*Proceedings of the International Conference on Industrial Engineering and Operations Management*

Publisher: IEOM Society International, USA DOI: 10.46254/NA10.20250325

Published: June 17, 2025

# Intelligent Manufacturing Defect Visual Inspection with Temporal Samples: An Approach of Accumulated Knowledge Graph Reasoning with YOLO Stream-based Detection

**Yiyun Fei and Mulang Song**
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA, USA
yfei38@gatech.edu

**Shu Wang and Xinping Deng**
TE Connectivity, Middletown,
Pennsylvania, USA

**Roger J. Jiao**
School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA, USA

## Abstract

In modern manufacturing, defect inspection plays a crucial role in ensuring product quality and compliance with specifications. However, traditional manual inspection methods are increasingly inadequate due to human error, subjectivity, and scalability limitations. This paper presents an intelligent system for automated detection and reasoning of time-varying samples in inspection videos. The proposed system comprises five interconnected modules: object detection, temporal aggregation, case retrieval, semantic labeling, and case retention. A YOLOv8-based model is employed to detect individual instances across video frames. Detections are then grouped using HDBSCAN clustering based on spatial and temporal continuity to form distinct defect cases. These cases are compared against a structured knowledge graph containing historical defect data using similarity measures. A semantic labeling process further refines the interpretation by leveraging a large language model to generate defect pattern descriptions, identify likely causes, and recommend appropriate actions. A real-world case study involving relay gross leak testing demonstrates the system's ability to achieve high detection accuracy, effectively classify defect patterns, and generate interpretable and actionable outputs. The system supports continuous learning by retaining new defect cases and their labeled interpretations in the knowledge graph. Overall, the proposed framework enhances consistency, traceability, and decision-making in visual inspection tasks, paving the way for scalable and intelligent quality control in manufacturing environments.

## Keywords
Intelligent Inspection System, Object Detection, Knowledge Graph, Case-Bd Reasoning, Large Language Model

## 1. Introduction
Defect inspection is a fundamental component of quality assurance in manufacturing. It ensures products meet specifications and quality standards before being delivered to customers. A typical inspection cycle involves several key stages: I. preparation, which includes defining the scope and defect criteria and identifying the reliable detection method; II. conducting the inspection to locate and/or classify defects; III. analyzing the defect features and comparing them against the defined criteria; IV. deciding on whether to accept or reject the product based on the findings. However, traditional manual inspection methods are increasingly proving inadequate in modern manufacturing environments. Human inspectors are prone to fatigue, subjectivity, and inconsistency, especially in repetitive or high-speed inspection tasks. Studies estimate that manual visual inspection is only about 80% effective (Smith 1993), which poses a significant risk to product quality and customer satisfaction. As production scales up and demands for precision and speed grow, these limitations become more pronounced. To address these challenges, the industry is undergoing a broader transition from manual to automated inspection systems under the advent of Industry 4.0. As part of the broader movement toward smart manufacturing, automated inspection plays a critical role in elevating product quality and operational efficiency. Automated inspection encompasses a wide range of technologies—including visual inspection, ultrasonic testing, X-ray, and other sensor-based methods.

<u>(1) Visual Defect Inspection</u>: Automated visual inspection has become one of the most widely adopted solutions due to advances in computer vision and artificial intelligence (Lerones and Fernández 2005; Aggour et al. 2019; Sundaram and Zeid 2023). These systems use image data captured from industrial cameras to identify, localize, and evaluate visual defects such as scratches, cracks, and discolorations. The adoption of deep learning models has significantly improved detection performance in terms of accuracy, speed, and generalizability. Vision detection systems not only eliminate human variability but also enable real-time defect detection.

<u>(2) Temporal Samples:</u> In many manufacturing applications, defects show indirectly through time-varying visual indicators. These indicators, or temporal samples, change their appearance across frames or time stamps. Examples include vapor trails, color diffusion, or bubble formations in fluid-based testing. One representative application is the detection of microleaks in high-reliability electrical components, such as relays and sealed connectors. These components often require hermetic sealing to prevent long-term degradation caused by moisture, dust, corrosive gases, or other environmental contaminants. Even the smallest leak can compromise performance or safety, especially in sectors like aerospace, automotive, and medical devices. Gross leak testing is commonly used to verify the airtightness of these components, where air leaks manifest as visible bubbles when the part is submerged in a liquid.

<u>(3) Intelligent Defect Reasoning:</u> While detecting the presence of temporal samples is important, it is not sufficient for effective defect management. Many visual cues observed over time are indirect indicators of underlying defects. Therefore, beyond identifying individual instances, the system must reason for their relationships: determining whether different observations stem from the same root cause, how they evolve over time, and what patterns they form. This involves grouping related temporal events, analyzing their spatiotemporal characteristics, and interpreting their contextual significance. Traditionally, such interpretations rely heavily on human expertise, which limits consistency, scalability, and the ability to learn from past cases.

To overcome these limitations, an intelligent inspection system capable of automated detection, defect reasoning, and knowledge retention is needed. This paper proposes a novel intelligent system that integrates real-time object detection, temporal pattern analysis, semantic reasoning, and a case-based knowledge graph to handle time-varying defect indicators. The system aims to improve defect traceability, reduce inspection effort, and support data-driven decision-making. By identifying meaningful patterns across temporal samples and linking them to potential root causes, it not only enhances inspection accuracy but also facilitates continuous improvement. Ultimately, the system helps manufacturers gain a deeper understanding of underlying quality issues.

## 2. Related Work on Emerging Technologies for Intelligent Defect Inspection
### 2.1 Visual Defect Inspection
Computer vision and deep learning have been increasingly applied to visual inspection tasks. For defect detection, algorithms based on Convolutional Neural Networks (CNNs), such as Faster R-CNN, Mask R-CNN, Feature Pyramid Networks (FPN), ResNet, and YOLO, have seen significant development and are widely adopted to address various detection challenges (Zhang et al. 2021; Ren et al. 2022; Hussain 2023). Among these, YOLO has emerged as particularly effective for real-time video-based defect detection due to its balance of accuracy and computational

efficiency (Lavanya and Pande 2023). The rapid evolution of the YOLO architecture—evident in successive versions such as YOLOv4 through YOLOv11—further underscores its potential and adaptability for industrial inspection applications (Murthy et al. 2022; Mao and Hong 2025).

