Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Intelligent Robotic Vision Systems for Automated Surface Defect Inspection in Advanced Manufacturing

**Reyan Hridoy Bhuiyan**

reyanhridoyj7500@gmail.com

Minnesota State University Mankato https://orcid.org/0009-0004-6449-1109

**Tenzing Choeyang**

University of Minnesota College of Science and Engineering: University of Minnesota Twin Cities College of Science and Engineering

**Umme Habiba Aesha**

Minnesota State University Mankato

**Research Article**

**Keywords:** Intelligent robotic inspection, automated quality inspection, defect detection, YOLOv8, machine vision, surface defects, deep learning, advanced manufacturing

**Posted Date:** September 28th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-10944087/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

# Intelligent Robotic Vision Systems for Automated Surface Defect Inspection in Advanced Manufacturing

**Reyan Hridoy Bhuiyan[1], Tenzing Choeyang[2], Umme Habiba Aesha[3]**

[1]Department of Mechanical Engineering, Minnesota State University, Mankato, Minnesota, USA
Corresponding Author: reyanhridoyj7500@gmail.com

[2]Department of Industrial and Systems Engineering, University of Minnesota, Minneapolis, Minnesota, USA

[3]Department of Mechanical Engineering, Minnesota State University, Mankato, Minnesota, USA

## Abstract

Advanced manufacturing environments require reliable inspection methods capable of identifying surface defects with high consistency while reducing dependence on subjective manual inspection. This study presents an intelligent robotic inspection framework for automated quality inspection in advanced manufacturing environments, integrating robotic handling, machine vision, image preprocessing, and deep-learning-based defect detection. The framework establishes an end-to-end inspection workflow covering image acquisition, defect detection and classification, defect localization, inspection decision-making, and quality reporting. To quantitatively evaluate the defect-detection component, a YOLOv8n model was trained and tested using the NEU-DET benchmark dataset containing six metallic surface-defect categories: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. An independent test set comprising 180 images and 412 annotated defect instances was used for final evaluation. The model achieved an overall precision of 71.8%, recall of 73.1%, mAP@50 of 76.3%, and mAP@50–95 of 42.9%. Class-level analysis showed the strongest performance for scratches, which achieved 98.2% recall and 97.3% mAP@50, while crazing was the most challenging category, with 42.0% recall and 45.1% mAP@50. The YOLOv8n model required 4.3 ms per image for inference, demonstrating low computational latency for the defect-detection stage. These findings provide experimental evidence supporting the feasibility of deep-learning-based defect detection as a component of an intelligent robotic inspection framework. However, physical robotic-cell cycle time, inspection

repeatability, severity assessment, manual inspection comparison, and validation using client-specific inspection data require further experimental investigation.

**Keywords:** Intelligent robotic inspection; automated quality inspection; defect detection; YOLOv8; machine vision; surface defects; deep learning; advanced manufacturing.

## 1. Introduction

### 1.1 Background

The growing demand for high-quality products, faster production cycles, and improved manufacturing efficiency has made quality control an essential component of modern production systems. Industries such as automotive, aerospace, electronics, medical-device manufacturing, and precision engineering operate under stringent quality requirements, where undetected defects can adversely affect product performance, reliability, and safety. Consequently, there is increasing interest in inspection technologies capable of maintaining product quality while supporting efficient and consistent production. Traditional manufacturing inspection has relied extensively on manual visual inspection performed by experienced personnel. Although human inspectors can provide valuable judgment in identifying defects, manual inspection is affected by factors such as fatigue, human error, inconsistent decision-making, and variation in inspection conditions. These limitations become more significant in high-volume manufacturing environments where inspectors must evaluate large numbers of components within limited time periods. Industrial automation has therefore encouraged the development of robot-based inspection systems that combine robotic manipulation, machine vision, and sensing technologies within manufacturing workflows. Compared with manual inspection, robotic inspection can provide controlled positioning, repeatable motion, and standardized image-acquisition conditions. Robots can position components or inspection sensors at predefined locations and viewing angles, providing a structured environment for automated visual inspection. The development of computer vision and deep-learning techniques has further expanded the capabilities of automated inspection systems. Modern object-detection models can identify and classify complex surface defects while simultaneously providing information about their spatial locations. These developments have contributed to the broader adoption of intelligent inspection technologies within smart

manufacturing and Industry 4.0 environments. Despite these advances, automated inspection remains challenging because manufacturing defects can vary considerably in appearance, size, shape, texture, and contrast. Changes in component geometry, surface characteristics, illumination, and manufacturing conditions can affect detection performance. Furthermore, many inspection approaches focus primarily on defect identification, while broader inspection workflows may also require defect classification, localization, severity assessment, and quality-related decision-making. These challenges motivate the development of integrated robotic inspection frameworks that combine controlled image acquisition with intelligent defect analysis. Such frameworks can provide a structured pathway from physical inspection and image acquisition to automated defect detection and quality assessment. In this study, an intelligent robotic inspection framework is developed to integrate robotic manipulation, machine vision, image processing, and deep-learning-based defect detection. The defect-detection component is experimentally evaluated using YOLOv8n on the NEU-DET benchmark dataset to establish quantitative evidence concerning its ability to identify and localize metallic surface defects.

