0	50.43	0.00	0	50.43	856.63	0.0
T1_3_Human	12	0	50.81	0.00	0	50.81	850.22	0.0
T2_3_Human	12	0	45.19	0.00	0	45.19	955.96	0.0
T1_1_Robot	0	12	0.00	31.95	0	31.95	1352.11	100.0
T2_1_Robot	0	12	0.00	31.96	0	31.96	1351.69	100.0
T1_2_Robot	0	12	0.00	31.94	0	31.94	1352.54	100.0
T2_2_Robot	0	12	0.00	31.92	0	31.92	1353.38	100.0
T1_3_Robot	0	12	0.00	31.97	0	31.97	1351.27	100.0
T2_3_Robot	0	12	0.00	31.95	0	31.95	1352.11	100.0

Lower thresholds yielded higher throughput. The highest UPH of 1728.18 was achieved in trial TR2_1_HRC, where the robot handled all assigned slots without human intervention, completing the task in under 27 seconds. Higher thresholds (e.g., HRC_95) increased low-confidence cases and human workload. The Robot (%) column shows that even with about 75% robot coverage (e.g., TR1_2_HRC_90, TR2_2_HRC_95), task durations stayed under 45 seconds and throughput remained above 1000 UPH, indicating that autonomy is maintained even with stricter confidence filtering.

TABLE III
SUMMARY COMPARISON OF HUMAN, ROBOT, AND HRC SETUPS

Setup	Avg T (s)	Std T	Avg UPH	Std UPH	R (%)
Human	50.49	2.75	858.21	48.84	0.0
Robot	31.95	0.02	1352.18	0.66	100.0
HRC (default)	27.29	1.50	1586.88	88.61	59.72
HRC 90%	36.28	6.12	1219.33	207.74	72.23
HRC 95%	43.13	3.85	1008.67	94.18	68.05

Table III presents a consolidated comparison of inspection performance under five setups: human-only, robot-only, and HRC at three confidence thresholds (default (70%), 90%, and 95%). Metrics include average completion time (Avg T), standard deviation (Std T), average units per hour (Avg UPH), UPH variability (Std UPH), and average robot contribution (R %).

The **HRC default** setup achieved the fastest time (**27.29s**) and highest throughput (**1586.88 UPH**). Human-only was slowest (50.49s, 858.21 UPH) but ensured 100% accuracy. Robot-only was also fast (31.95s) but lacked fallback for uncertain predictions. Increasing the confidence threshold from

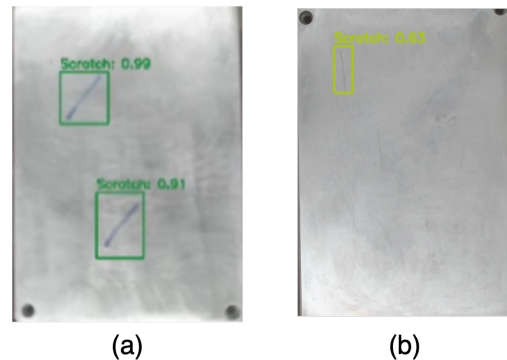


Fig. 6. Example detection results in a real inspection scenario. (a) high-confidence detection, where the model clearly identifies scratches with scores above 0.9, allowing the robot to proceed autonomously. (b) low-confidence detection, where the model expresses uncertainty. In such cases, the system flags the SSD for human verification to avoid potential misdetection or overlooked defects.

70% to 95% raised completion time and reduced throughput, reflecting greater reliance on human intervention. These results show that HRC can balance speed and robustness, with the robot handling most cases and delegating ambiguous ones to the human.

Figure 6 shows example detection results. In (a), scratches are confidently identified (scores 0.90, 0.91), allowing autonomous classification. In contrast, subfigure (b) presents a case where the model produces a low-confidence score of 0.63. Although a defect is detected, the low score suggests uncertainty—either due to subtle features, surface noise, or insufficient defect visibility. To mitigate the risk of misclassi-

fication or missed defects, the system defers the decision to a human operator. This fallback mechanism ensures reliability by integrating human oversight when the model's confidence is below the safety threshold.

D. Discussion

The results show that while autonomous robotic inspection achieves high throughput when confidence scores are high, its reliability declines when defect visibility is subtle or ambiguous. In these cases, human involvement is essential to maintain inspection quality, justifying Human-Robot Collaboration (HRC) as a robust approach in industrial inspection.

HRC is particularly beneficial when the AI model produces low-confidence detections. For instance, in Fig. 6(b), the robot correctly identified a potential scratch but with a confidence score of 0.63, below the threshold. Instead of risking an incorrect decision, the system flagged the case for human validation, safeguarding against misdetection. Trials such as TR1_1_HRC_90 and TR1_1_HRC_95 show that routing low-confidence slots to the human maintains high inspection accuracy with minimal impact on efficiency. Table II shows that even with 2–5 low-confidence slots per trial, completion time remains under 48 seconds and throughput (UPH) exceeds 900 units/hour. Notably, trials with around 75% robot autonomy (e.g., TR2_2_HRC_90 at 1394.77 UPH) demonstrate that effective task distribution enables both speed and reliability.