## 2.2 Data Clustering

Data clustering is an unsupervised machine learning technique that organizes data points into groups, or clusters, such that points within a cluster exhibit higher similarity to one another than to those in different clusters. Clustering algorithms are broadly categorized by their approach: partition-based methods like K-Means, density-based methods such as DBSCAN, and hierarchical approaches like agglomerative clustering (Ahmad 2015). HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) represents a hybrid technique that combines density-based and hierarchical strategies to overcome limitations of DBSCAN. Specifically, HDBSCAN improves clustering performance by adapting to local density variations and automatically determining the number of clusters, while effectively identifying and excluding noise points (Campello et al. 2013; Bushra and Yi 2021). This makes HDBSCAN a more flexible and robust alternative to traditional methods such as K-Means and DBSCAN, especially when working with data exhibiting variable density.

## 2.3 Case-Based Reasoning with Knowledge Graph and Large Language Models

Case-Based Reasoning (CBR) is an artificial intelligence problem-solving methodology that solves new problems by reusing and revising solutions from similar past cases. It has been widely used in manufacturing related applications, such as equipment fault diagnosis, product design, and process design (Chen et al. 2022; Li et al. 2022; Khosravani and Nasiri 2020). As the limitations of standalone language models become more evident in tasks requiring contextual reasoning, an increasing number of studies have explored the integration of Case-Based Reasoning with Large Language Models to improve accuracy, generalizability, and reasoning performance on task-specific challenges (Wilkerson and Leake 2024; Sourati et al. 2023; Yang 2024). In parallel, the rising demand for semantically rich and interpretable reasoning has also driven recent efforts to integrate CBR with Knowledge Graphs, particularly to enhance the capabilities of recommendation systems and knowledge-based question answering (Ge et al. 2024; Li et al. 2024) Recent studies have also applied these hybrid reasoning frameworks to visual defect inspection, highlighting their potential in intelligent algorithm selection and defect mitigation in manufacturing settings (Wang et al. 2024; Wang et al. 2024).

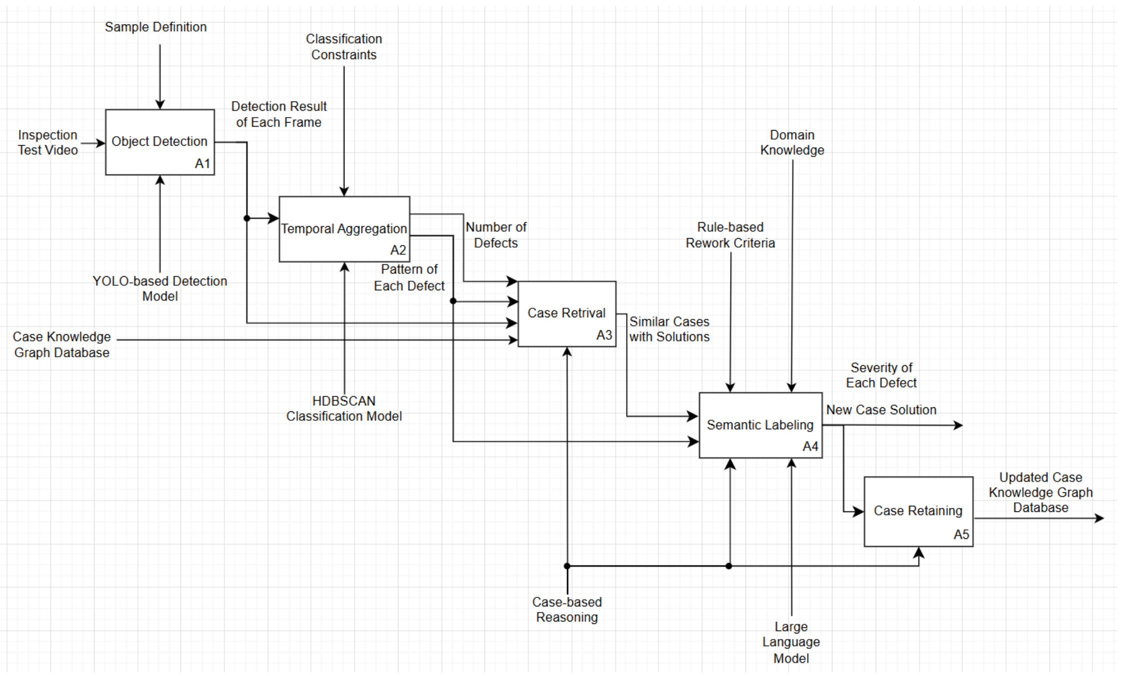


Figure 1. System Architecture of the Intelligent Visual Inspection System for Temporal Sample-Based Defect Detection and Reasoning

## 3. System Analysis and Design

The IDEF0 diagram presented in Figure 1 models the workflow of the proposed intelligent defect detection and reasoning system for manufacturing inspection involving temporal samples. Each box in the IDEF0 framework represents a major functional activity, governed by four essential components: Inputs, Controls, Mechanisms, and Outputs. The system integrates computer vision, spatiotemporal analysis, and Case-Based Reasoning through five interconnected modules (A1–A5), enabling automated interpretation and recommendation of time-varying defect indicators in inspection videos. The pipeline begins with video-based detection of temporal samples and progresses through pattern aggregation, retrieval of similar cases, semantic labeling, and ultimately the retention of new knowledge. Each module contributes to a distinct stage in the defect analysis workflow: identifying individual sample instances (A1), consolidating them into temporally and spatially coherent patterns (A2), retrieving comparable historical cases from a knowledge graph (A3), generating recommended solutions using semantic reasoning (A4), and storing the new cases for future reuse (A5). The system is designed to leverage domain-specific AI models and a structured graph-based memory to automate traditional manual processes, thereby improving detection consistency, enhancing interpretability, and minimizing the need for human intervention.

## 4. Integrated Framework for Temporal Visual Defect Analysis

### 4.1 Stream-based Detection of Temporal Samples via YOLO

The system begins with an input inspection video, from which individual frames are extracted and analyzed using a YOLO-based object detection model trained to identify all visible sample instances. Figure 2 illustrates the generalized architecture of the YOLO model, which consists of three primary components: the Backbone, Neck, and Head. The Backbone is responsible for extracting feature maps from the input image, typically employing convolutional layers, Cross Stage Partial (CSP) connections, and residual blocks to enhance feature representation. The Neck fuses features across multiple scales using modules such as the Feature Pyramid Network (FPN), thus the model detects objects of varying sizes and appearances. The Head of the model performs the final prediction, outputting bounding box coordinates, class probabilities, and scores for each detected object. The outputs generated by this module include metadata for each detected instance: spatial coordinates (x and y center), predicted class, bounding box dimensions, and frame-level timestamp. These results serve as the foundational input for subsequent stages in the system pipeline.