### 1.2 Problem Statement

Consistent product quality remains a major challenge in manufacturing despite continuing advances in industrial automation. Manufacturing defects may include cracks, scratches, dents, uneven surfaces, inclusions, pitted regions, rolled-in scale, and other surface irregularities. Such defects can arise from material variation, equipment conditions, manufacturing processes, and environmental factors. Failure to identify defects at an appropriate stage can result in rework, material waste, production delays, product rejection, and economic losses. Manual visual inspection remains common in many manufacturing environments, but its effectiveness depends on factors including inspector experience, concentration, fatigue, and subjective judgment. These factors can make consistent inspection difficult, particularly when large numbers of components must be examined within short production cycles. Automated vision-based inspection systems have been developed to address some of these limitations by combining imaging systems with computational image-analysis techniques. However, automated inspection systems may still encounter difficulties when defects exhibit substantial visual variation or when inspection conditions change. Variations in illumination, surface texture, component geometry, and defect appearance can reduce detection reliability. In addition, some inspection approaches focus

primarily on identifying defective regions without providing an integrated workflow for defect classification and localization. Robotic inspection technology provides an opportunity to improve the consistency of image acquisition by automating component positioning and inspection trajectories. Nevertheless, the effectiveness of a robotic inspection system ultimately depends on the capability of its vision and defect-analysis components. A robotic system that reliably acquires images but cannot accurately identify and localize defects cannot provide effective automated quality control. The research therefore addresses the need for an integrated inspection framework that combines robotic inspection with intelligent visual defect analysis. The proposed framework provides an architecture for automated image acquisition, defect detection, classification, localization, and inspection decision-making. To provide experimentally verifiable evidence, the present study focuses its quantitative evaluation on the deep-learning defect-detection component using YOLOv8n and the NEU-DET metallic surface-defect benchmark.

### 1.3 Research Goal

The primary goal of this research is to develop an intelligent robotic inspection framework capable of supporting automated visual inspection and defect analysis in advanced manufacturing environments. The proposed framework integrates robotic manipulation, machine vision, image processing, and intelligent defect analysis within a unified inspection workflow. The framework is designed to support the automated acquisition of inspection images, detection and classification of manufacturing defects, localization of defective regions, and subsequent inspection decision-making. These capabilities are intended to reduce dependence on subjective visual inspection and provide a structured basis for automated quality assessment. The research also aims to experimentally evaluate the defect-detection component of the proposed framework using established performance measures. In particular, the YOLOv8n detector is evaluated using precision, recall, F1-score, mAP@50, and mAP@50–95, together with computational processing time. The experimental evaluation provides quantitative evidence concerning the effectiveness of deep-learning-based defect detection within the broader robotic inspection architecture. The study further identifies the engineering requirements and additional validation activities necessary for deployment of such a framework in a physical manufacturing inspection environment. These include validation using application-specific inspection data, repeatability testing, severity assessment, and evaluation of complete robotic inspection-cycle performance.

## 1.4 Research Objectives

To achieve the research goal, the following objectives are established:

1. To develop an intelligent robotic inspection framework that integrates robotic manipulation, machine vision, image processing, and intelligent defect analysis for automated quality inspection in advanced manufacturing environments.
2. To implement a deep-learning-based defect detection and classification component capable of identifying multiple categories of manufacturing surface defects using the YOLOv8n object-detection architecture.
3. To provide a framework for defect localization and inspection assessment in which detected defects can be spatially identified and subsequently incorporated into quality-related inspection decisions.
4. To experimentally evaluate the YOLOv8n defect-detection component using quantitative performance measures, including precision, recall, F1-score, mAP@50, mAP@50–95, and computational processing time.
5. To analyze class-level detection performance across different metallic surface-defect categories and identify defect types that present greater challenges for automated detection.
6. To develop engineering recommendations for the implementation and further validation of intelligent robotic inspection systems in advanced manufacturing environments.

These objectives provide the basis for designing the proposed inspection framework and evaluating its experimentally validated defect-detection component while distinguishing benchmark-model performance from system-level performance that requires additional physical validation.

## 1.5 Research Contributions

This research contributes to the development of intelligent robotic inspection and automated quality-control systems by combining a proposed robotic inspection architecture with an experimentally evaluated deep-learning defect-detection component. The principal contributions are as follows:

1. An integrated intelligent robotic inspection framework is proposed that combines robotic manipulation, machine vision, image processing, defect analysis, and inspection decision-making within a unified architecture.
2. A YOLOv8n-based defect-detection component is implemented for automated detection and classification of multiple metallic surface-defect categories.
3. Experimental evaluation of the defect-detection component is conducted using the NEU-DET benchmark dataset, providing quantitative measurements of precision, recall, F1-score, mAP@50, and mAP@50–95.
4. Class-level defect analysis is provided to identify differences in detection performance across crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches.
5. Computational performance is quantified through measurement of preprocessing, inference, and postprocessing time, providing evidence concerning the computational characteristics of the detection component.
6. Engineering recommendations and validation requirements are identified for extending the benchmark evaluation toward physical robotic-cell deployment, including application-specific datasets, repeatability testing, severity assessment, and complete inspection-cycle evaluation.

Together, these contributions establish a technically grounded basis for integrating deep-learning-based defect detection into intelligent robotic inspection workflows while clearly distinguishing experimentally demonstrated results from system capabilities requiring further validation.