From Table III, the HRC 70% configuration achieves the best overall performance, with an average completion time of 27.29 seconds and UPH of 1586.88. This setup lets the robot handle confident cases autonomously while involving the human only when needed. In contrast, HRC 95% lowers throughput and increases completion time due to stricter confidence requirements. The robot-only setup is fast but lacks fallback for ambiguous cases, while the human-only baseline is reliable but slower. These findings confirm that HRC is not just a fallback but a necessary feature for dependable industrial inspection. Confidence-aware delegation balances throughput and robustness.

V. CONCLUSION

This paper presented a human-robot collaborative inspection system for detecting surface defects on metallic SSD cases, combining an autonomous robotic arm, an AI-based vision model, and a fallback mechanism where low-confidence predictions are verified by a human. A Behavior Tree framework governs task execution.

Experiments across multiple trials and confidence thresholds show that collaboration improves robustness with minimal impact on throughput. The robot achieved over 1600 UPH under confident conditions, while human validation of uncertain cases maintained UPH above 900. These results confirm that confidence-aware task sharing ensures both speed and reliability. The approach demonstrates the potential of combining AI-driven automation with selective human oversight to meet stringent industrial quality control requirements. It also provides a scalable framework adaptable to other inspection tasks involving reflective materials or subtle defect patterns.

Limitations include the absence of quantitative accuracy metrics and subjective workload assessment, and tests conducted only in a lab environment. Future work will extend evaluation to factory settings, apply model calibration to reduce low-confidence cases, benchmark accuracy against ground truth, and conduct user studies (e.g., NASA-TLX) to assess cognitive load and trust. We will also explore ontology-based reasoning [14] for task introspection and explanation generation [15].

ACKNOWLEDGMENT

This work was partially supported by the National Science and Technology Council, Taiwan, under grant number NSTC 114-2927-I-A49-504; and by the Spanish National Research Council (CSIC) under grant number CSIC-BILAT23120.

REFERENCES

- [1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human-robot collaboration," *Autonomous Robots*, vol. 42, pp. 957–975, Jun 2018.
- [2] S. Izquierdo-Badiola, G. Canal, C. Rizzo, and G. Alenyà, "Improved task planning through failure anticipation in human-robot collaboration," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7875–7880, 2022.
- [3] A. Olivares-Alarcos, S. Foix, and G. Alenyà, "Time-to-contact for robot safety stop in close collaborative tasks," 2023.
- [4] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14298–14308, 2022.
- [5] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10778–10787, 2020.
- [6] S. D. Sanjaya and H. I. Lin, "Automated inspection of flywheel metallic objects using deep learning-based defect detection," in *2024 International Automatic Control Conference (CACS)*, pp. 1–6, IEEE, Oct 2024.
- [7] H. I. Lin, S. D. Sanjaya, and L. Rupa, "Defect detection on metal laptop cases by up-sampling and down-sampling method," in *2024 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–6, IEEE, Sep 2024.
- [8] A. Castro, F. Silva, and V. Santos, "Trends of human-robot collaboration in industry contexts: Handover, learning, and metrics," *Sensors*, vol. 21, no. 12, p. 4113, 2021.
- [9] D. Rodríguez-Guerra, G. Sorrosal, I. Cabanes, and C. Calleja, "Human-robot interaction review: Challenges and solutions for modern industrial environments," *IEEE Access*, vol. 9, pp. 108557–108578, 2021.
- [10] C. Gäbert, S. Kaden, and U. Thomas, "Generation of human-like arm motions using sampling-based motion planning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2534–2541, 2021.
- [11] T. A. Bach, A. Khan, H. Hallock, G. Beltrão, and S. Sousa, "A systematic literature review of user trust in ai-enabled systems: An hci perspective," *International Journal of Human-Computer Interaction*, vol. 40, pp. 1251–1266, Mar 2024.
- [12] E. Prati, V. Villani, F. Grandi, M. Peruzzini, and L. Sabatini, "Use of interaction design methodologies for human-robot collaboration in industrial scenarios," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3126–3138, 2022.
- [13] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "Simplenet: A simple network for image anomaly detection and localization," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20402–20411, 2023.
- [14] A. Olivares-Alarcos, S. Foix, S. Borgo, and Guillem Alenyà, "Ocr – an ontology for collaborative robotics and adaptation," *Computers in Industry*, vol. 138, p. 103627, 2022.
- [15] A. Olivares-Alarcos, A. Andriella, S. Foix, and G. Alenyà, "Robot explanatory narratives of collaborative and adaptive experiences," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11964–11971, 2023.