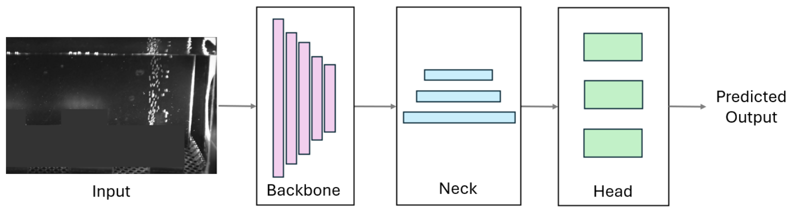


Figure 2. Generalized YOLO Structure Enabling Visual Temporal Sample Detection

### 4.2 Temporal Aggregation of Framewise Detections into Defect Patterns

Temporal aggregation plays a critical role in converting frame-level detections into temporally and spatially coherent defect cases. This module bridges raw detection outputs and structured defect patterns suitable for downstream reasoning tasks. Its primary function is to group detected instances—often scattered across consecutive frames—into distinct clusters that represent higher-level defect cases. Aggregation is guided by domain-specific logic that assumes events originating from the same root cause tend to occur within limited spatial regions and over constrained time intervals. To implement this grouping, a clustering algorithm HDBSCAN, an unsupervised density-based clustering algorithm that does not require a predefined number of clusters, is applied based on selected spatial features. This aggregation step outputs structured representations enriched with features such as average size, detection duration, and frequency—attributes that are later used in similarity-based case retrieval.

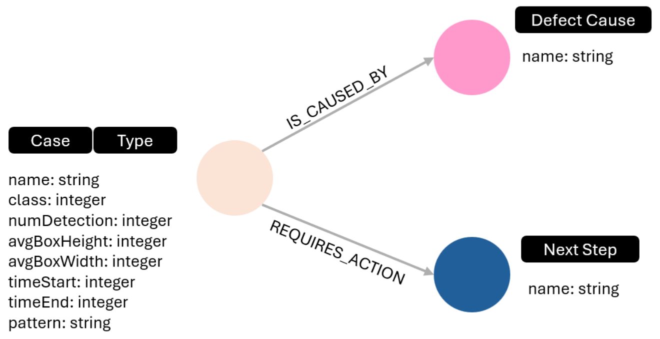


Figure 3. Data Model of the Knowledge Graph for Storing Defect Cases

### 4.3 Retrieval of Similar Defect Case Using a Knowledge Graph Framework

To support Case-Based Reasoning, the system employs a case knowledge graph database, which serves as a dynamic repository of historical defect cases. The data model of the graph database is shown in Figure 3. This model consists of three node types: Case, Defect Cause, and Next Step. Each Case node stores attributes such as object class, detection count, average size, time span, and a pattern description. Cases are linked to their causes via the IS_CAUSED_BY relationship and to recommended actions via the REQUIRES_ACTION relationship. Case retrieval compares the current defect case (derived from A2) against the historical cases using local and global similarity measures. The top K most similar cases are retrieved to provide contextual knowledge and guide defect interpretation and decision-making.

### 4.4 Semantic Labeling for Root Cause and Action Inference

Semantic Labeling integrates rule-based reasoning and a large language model with domain-specific knowledge to interpret retrieved cases and label the new defect with a recommended solution. Solutions are represented through three components. The first is the linked DefectCause node. The second is the linked NextStep node. The third is a property of the Case node called Pattern, which stores a descriptive summary of the defect pattern generated by a large language model. The semantic labeling process first generates the Pattern description for the new case, then compares it with the patterns of the top K retrieved cases. Based on this comparison, the system determines the most likely defect cause and the appropriate next step. This step embodies the reuse and revision phase of Case-Based Reasoning and reflects expert criteria in defect interpretation and disposition. Factors such as severity level, rework eligibility, or scrappage decision are determined here.

### 4.5 Case Retaining for Knowledge Graph Expansion

Once the newly generated case—along with its semantically labeled solutions—is validated, it is added to the case knowledge graph for future use. By retaining each case along with its associated cause, action, and descriptive pattern, the system expands its knowledge base over time. This process ensures continuous learning and supports system evolution as new types of defects are encountered and addressed.

## 5. Case Study of Visual Defect Inspection in Electrical Relay Manufacturing

To evaluate the feasibility of the proposed system, a case study was conducted on a real-world application involving bubble detection during gross leak testing of electrical relays. Relay gross leak testing is a critical quality inspection process designed to ensure the integrity and reliability of relay devices, which are essential components in electrical circuits and systems. The current testing process requires an operator to fully submerge a tray of relays into a fluid medium heated in a dedicated testing equipment and monitor it for thirty seconds to ensure no air bubbles emerge from any of the relays during this period. This is a labor-intensive process and is prone to human error. The consequence of not catching defective relays increases the material cost from the subsequent manufacturing processes and decreases customer satisfaction if they are sent to the customer. In order to mitigate these risks and enhance the efficiency and accuracy of the testing process, an automated inspection solution is proposed. The goal of the project is to automatically detect bubbles without human intervention. This section demonstrates how the system components described in the previous chapter are integrated and applied to process raw inspection video, identify and classify defects, and generate actionable insights based on historical cases.

## 5.1 YOLO-Based Bubble Detection

For this case study, three distinct types, or classes, of bubbles were defined based on their visual appearance, as illustrated in Figure 4.

- **Type 1 (purple):** Large, isolated bubbles that emerge one at a time. These bubbles are characterized by two bright symmetrical reflections on either side and a dark central region.
- **Type 2 (yellow):** A continuous stream of medium-sized bubbles, where each bright reflection corresponds to a separate bubble. These streams tend to ascend toward the surface of the fluid.
- **Type 3 (red):** A narrow, vertical column of small bubbles forming a fine stream. Unlike Type 2, this stream tends to dissipate midway through the fluid and does not reach the surface. It appears as a single column of dot-like formations and often indicates subtle or dispersed leakage behavior.

(1) Dataset Composition and Annotation: The object detection task was implemented using the YOLOv8 model, chosen for its performance on speed and accuracy. The dataset includes more than 6,000 labeled images, of which 2000 augmented images. Among these images, approximately 4,500 images (~70%) are used for training, 1000 (~15%) for validation, and another 1000 (~15%) for testing. Converting the number of images to the number of annotations in each class, there are 10,974 single bubbles, 6,864 large stream of bubbles, and 3,432 small stream of bubbles. The imbalance in the classes reflects the nature of the bubble and their actual probabilities of occurrence during product inspection.