### 1.6 Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews existing literature on manufacturing quality inspection, machine-vision-based defect detection, robotic inspection systems, deep-learning-based inspection, and related automated quality-control technologies. The section examines existing approaches and identifies the research gap addressed by the study. Section 3 presents the proposed intelligent robotic inspection framework. It describes the overall system architecture, robotic inspection platform, image-acquisition process, image-preprocessing procedures, defect detection and classification module, defect localization strategy, severity-assessment concept, and inspection decision process. Section 4 describes the experimental

methodology used to evaluate the defect-detection component. It presents the NEU-DET dataset, experimental environment, YOLOv8n implementation, model-training configuration, evaluation procedure, and performance metrics. Section 5 presents the experimentally obtained results for the YOLOv8n defect-detection component. The section reports class-level and overall detection performance, localization-related metrics, computational processing time, and analysis of the observed defect-detection behaviour. Section 6 discusses the experimental findings, their implications for intelligent robotic inspection, the factors influencing detection performance, and the limitations of the current evaluation. It also identifies areas requiring further physical and application-specific validation. Section 7 provides engineering recommendations for implementing and extending intelligent robotic inspection systems in manufacturing environments. The recommendations address inspection configuration, image acquisition, model deployment, validation, and system integration. Finally, Section 8 concludes the study by summarizing the proposed framework, experimentally verified findings, contributions, limitations, and directions for future development.

## 2. Related Work

### 2.1 Manual Inspection in Manufacturing

Manual inspection has historically been an important component of quality assurance in manufacturing because experienced inspectors can adapt their judgment to different products, defect characteristics, and production conditions. Within manufacturing industries, inspectors may examine components for defects such as cracks, scratches, dents, corrosion, contamination, dimensional irregularities, and assembly problems. Human inspectors can also adapt their assessment when examining complex geometries or defect patterns that may be difficult to represent using predefined inspection rules. Consequently, manual inspection continues to be used in manufacturing applications where human expertise and contextual judgment are important. However, manual inspection has several limitations in modern high-volume manufacturing environments. Inspection performance depends strongly on human perception, experience, concentration, and physical condition. As production volumes increase, inspectors may be required to evaluate large numbers of components within short periods, increasing the potential for fatigue and inconsistent inspection decisions. Subtle defects may consequently be overlooked, while

variations in individual judgment may produce inconsistent inspection outcomes. Another limitation is repeatability. Different inspectors may reach different conclusions concerning the presence, type, or seriousness of a defect because inspection decisions can depend on individual experience and judgment. Even the same inspector may produce different decisions when examining similar components at different times or under different working conditions. Such variability creates challenges for manufacturers seeking consistent and measurable quality-control processes. Manual inspection also introduces economic and logistical considerations. Manufacturing organizations must allocate personnel and training resources to maintain inspection capability, while increasing production volumes may require additional inspection personnel. Increased labor requirements can raise operating costs without necessarily providing proportional improvements in inspection consistency or throughput. The challenges associated with manual inspection become more pronounced as modern products incorporate complex geometries, tighter dimensional tolerances, diverse materials, and increasingly demanding quality requirements. Manufacturers must simultaneously maintain high product quality and production throughput, creating a need for inspection technologies that can provide consistent performance at production speed. Manual inspection nevertheless remains relevant for specialized applications where human expertise and contextual judgment are important. However, its limitations have encouraged the development of automated inspection technologies based on machine vision, intelligent image analysis, and robotics. These technologies provide the foundation for increasingly automated approaches to manufacturing quality control.

### 2.2 Vision-Based Automated Inspection

The limitations associated with manual inspection have encouraged the adoption of vision-based automated inspection systems designed to improve inspection consistency, speed, and scalability. Such systems employ imaging devices, controlled illumination, and computational image-analysis techniques to examine manufactured components with reduced dependence on continuous human intervention. Vision-based inspection has therefore become an important technology for automated quality assurance in modern manufacturing. Early automated vision systems relied primarily on conventional image-processing techniques, including image segmentation, thresholding, edge detection, morphological operations, feature extraction, and texture analysis. These methods can be effective when inspection environments are well controlled and defect

characteristics are relatively predictable. Stable illumination, consistent component orientation, and uniform surface characteristics can allow rule-based image-processing approaches to achieve efficient inspection performance. However, conventional image-processing approaches can be sensitive to changes in illumination, surface appearance, material reflectivity, component orientation, and defect characteristics. Manufacturing defects may differ considerably in size, shape, texture, location, and contrast, making it difficult to develop manually engineered rules that generalize effectively across different inspection conditions. These limitations become particularly important when detecting subtle defects or when components exhibit complex surface characteristics. Advances in industrial imaging hardware and computational technologies have improved automated inspection capabilities. Higher-resolution cameras, controlled illumination systems, faster processors, and modern machine-learning techniques enable inspection systems to extract increasingly complex visual information from manufacturing images. Instead of relying exclusively on manually engineered features, intelligent inspection approaches can learn visual representations from labelled data and use those representations to identify defect patterns. Nevertheless, many automated vision systems continue to focus primarily on determining whether a defect is present. In practical quality-control applications, additional information such as defect category and spatial location may be required to support appropriate inspection decisions. Accurate localization can identify where a defect occurs, while classification can distinguish among different defect types. Severity assessment may additionally be required in applications where the size, extent, or characteristics of a defect determine whether a component should be accepted, repaired, or rejected. Another limitation is that vision systems are frequently developed as independent inspection components rather than as part of an integrated robotic manufacturing workflow. When image acquisition, robotic positioning, defect analysis, and inspection decisions operate as disconnected processes, coordination and automation can become more difficult. This has encouraged research into robotic inspection systems that combine controlled physical inspection with intelligent visual analysis.