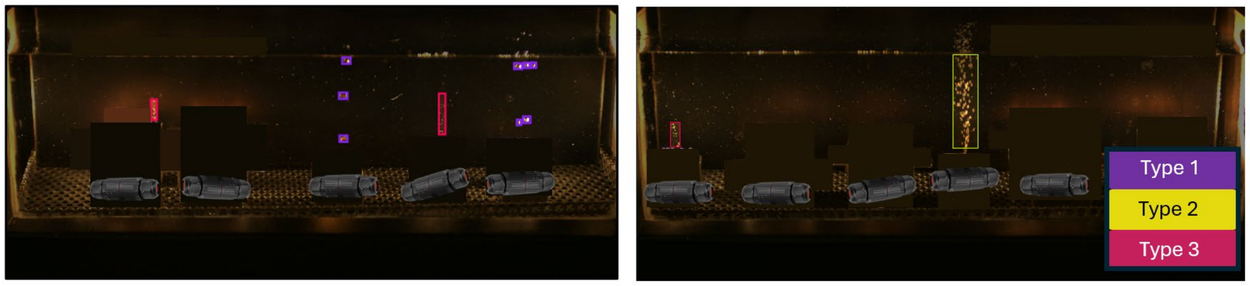


Figure 4. Examples of Three Bubble Patterns Observed During Relay Leak Testing (Original parts are visually masked and substituted with publicly available samples for IP protection)

(2) Model Configuration and Evaluation: To obtain the model with best performance, different sizes of the YOLOv8 models were tested along with varying image color settings and preprocessing operations. As summarized in Table 1, four model configurations (v1–v4) were evaluated. All models applied common preprocessing steps including auto-orientation and static cropping. Model v1 used grayscale images with a smaller "nano" model size, while models v2–v4 employed the "medium" size. Models v3 and v4 additionally incorporated a tiling strategy that split each frame into one row and two columns to better capture bubble features in smaller regions. Model v4 also introduced color images, aiming to leverage chromatic cues for enhanced detection accuracy.

Table 1. Model Configuration and Performance Comparison for Bubble Detection

| | Model v1 | Model v2 | Model v3 | Model v4 |
|---|---|---|---|---|
| Image Color | Grayscale | Grayscale | Grayscale | Color |
| Model Size | Nano | Medium | Medium | Medium |
| Image Preprocessing | 1. Auto-Orient<br>2. Static Crop<br>3. Resize: 640 | 1. Auto-Orient<br>1. Static Crop<br>2. Resize: 640 | Auto-Orient<br>Static Crop<br>Tile: 1 row x 2 col.<br>Resize: 640 | 1. Auto-Orient<br>2. Static Crop<br>3. Tile: 1 row x 2 col.<br>4. Resize: 640 |
| Precision (Val/Test) | 91% / 89% | 93% / 91% | 93% / 92% | 93% / 93% |
| Recall (Val/Test) | 88% / 78% | 89% / 86% | 96% / 92% | 96% / 90% |

(3) Performance Metrics: To assess detection performance, three key metrics were examined: precision, recall, and the F1 score. Precision measures the model's ability to avoid false positives, i.e., how many of the predicted bubbles are actually correct. It is defined as the ratio of true positive detections to all positive predictions. Recall, on the other

hand, quantifies the model's ability to detect all relevant instances. It is defined as the ratio of true positives to the total number of actual positives in the dataset. The precision and recall values were based on bubble instance instead of the number of defectives. The F1 score serves as the harmonic mean of precision and recall. Among the configurations tested, Models v3 and v4 achieved the highest recall scores (96% on validation, and 92% and 90% on the test set, respectively), indicating strong sensitivity in detecting bubble types. Precision across v2 to v4 remained consistently high, with v4 reaching a peak of 93% on both validation and test datasets. Despite the slightly higher performance of model v4, Model v3 was ultimately selected for deployment due to its comparable accuracy while operating on grayscale images, which incur lower computational costs. Figure 5 (a,b,c,d) shows the F1-confidencec curve and the precision-recall curve for both of the validation and test sets.

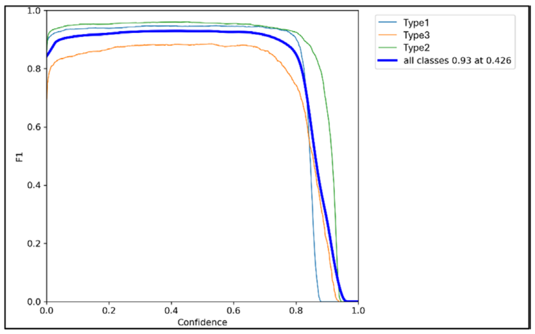


Figure 5 (a) F1-Confidence Curve of Validation Set

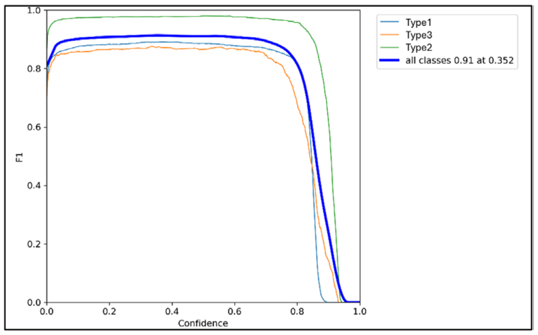


Figure 5 (b) F1-Confidence Curve of Test Set

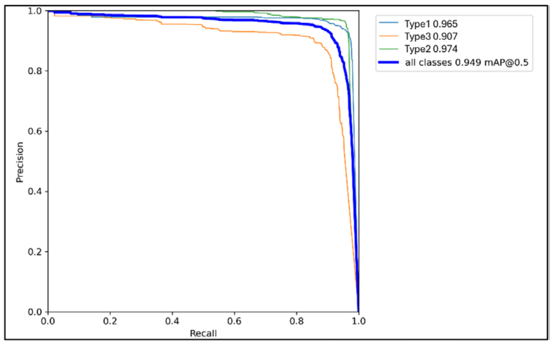


Figure 5 (c) Precision-Recall Curve of Validation Set

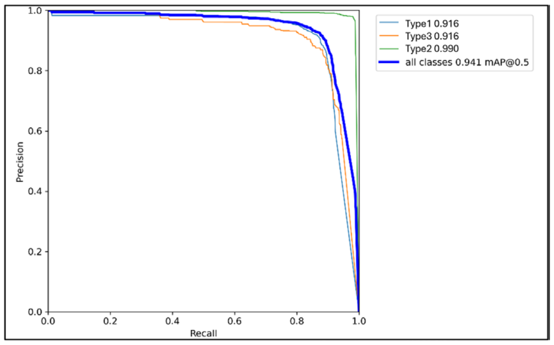


Figure 5 (d) Precision-Recall Curve of Test Set

<u>(4) Prediction Result:</u> The model outputs the predicted bounding box location, bubble class, and associated confidence score for each detected instance. Figure 6 illustrates an example of the visualized detection results from a single frame. The model processes each frame of the video sequentially and records information for every detected bubble instance. This detection data—including position, class, and confidence, will then be passed to the next stage for further analysis to determine the number of distinct defects present in the video and their bubble patterns.

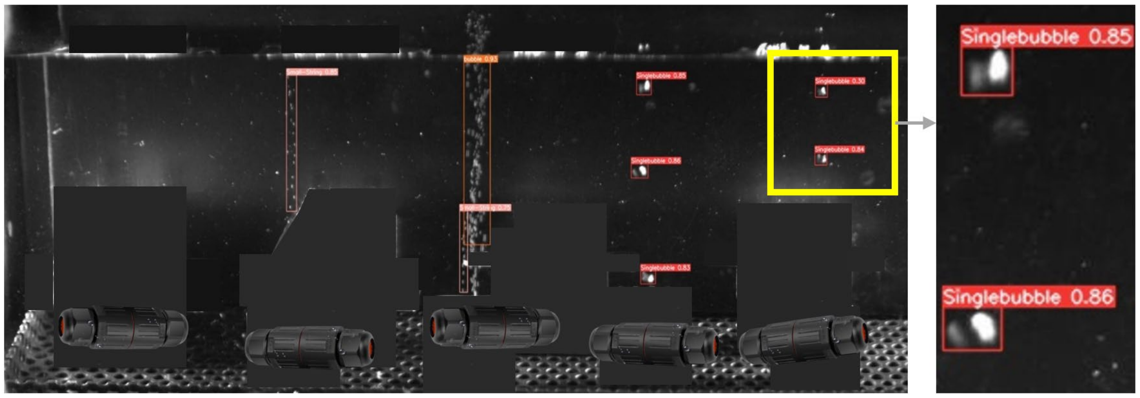


Figure 6. Example of Detection Result:
There are five type 1 bubbles (on the right in red), one type 2 bubbles (in the middle in orange), and two type 3 bubbles (in the middle and on the left in pink). Each instance has a confidence score on the top of the bounding box (Original parts are visually masked and substituted with publicly available samples for IP protection)

## 5.2 Temporal Aggregation and Defect Classification

After object detection is performed on individual frames, the next step involves temporal aggregation and defect classification, where detected bubble instances are grouped into higher-level defect patterns based on their spatial and temporal continuity. This process enables the system to identify whether multiple bubble detections across consecutive frames correspond to a single defect event or represent separate occurrences. Figure 7 shows a 2D visualization of all bubble detections grouped by class, with Type 1 (large single bubbles), Type 2 (medium streams), and Type 3 (small streams) indicated in red, blue, and green, respectively. Each colored box represents a detected bubble, with their horizontal and vertical positions reflecting the x- and y-coordinates in image space. Notably, the distinct horizontal clustering of bubble types reflects their spatial localization, while vertical elongation—particularly for Types 2 and 3—highlights the trajectory consistency across time. To further illustrate the outcome of aggregation, Figure 8 presents a 3D scatter plot of bubble center trajectories over time. The x- and y-axes represent the spatial center of detected bubbles in pixel coordinates, while the z-axis indicates the detection order, corresponding to the frame index. The color gradient encodes detection time, offering insight into the motion patterns of bubbles throughout the video. Vertical clustering of points with gradual color changes suggests a temporally consistent bubble stream, whereas isolated points may indicate sporadic single-bubble events or difference in the types of bubble. By analyzing these spatiotemporal patterns, the system classifies bubble sequences into distinct clusters. Each cluster represents one defect case. Aggregation criteria include thresholds on spatial proximity and detection frequency. This step is essential for reducing false positives and avoiding duplicate defects across frames.