### 2.3 Intelligent Robotic Inspection Systems

The convergence of industrial robotics, machine vision, and intelligent automation has expanded the capabilities of automated manufacturing inspection. Unlike conventional vision-based inspection systems that primarily focus on image analysis, robotic inspection systems can integrate

robotic manipulators with imaging and sensing technologies to automate component positioning and image acquisition. This integration enables repeatable inspection trajectories, controlled sensor positioning, and standardized image-acquisition conditions. Industrial robots can address several limitations associated with manual inspection. Their ability to execute predefined movements allows cameras and other sensors to be positioned consistently with respect to the inspected component. Maintaining consistent viewing angles and working distances can reduce variations in image acquisition and provide a more controlled basis for subsequent defect analysis. Robots can also perform repetitive inspection movements without the fatigue associated with human operators, making them potentially suitable for high-volume manufacturing applications. Recent developments in robotic sensing have expanded inspection capabilities through the use of high-resolution cameras, depth sensors, laser scanners, infrared systems, and three-dimensional vision technologies. These sensing modalities can provide additional information about component surfaces and geometries and can support inspection from multiple viewpoints. Multi-view acquisition may be particularly useful when defects occur on curved surfaces, complex geometries, or partially occluded regions. Despite these developments, the level of autonomy achieved by robotic inspection systems varies considerably across applications. In some implementations, robots primarily perform positioning or material-handling tasks while defect analysis is performed by separate image-processing systems. Such architectures can introduce additional integration requirements and may limit the degree to which inspection operations can be performed as a unified autonomous workflow. Adaptability also remains an important challenge. Manufacturing components can differ in geometry, surface texture, material properties, and defect characteristics. A robotic inspection system developed for a specific component or production process may therefore require recalibration, retraining, or reconfiguration when applied to another product or inspection environment. A further challenge concerns the integration of the complete inspection workflow. Many studies concentrate on particular components of inspection, such as robotic positioning, image acquisition, or defect detection, rather than addressing the complete sequence from image acquisition through defect analysis and inspection decision-making. Integrating these stages can provide a more coherent architecture for intelligent quality control. These observations motivate the continued development of robotic inspection frameworks that combine robotic manipulation, controlled image acquisition, and intelligent defect analysis. Within such frameworks, reliable defect detection and classification remain fundamental

components because the quality of subsequent inspection decisions depends directly on the accuracy and robustness of the visual analysis system.

### 2.4 Intelligent Defect Detection and Classification

The development of artificial intelligence and computer vision has substantially expanded the capabilities of automated manufacturing inspection. Unlike conventional image-processing approaches that depend heavily on manually engineered features and predefined decision rules, deep-learning models can learn visual representations from labelled inspection data. This capability has made deep learning particularly relevant to defect detection and classification tasks involving complex surface patterns. Convolutional neural networks (CNNs) have been widely applied to manufacturing inspection because of their ability to learn hierarchical visual features from images. CNN-based approaches can automatically extract relevant characteristics from inspection images and use these representations to classify different defect categories. Such methods have been investigated across applications including surface inspection, weld inspection, electronic-component inspection, and other manufacturing quality-control tasks. However, image-level classification does not directly provide the spatial position of a defect. Object-detection architectures address this limitation by simultaneously identifying defect categories and estimating their locations within an image. This capability is important for manufacturing inspection because a system may need to determine not only whether a component contains a defect but also where the defect is located. Modern object-detection architectures have consequently become important in automated defect inspection. The YOLO family is particularly relevant to applications requiring a balance between detection capability and computational efficiency. Its single-stage detection structure supports rapid inference and makes YOLO-based approaches suitable for applications where inspection decisions must be generated within relatively short processing times. Image segmentation provides an additional level of spatial information by estimating defect boundaries at the pixel level. Segmentation can provide information concerning defect shape and area that cannot be obtained directly from conventional bounding-box detection. However, the appropriate detection methodology depends on the inspection objective, available annotations, computational requirements, and characteristics of the target defects. Several challenges remain in intelligent defect detection. Manufacturing datasets can contain substantial variation in defect appearance, class distribution, illumination, material properties, surface finish, viewing angle, and background

characteristics. Small or low-contrast defects may be particularly difficult to detect, while visually similar defect categories can create classification challenges. Model performance can therefore vary considerably between defect classes even when overall detection performance appears satisfactory. Another important consideration is the distinction between model-level defect detection and complete robotic inspection. A high-performing detection model does not by itself constitute a complete autonomous inspection system. Practical deployment also requires reliable image acquisition, physical positioning, system communication, inspection decision logic, and application-specific validation. Consequently, deep-learning detection should be considered an important component within a broader robotic inspection architecture. In this study, YOLOv8n is used as the experimental defect-detection component of the proposed intelligent robotic inspection framework. Its performance is evaluated on the NEU-DET benchmark dataset across six metallic surface-defect categories. The evaluation provides quantitative evidence concerning precision, recall, mAP@50, mAP@50–95, class-level performance, and computational processing time.