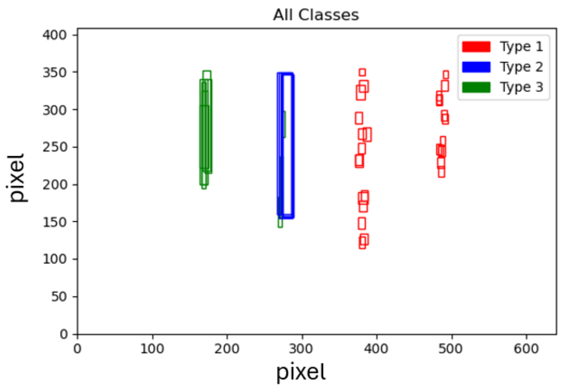


Figure 7. 2D Aggregated Detection Result

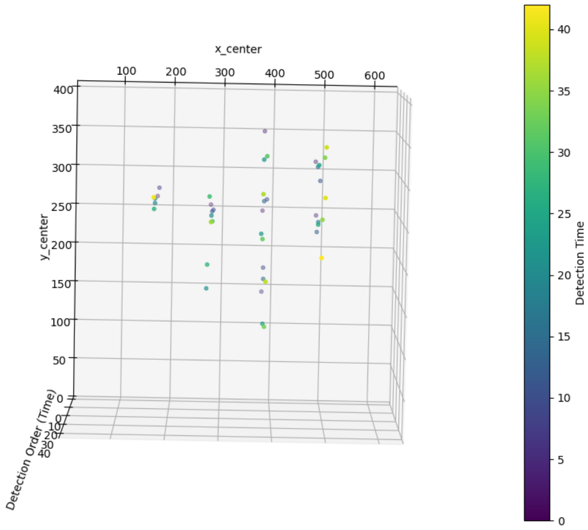


Figure 8. 3D Bubble Center Trajectory over Time

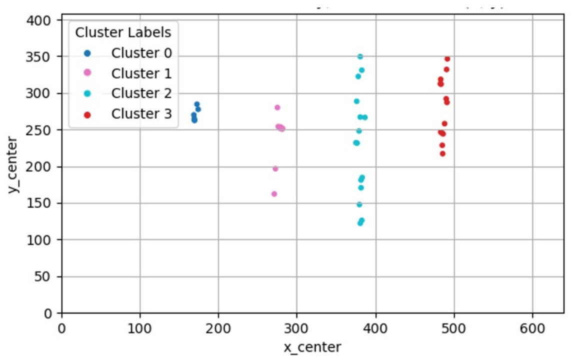
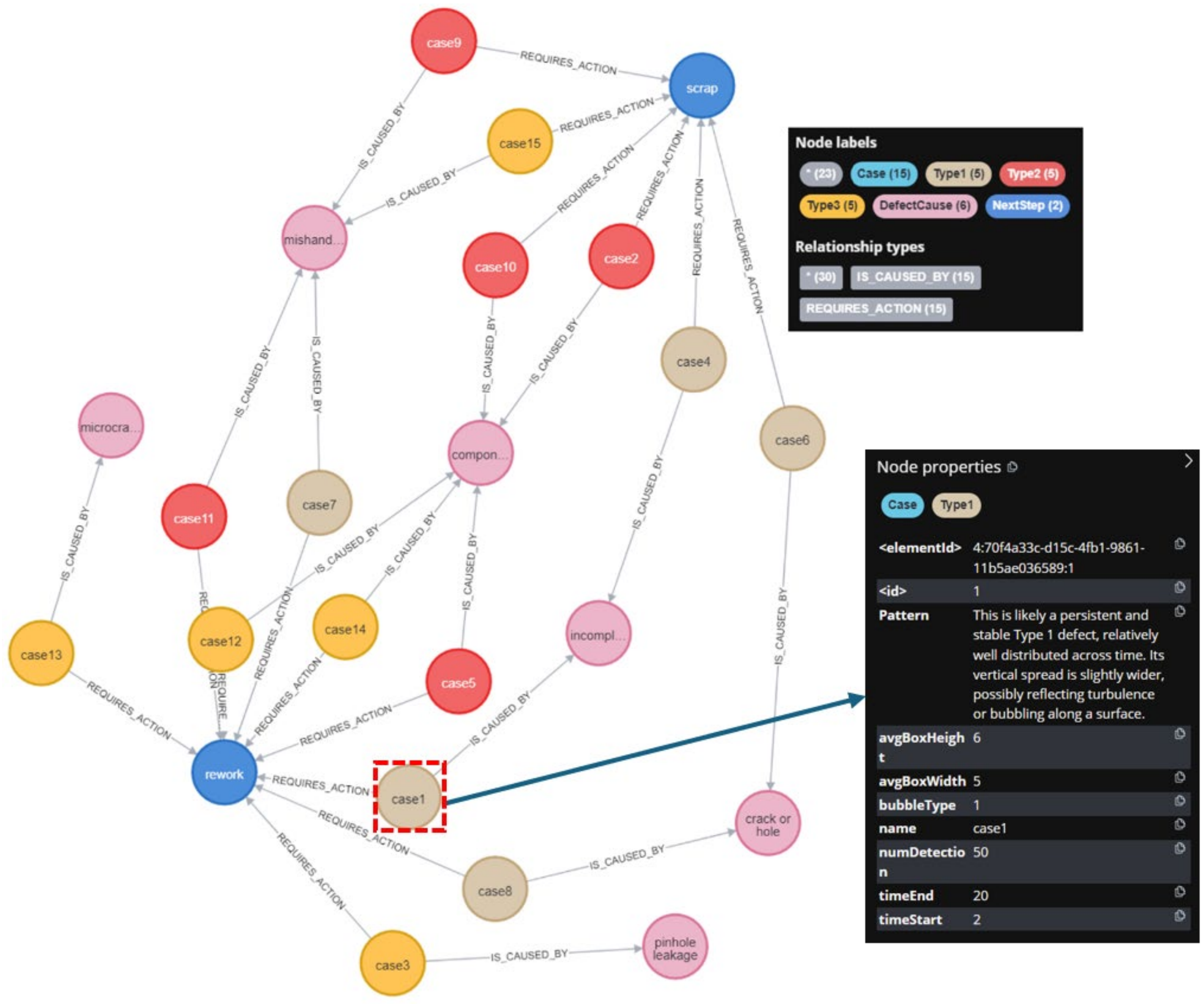


Figure 9. Clustering Result of the Detected Bubbles Using HDBSCAN

As shown in Figure 9, the clustering results were generated using HDBSCAN. In this implementation, clustering was based solely on the x-center coordinate of each detected bubble, reflecting the assumption that bubbles associated with the same defect typically occur within a narrow horizontal region. The y-coordinate is used only for visualization purposes. The minimum cluster size required to define a dense region that qualifies as a cluster was set to 5, thereby suppressing false detections and noise. HDBSCAN successfully identified four distinct clusters, each assigned a unique color in the plot. The clear horizontal separation among clusters indicates strong spatial locality of defect events. The effectiveness of HDBSCAN lies in its ability to automatically identify the optimal number of clusters based on local density, while also recognizing outliers as noise points when necessary. The clustering results form the basis for downstream pattern analysis and case retrieval.

### 5.3 Case-Based Reasoning with Knowledge Graph and Large Language Model



Node properties

Case Type1

| | |
|---|---|
| <elementId> | 4:70f4a33c-d15c-4fb1-9861-11b5ae036589:1 |
| <id> | 1 |
| Pattern | This is likely a persistent and stable Type 1 defect, relatively well distributed across time. Its vertical spread is slightly wider, possibly reflecting turbulence or bubbling along a surface. |
| avgBoxHeight | 6 |
| avgBoxWidth | 5 |
| bubbleType | 1 |
| name | case1 |
| numDetection | 50 |
| timeEnd | 20 |
| timeStart | 2 |

Figure 10. Case Knowledge Graph Database Representing Structured Bubble Cases, Causal Relationships, and Recommended Actions for Semantic Reasoning and Retrieval

To support defect interpretation and continuous improvement, the system incorporates a knowledge graph-based case reasoning framework. As shown in Figure 10, the graph comprises multiple node types, each representing a critical element of the defect analysis process. The core components are the Case nodes, which store structured information about individual defect cases. Each Case node is additionally labeled with the corresponding bubble type (Type 1 in brown, Type 2 in yellow, or Type 3 in red) to enhance visual distinction and facilitate interpretation. Each case node contains key attributes such as average bounding box size, bubble count, and detection duration, as shown in the node properties of an example case in the lower right corner of Figure 10. In addition to case information, the graph includes DefectCause nodes (pink), which store potential root causes of the observed defect patterns. These are linked to Case nodes through the IS_CAUSED_BY relationship. For instance, *case5* is linked to the cause "component misalignment," suggesting an association between that bubble pattern and a specific underlying defect mechanism. Further downstream, the graph also includes NextStep nodes (blue), representing possible disposition actions for each defect, such as rework or scrap. The REQUIRES_ACTION relationship connects each Case node to its corresponding decision node. For example, *case9* and *case15* are both linked to the action node *scrap*, whereas *case12* and *case14* are directed toward *rework*.