### 2.5 Research Gap and Motivation

The reviewed literature demonstrates substantial progress in manufacturing inspection through the integration of manual expertise, machine vision, deep learning, and robotics. Manual inspection remains useful in applications requiring human judgment but is affected by operator dependency and limited scalability. Automated vision systems improve consistency and processing speed, while deep-learning-based detection methods provide increasingly effective mechanisms for identifying and localizing manufacturing defects. Robotic platforms further improve inspection consistency by automating positioning and image acquisition.

Nevertheless, several research gaps remain.

First, existing studies frequently focus on individual inspection functions rather than presenting an integrated inspection architecture. Robotic positioning, image acquisition, defect detection, localization, and quality assessment may be treated as separate components. This fragmentation can make it difficult to establish a coherent workflow for autonomous inspection. Second, defect detection performance can vary substantially between defect categories. Overall model metrics may therefore conceal weaknesses associated with particular defect types. Class-level analysis is important for determining which defects are readily detected and which require further model

development or additional training data. Third, many studies emphasize model-level detection performance without sufficiently considering the relationship between computational performance and practical inspection requirements. For manufacturing applications, detection accuracy must be considered together with processing latency and the requirements of the intended inspection workflow. Fourth, there remains a distinction between benchmark-based evaluation and validation within a physical industrial inspection cell. A model can demonstrate strong performance on a public benchmark dataset while still requiring application-specific validation before deployment on real production components. Differences in camera configuration, illumination, component geometry, surface characteristics, and defect distributions can affect real-world performance. The present study addresses these gaps by proposing an intelligent robotic inspection framework and experimentally evaluating its deep-learning defect-detection component using the NEU-DET metallic surface-defect dataset. The evaluation uses an independent test set of 180 images containing 412 annotated defect instances and examines overall and class-level detection performance using precision, recall, mAP@50, and mAP@50–95. Computational processing time is also reported to provide an indication of the model's inference characteristics.The resulting framework provides a basis for integrating deep-learning defect detection into robotic inspection workflows while maintaining a clear distinction between experimentally demonstrated model performance and system-level capabilities requiring further physical validation. This distinction is important for translating benchmark-based research into reliable industrial inspection systems.

## 3. Proposed Intelligent Robotic Inspection Framework

### 3.1 Overall Framework Architecture

This study proposes an intelligent robotic inspection framework that integrates robotic manipulation, machine vision, automated defect detection, defect localization, quality assessment, and inspection decision-making within a unified manufacturing inspection workflow. The framework is designed to reduce dependence on manual visual inspection while improving inspection consistency, repeatability, and the speed of defect identification. Rather than treating image acquisition, defect analysis, and quality evaluation as independent activities, the proposed architecture organizes these functions into a coordinated inspection pipeline. The overall framework consists of eight principal stages: robotic positioning, image acquisition, image

preprocessing, defect detection and classification, defect localization, defect severity assessment, inspection decision-making, and quality reporting. The stages are arranged sequentially so that information generated at one stage provides the input required by the subsequent stage. The inspection process begins with the robotic positioning stage, in which the industrial robot positions the component or inspection viewpoint within the predefined inspection workspace. In the intended industrial configuration, the robotic platform provides repeatable positioning and controlled movement, thereby establishing consistent conditions for image acquisition. The use of robotic positioning is intended to reduce variations associated with manual handling and to support repeatable inspection trajectories. Following positioning, the image acquisition stage obtains visual information from the inspected component. The quality of the acquired images is important because the performance of subsequent defect analysis depends directly on the visibility and consistency of defect characteristics. Appropriate camera positioning and controlled illumination are therefore incorporated into the inspection architecture to support reliable visual inspection. The acquired images are subsequently processed before being supplied to the defect detection model. The preprocessing stage provides a standardized representation suitable for automated analysis. In the experimental implementation reported in this study, the defect detection component uses the YOLOv8n object-detection model to identify and classify manufacturing surface defects. The defect detection and classification stage constitutes the principal experimentally evaluated intelligent component of the framework. YOLOv8n analyzes each inspection image and produces predictions consisting of defect categories, confidence values, and bounding-box coordinates. The experimental dataset contains six manufacturing surface-defect categories: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. These predictions provide both categorical and spatial information for subsequent inspection analysis. The localization stage uses the predicted bounding boxes to determine the spatial position of detected defects within the inspection image. This information is particularly important for manufacturing quality control because identifying the existence of a defect alone does not indicate where the defect occurs on the inspected component. Bounding-box localization therefore provides a practical representation of defect position and extent for downstream quality evaluation. The proposed framework also incorporates a defect severity assessment stage. This stage is intended to evaluate the significance of detected defects using characteristics such as defect category, detected region, geometric extent, and predefined engineering acceptance criteria. The severity assessment is designed as a

downstream quality-evaluation component of the framework rather than as an independently trained model in the present experimental study. The inspection decision stage subsequently translates inspection information into a manufacturing quality outcome. The proposed framework defines three possible decision categories: Pass, Rework, and Reject. A Pass decision represents an acceptable component, Rework represents a component containing a potentially correctable defect, and Reject represents a component containing a defect that exceeds the applicable acceptance criteria. Finally, the inspection reporting stage records the inspection outcome and relevant detection information. A complete inspection record can contain the defect class, confidence value, localization information, severity assessment, processing time, and final quality decision. Such records can support traceability, quality monitoring, production analysis, and future process improvement.

The complete workflow can therefore be represented as: The overall architecture of the proposed intelligent robotic inspection framework is shown in Figure 1.