For case retrieval, Table 2 presents the attributes used for similarity calculation, along with their corresponding weights and value ranges. Among these, Bubble Type Matching is a discrete attribute. The local similarity function for discrete attribute is defined in Equation 1, where $a_i$ denotes the value of the new case and $b_i$ denotes the value of the old case. If the bubble types match, the similarity returns 1; otherwise, it returns 0. The remaining four attributes are continuous, and their local similarity is calculated using the function shown in Equation 2. To accommodate both discrete and numerical attributes in the similarity calculation, the global similarity formula in Equation 3 applies different local similarity functions based on attribute type. In this formula, p represents the total number of attributes, and w denotes the assigned weight of each attribute. The semantic labeling process begins by generating a pattern description for each defect case, capturing both the temporal and spatial characteristics of bubble behavior. Based on the generated pattern, the system then determines the most likely defect cause and recommends an appropriate next step. The results are summarized in Table 3. Each row in the table corresponds to a clustered defect case, with the Pattern column providing a high-level, natural language summary produced by a large language model. By combining pattern interpretation with rule-based and similarity-informed reasoning, the semantic labeling module produces both descriptive and actionable outputs, closing the loop between detection and decision support.

Table 2. Attribute Definitions with Severity Weights and Value Ranges

| Attribute | Severity Weight $w_i$ | Range |
|---|---|---|
| Bubble Type Matching | 10 | $x_1 \in \{0,1\}$ |
| Number of bubble instances | 1 | $x_2 \in [0,100]$ |
| Average Box Width | 0.5 | $x_3 \in [0,50]$ |
| Average Box Height | 1 | $x_4 \in [0,250]$ |
| Defect Duration | 5 | $x_5 \in [0,30]$ |

$$(a_1, b_1) \ = \{1, \ if \ a_1 = b_1 \ \ 0, \ if \ a_1 \neq b_1 \tag{1}$$

$$(a_1, b_1) \ = 1 - \frac{|a_i - b_i|}{range}, \ \ i = 2,3,4,5 \tag{2}$$

$$(A, B) \ = \frac{1}{\Sigma_{i=1}^{p} w_i} \sum_{i=1}^{p} \ \ w_i(a_i \cdot b_i) \tag{3}$$

Table 3. Example – Solutions of the Four New Cases

| Cluster | Pattern | Defect Cause | Next Step |
|---|---|---|---|
| 0 | Cluster 0 is pure Type 3, located at the leftmost position. It appears early in time, which may suggest a recurring early-stage defect at a fixed physical location. This is a stable, consistent leak type. | Pinhole Leakage | Rework |
| 1 | This cluster is a mixture of two bubble types, which is significant: <br>● Early detections at this location are Type 3, then shift to Type 2 <br>● This could imply a transition of defective behavior over time, such as worsening leaks or a physical condition that evolves (e.g., pressure increases causing new bubble behavior) <br>This cluster deserves attention due to its type of shift over time. | Component Misalignment | Rework |
| 2 | Cluster 2 corresponds to a pure Type 1 defect, relatively well distributed across time. Its vertical spread is slightly wider, possibly reflecting turbulence or bubbling along a surface. <br>This is likely a persistent and stable Type 1 defect. | Incomplete Weld Seam | Scrap |
| 3 | Another pure Type 1 cluster, but it tends to occur later in the detection sequence. That suggests this defect might emerge or worsen over time, perhaps due to pressure buildup or heating over time. | Mishandled Part | Rework |

## 5. Conclusions

This paper presents an intelligent system for automated detection and reasoning of temporal defect patterns in manufacturing inspection videos. The system comprises five interconnected modules—object detection, temporal aggregation, case retrieval, semantic labeling, and case retaining. At the core of the system is a YOLO-based object detection module, which efficiently identifies individual instances across inspection frames. These frame-level detections are then transformed into coherent spatiotemporal patterns through a temporal aggregation step using HDBSCAN. These structured patterns are subsequently matched against a knowledge graph of historical cases through a Case-Based Reasoning framework. This enables contextual interpretation by retrieving similar past cases. Further, semantic labeling integrates pattern descriptions generated by a large language model with rule-based logic to infer the most probable defect causes and recommend next steps. By retaining each newly analyzed and labeled case in the knowledge graph, the system promotes continuous learning and adaptation to emerging defect types. The case study demonstrates the feasibility and effectiveness of the proposed framework. Future work will extend the system's generalizability to other visual inspection scenarios and explore additional integration with upstream manufacturing data to further improve diagnostic precision and actionable insight generation.

## References

Ahmad, P.H., Performance evaluation of clustering algorithm using different datasets, *Journal of Information Engineering and Applications*, vol. 5, no. 1, 2015. Available: http://www.iiste.org

Aggour, K.S., Gupta, V.K., Ruscitto, D. and others, Artificial intelligence/machine learning in manufacturing and inspection: A GE perspective, *MRS Bulletin*, vol. 44, no. 7, pp. 545–558, 2019.

Bushra, A.A. and Yi, G., Comparative analysis review of pioneering DBSCAN and successive density-based clustering algorithms, *IEEE Access*, vol. 9, pp. 87918–87935, 2021.

Campello, R.J.G.B., Moulavi, D. and Sander, J., Density-based clustering based on hierarchical density estimates, *Lecture Notes in Computer Science*, vol. 7819, 2013. https://doi.org/10.1007/978-3-642-37456-2_14

Chen, M., Qu, R. and Fang, W., Case-based reasoning system for fault diagnosis of aero-engines, *Expert Systems with Applications*, vol. 202, 2022. https://doi.org/10.1016/j.eswa.2022.117350

Ge, W., Zhou, J., Zheng, P., Yuan, L. and Rottok, L.T., A recommendation model of rice fertilization using knowledge graph and case-based reasoning, *Computers and Electronics in Agriculture*, vol. 219, 2024. https://doi.org/10.1016/j.compag.2024.108751

Hussain, M., YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection, *Machines*, vol. 11, no. 7, pp. 677, 2023.

Khosravani, M.R. and Nasiri, S., Injection molding manufacturing process: review of case-based reasoning applications, *Journal of Intelligent Manufacturing*, vol. 31, pp. 847–864, 2020. https://doi.org/10.1007/s10845-019-01481-0

Lavanya, G. and Pande, S.D., Enhancing real-time object detection with YOLO algorithm, *EAI Endorsed Transactions on Internet of Things*, vol. 10, Dec. 2023.