**Robot Positioning**

↓

**Image Acquisition**

↓

**Image Preprocessing**

↓

**YOLOv8n Defect Detection and Classification**

↓

**Bounding-Box Localization**

↓

**Defect Severity Assessment**

↓

**Pass / Rework / Reject Decision**

↓

**Inspection Report**

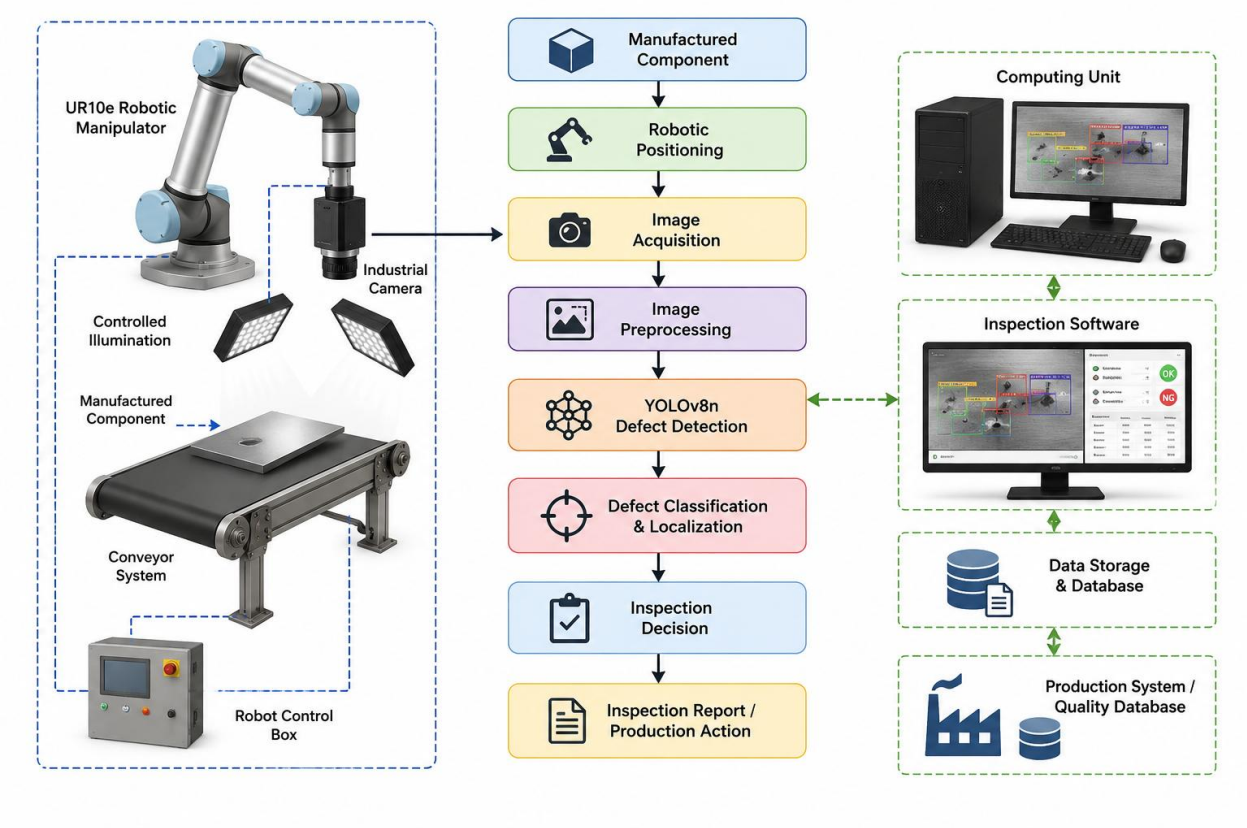



*Figure 1. Overall architecture of the proposed intelligent robotic inspection framework integrating robotic positioning, industrial image acquisition, YOLOv8n-based defect detection, defect localization, and automated inspection decision-making.*

An important distinction is made between the complete proposed framework and the experimental validation conducted in this study. The present quantitative experiment specifically validates the YOLOv8n-based defect detection, classification, and bounding-box localization component using the NEU-DET dataset. The severity assessment, automated quality decision, and reporting stages represent the downstream components of the proposed inspection architecture and provide the pathway for integrating the validated detection module into a complete industrial inspection system.

### 3.2 Robotic Inspection Platform

The physical inspection platform provides the robotic and sensing foundation required for implementing the proposed inspection framework. The intended system integrates an industrial

robotic manipulator with an industrial vision system, controlled illumination, material-handling equipment, and an industrial control interface. The purpose of this configuration is to provide repeatable positioning and consistent image acquisition during automated inspection. The robotic manipulator used for the intended inspection configuration is a Universal Robots UR10e six-axis collaborative robot. The UR10e provides a payload capacity of approximately 12.5 kg and a reach of approximately 1300 mm. Within the proposed inspection workflow, the robot is responsible for positioning the inspection viewpoint or component according to predefined inspection poses. Its repeatable six-axis movement provides the flexibility required to inspect different surfaces and viewpoints while maintaining controlled positioning. The vision component provides the visual information required for defect analysis. In the intended inspection configuration, the imaging system is used to capture surface information from the component under controlled inspection conditions. Camera positioning and triggering are coordinated with the robotic inspection sequence to minimize motion-related image variation and maintain consistent image acquisition. Controlled illumination is incorporated to improve the visibility of surface characteristics and reduce unwanted variations caused by shadows, reflections, or non-uniform lighting. Stable illumination is particularly important when detecting subtle surface defects because changes in illumination can alter the apparent visual characteristics of a defect. A conveyor or suitable material-handling mechanism can be integrated into the inspection cell to transport components into and out of the inspection position. The robot, imaging system, and material-handling system can be coordinated through the industrial control architecture to establish a repeatable inspection cycle. The principal components of the proposed robotic inspection platform are summarized in Table 1.

**Table 1. Principal components of the proposed robotic inspection platform**

| Component | Function | Typical Specification |
|---|---|---|
| Industrial Robot | Positions inspection sensor or component | Six-axis robotic manipulator with high repeatability |
| Industrial Camera | Captures inspection images | High-resolution RGB industrial camera |
| Illumination Unit | Provides controlled lighting | LED ring light, dome light, or directional lighting |
| Conveyor System | Transports manufactured components | Variable-speed automated conveyor |
| Industrial Controller | Coordinates system operation | PLC or industrial computer with real-time communication |

TThe integration of these components provides the physical foundation for an automated inspection cell. The robotic platform is intended to provide repeatable positioning, while the vision system and illumination provide standardized visual information for the intelligent defect-analysis module. For confidentiality, product-specific information not required to explain or reproduce the experimental methodology is not disclosed in this study.

### 3.3 Image Acquisition

Image acquisition is the first visual-processing stage of the proposed inspection workflow. Its purpose is to obtain inspection images containing sufficient visual information for subsequent defect detection and localization. During the intended robotic inspection process, the component is positioned within the inspection workspace by the robotic manipulator. Image acquisition is initiated after the inspection position has been established. This arrangement is intended to minimize motion-related image variation and to provide consistent viewing conditions across successive inspection cycles. The positioning of the imaging system is selected to provide adequate visibility of the surface regions being inspected. For components with relatively simple surfaces, a single inspection viewpoint may be sufficient. More complex components may require additional viewpoints to increase inspection coverage. Robotic movement provides the flexibility to change the relative position between the component and imaging system when required. Illumination is maintained as consistently as possible during image acquisition because surface defects can exhibit

significant changes in appearance under different lighting conditions. Controlled lighting therefore supports more stable visual representations and reduces the possibility that illumination changes will be interpreted as defect characteristics. The acquired images form the input to the subsequent image-processing and defect-detection stages. In the experimental component of this research, the YOLOv8n model was evaluated using the NEU-DET dataset, which provides annotated images of six categories of steel-surface defects. Consequently, the dataset-based experiment provides controlled validation of the defect-detection component of the proposed inspection architecture, while the robotic platform provides the intended physical mechanism for automated image acquisition.

### 3.4 Image Preprocessing

Image preprocessing is used to provide a standardized image representation for automated defect analysis. Preprocessing is particularly important in manufacturing inspection because differences in image dimensions, visual quality, and acquisition conditions can influence model performance. In the present implementation, the YOLOv8n/Ultralytics processing pipeline was used to prepare the input images for model training and inference. The dataset was organized according to the YOLO object-detection format, with separate image and label directories for the training, validation, and test subsets. Each annotation identifies the defect category and its corresponding bounding-box coordinates.

The experimental dataset contains six defect classes:

1. Crazing
2. Inclusion
3. Patches
4. Pitted surface
5. Rolled-in scale
6. Scratches

The image-processing pipeline converts the input images into the representation required by YOLOv8n during model training and inference. Standardized input processing enables the model to operate consistently across the dataset and supports efficient batch-based computation during

training. Unlike a manually engineered image-processing pipeline based on fixed operations such as histogram equalization or edge enhancement, the present experiment relies on the preprocessing procedures incorporated into the YOLOv8/Ultralytics training pipeline. This approach avoids introducing additional handcrafted transformations that were not experimentally evaluated. The preprocessing stage therefore produces the standardized input required by the YOLOv8n detection model.

The implemented processing sequence can be summarized as:

NEU-DET Image

↓

YOLO-Compatible Image/Annotation Preparation

↓

YOLOv8n Input Processing

↓

Defect Detection

This approach maintains consistency between training and inference while allowing the detection model to learn discriminative visual patterns directly from the annotated defect images.

### 3.5 YOLOv8n Defect Detection and Classification Model

The defect detection and classification component constitutes the principal intelligent-analysis module experimentally evaluated in this study. YOLOv8n was selected as the object-detection model because it provides simultaneous defect localization and classification while maintaining a relatively lightweight computational architecture suitable for real-time or near-real-time inspection applications. The YOLOv8n model processes an input inspection image and predicts the locations and categories of manufacturing defects. Each prediction consists primarily of a bounding box, a defect class, and a confidence score. The combination of these outputs enables the system to determine both what type of defect is present and where it occurs within the image. The

experimental implementation used the Ultralytics YOLO framework. The final trained model was saved as:

best.pt

The trained model contained approximately 3.01 million parameters and required approximately 8.1 GFLOPs, providing a relatively compact detection architecture. The final evaluation was performed using an NVIDIA Tesla T4 GPU environment. The model was trained using annotated NEU-DET images containing six defect classes. The training process used the corresponding YOLO-format bounding-box annotations to learn the visual characteristics and spatial distribution of the defect categories.