Lerones, P., Fernández, J., García-Bermejo, J. and others, Total quality control for automotive raw foundry brake disks, *International Journal of Advanced Manufacturing Technology*, vol. 27, pp. 359–371, 2005. https://doi.org/10.1007/s00170-004-2165-9

Li, C., Wang, D. and Yang, W., Case representation and retrieval for complex product design based on case-based reasoning, *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 3, pp. 2985–3002, 2022. https://doi.org/10.3233/JIFS-212927

Li, J., Luo, X. and Lu, G., GS-CBR-KBQA: Graph-structured case-based reasoning for knowledge base question answering, *Expert Systems with Applications*, vol. 257, 2024. https://doi.org/10.1016/j.eswa.2024.125090

Mao, M. and Hong, M., YOLO object detection for real-time fabric defect inspection in the textile industry: A review of YOLOv1 to YOLOv11, *Sensors*, vol. 25, no. 7, pp. 2270, 2025. https://doi.org/10.3390/s25072270

Murthy, J.S., Siddesh, G.M., Lai, W.-C., Parameshachari, B.D., Patil, S.N. and Hemalatha, K.L., Object Detect: A real-time object detection framework for advanced driver assistant systems using YOLOv5, *Wireless Communications and Mobile Computing*, 2022. https://doi.org/10.1155/2022/9444360

Ren, Z., Fang, F., Yan, N. and others, State of the art in defect detection based on machine vision, *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 9, pp. 661–691, 2022. https://doi.org/10.1007/s40684-021-00343-6

Smith, B., Six-sigma design (quality control), *IEEE Spectrum*, vol. 30, no. 9, pp. 43–47, 1993. https://doi.org/10.1109/6.275174

Sourati, Z., Ilievski, F., Sandlin, H.-Â. and Mermoud, A., Case-based reasoning with language models for classification of logical fallacies, *arXiv preprint,* 2023. https://doi.org/10.48550/arXiv.2301.11879

Sundaram, S. and Zeid, A., Artificial intelligence-based smart quality inspection for manufacturing, *Micromachines*, vol. 14, no. 3, pp. 570, 2023. https://doi.org/10.3390/mi14030570

Wang, S., Tao, C., Song, M., Fei, Y., Shan, L. and Jiao, R., GPT-powered case-based domain knowledge modeling and reasoning for cognitive intelligent manufacturing defect mitigation, *Proceedings of the 7th European Industrial Engineering and Operations Management Conference*, July 2024.

Wang, S., Zou, P., Gong, X., Song, M., Peng, J. and Jiao, J.R., Visual analytics and intelligent reasoning for smart manufacturing defect detection and judgement: A meta-learning approach with knowledge graph embedding case-based reasoning, *Journal of Industrial Information Integration*, vol. 37, 2024.

Wilkerson, K. and Leake, D., On implementing case-based reasoning with large language models, *Case-Based Reasoning Research and Development, Lecture Notes in Computer Science*, vol. 14775, 2024. https://doi.org/10.1007/978-3-031-63646-2_26

Yang, R., CaseGPT: A case reasoning framework based on language models and retrieval-augmented generation, *arXiv preprint* arXiv:2407.07913, 2024. https://doi.org/10.48550/arXiv.2407.07913

Zhang, Y., Wa, S., Sun, P. and Wang, Y., Pear defect detection method based on ResNet and DCGAN, *Information*, vol. 12, no. 10, pp. 397, 2021. https://doi.org/10.3390/info12100397

## Biographies

**Yiyun (Cindy) Fei** is a Ph.D. candidate in Mechanical Engineering at Georgia Institute of Technology, Georgia, USA, where she also earned her bachelor's and master's degree in the same field. Her research focuses on the modeling, analysis, and optimization of complex operational systems to support data-driven design and decision-making. Yiyun worked as a Manufacturing and Process Development Engineer at TE Connectivity, where she gained four years of hands-on industry experience. During her time there, she led and contributed to multiple projects involving AI-based machine vision, process automation, and discrete event simulation of manufacturing systems. More information is available on her LinkedIn profile: www.linkedin.com/in/yiyun-fei

**Mulang Song** is a Ph.D. candidate in the School of Mechanical Engineering at the Georgia Institute of Technology (USA). He is graduating in Mechanical Engineering, with research that spans both mechanical and industrial engineering domains. He received both his B.S. and M.S. degrees in Mechanical Engineering with a focus on Computer Science, also from Georgia Tech. He has led and contributed to engineering projects through direct collaboration with industry, applying research in real-world production environments. His research interests include advanced manufacturing systems, intelligent production systems, and industrial AI applications. More info about his research: https://scholar.google.com/citations?user=jLL4nycAAAAJ

**Shu Wang** is a smart manufacturing engineer at TE Connectivity. His expertise spans in industrial AI and data analytics, engineering design decision-making, human-automation interaction, and operations optimization. He holds a Ph.D. in Mechanical Engineering from Georgia Tech, along with dual master's degrees in Mechanical Engineering and Electrical and Computer Engineering. He earned his bachelor's degree in Electrical engineering from Wuhan University. More info about his research: https://scholar.google.com/citations?user=_R1ldZkAAAAJ&hl=en

**Xinping Deng** is an engineering manager focusing on smart factory with a focus on automation, robotics, industrial AI applications, manufacturing data analysis and innovation. He holds a master degree in Mechanical and Aerospace Engineering from Cornell University and a bachelor degree in Mechanical Engineering from Worcester Polytechnic Institute. More information is available on his LinkedIn profile: www.linkedin.com/in/xinping-deng

**Roger Jiao** is the editor-in-chief of Journal of Engineering Design and an associate professor of mechanical and industrial engineering at Georgia Tech, USA. Prior to joining the School of Mechanical Engineering at Georgia Tech in December 2008, he was an Assistant Professor and then Associate Professor in the School of Mechanical and Aerospace Engineering at Nanyang Technological University, Singapore. Before his career in Singapore, he was a Visiting Scholar in the Department of Industrial Engineering and Engineering Management at Hong Kong University of Science and Technology from 1998 to 1999. From 1993 to 1994, he was a Lecturer of Industrial Engineering in the School of Management at Tianjin University, China, and from 1988 to 1990, he worked as an Associate Lecturer in the Department of Industrial Design at Tianjin University of Science and Technology, China. More info about his research: https://scholar.google.com/citations?user=9yikEHAAAAAJ&hl=en&oi=ao.