During inference, YOLOv8n produces predictions for each detected defect. Predictions include:

- defect category;
- bounding-box coordinates;
- confidence score; and
- multiple-defect predictions where more than one defect occurs in an image.

These outputs form the basis for the localization stage and provide the principal evidence used for evaluating defect-detection performance. The experimentally evaluated model demonstrated an overall precision of 0.718, recall of 0.731, mAP@0.5 of 0.763, and mAP@0.5:0.95 of 0.429 on the 180-image test set containing 412 annotated defect instances. The class-specific results demonstrated substantial variation across defect categories. The strongest mAP@0.5 performance was obtained for scratches (0.973) and patches (0.925), whereas crazing (0.451) produced the lowest class-specific mAP@0.5. This variation indicates that defect morphology and visual characteristics influence detection performance and is examined further in the results and discussion sections.

### 3.6 Defect Localization

Defect localization determines the spatial position of each defect identified by the YOLOv8n model. Unlike image-level classification, object detection provides spatial information by predicting a bounding box around each detected defect. For every retained prediction, the model

generates bounding-box coordinates together with the predicted defect category and confidence score. The bounding box provides an estimate of the location and spatial extent of the defect within the inspection image.

The localization procedure can therefore be summarized as:

**YOLOv8n** **Prediction**

↓

**Confidence** **Evaluation**

↓

**Bounding-Box** **Identification**

↓

**Defect** **Position** **and** **Extent**

↓

**Localization Output**

When multiple defects occur within the same image, the model can generate multiple bounding boxes, allowing the system to distinguish between separate defect instances. Localization performance is evaluated using object-detection metrics that compare predicted bounding boxes with the corresponding ground-truth annotations. The principal metrics used in this study include precision, recall, mAP@0.5, and mAP@0.5:0.95. These measures collectively evaluate the ability of the model to correctly identify defect instances and place their predicted bounding boxes in agreement with the annotated defect regions. The localization output also provides information that can subsequently be used by the proposed severity-assessment module. For example, bounding-box dimensions and location can serve as inputs to downstream engineering rules where defect size and position are relevant to quality evaluation. Importantly, the present implementation uses bounding-box detection rather than pixel-level segmentation. Therefore, localization results reported in this study should be interpreted as bounding-box localization rather than precise pixel-level defect boundary measurement.

### 3.7 Defect Severity Assessment

Detection and localization establish whether a defect exists, what category it belongs to, and where it is located. However, manufacturing quality decisions may additionally require an assessment of the significance of the detected defect. For this reason, the proposed framework incorporates a downstream defect severity assessment stage. The severity-assessment module is designed to combine information obtained from the detection and localization stages with engineering acceptance criteria. Relevant characteristics may include defect category, detected region, bounding-box dimensions, estimated defect extent, and the location of the defect relative to critical component regions.

The proposed framework categorizes defect severity into three levels:

- **Minor:** a defect that falls within acceptable manufacturing limits and has negligible influence on the intended function or quality of the component.
- **Moderate:** a defect that exceeds normal acceptance limits but may be correctable through an appropriate rework or additional inspection procedure.
- **Critical:** a defect that exceeds the applicable engineering acceptance criteria and may compromise product performance, reliability, or safety.

The conceptual severity function is represented as:

$$S = f(Ds, Da, Dl, C, T)$$

where S represents severity, Ds represents defect size, Da represents defect area or extent, Dl represents defect location, C represents model confidence, and T represents engineering decision thresholds. The severity module is intended to provide a connection between automated defect detection and engineering quality assessment. However, severity classification was not independently trained or quantitatively validated in the YOLOv8n experiment reported in this study. Therefore, the severity-assessment stage is presented as a proposed downstream component of the complete robotic inspection framework rather than as a separately validated experimental result. This distinction ensures that the reported experimental findings remain limited to the capabilities directly demonstrated by the available dataset and trained YOLOv8n model.

### 3.8 Inspection Decision Module

The inspection decision module represents the final proposed stage of the intelligent robotic inspection architecture. Its purpose is to translate defect-analysis information into an actionable manufacturing quality decision. The decision module receives information from the preceding inspection stages, including defect category, confidence, localization information, and, where implemented, severity assessment. These outputs can be compared against predefined engineering acceptance criteria to determine the appropriate disposition of the inspected component.

Three decision categories are defined within the proposed framework:

**Pass**

A component is assigned Pass when no unacceptable defect is detected or when all detected defects remain within the permitted manufacturing tolerance.

**Rework**

A component is assigned Rework when a detected defect exceeds the normal acceptance criterion but can potentially be corrected through an appropriate manufacturing or finishing operation.

**Reject**

A component is assigned Reject when the detected defect exceeds the applicable acceptance criteria or represents a condition for which correction is not considered appropriate.

The proposed decision workflow is:

**Detection Results**

↓

**Defect Classification and Localization**

↓

**Severity/Engineering Criteria**

↓

**Quality Decision**

↓

**Pass / Rework / Reject**

The decision module is intended to reduce subjective interpretation and provide a consistent basis for quality-control decisions. It can also provide a structured record of inspection outcomes for traceability and quality monitoring. However, similar to the severity-assessment stage, the Pass/Rework/Reject decision logic was not independently evaluated as a classification experiment in the present study. The quantitative experimental validation reported in this paper focuses on the YOLOv8n defect-detection component using the NEU-DET dataset. The decision module therefore represents the proposed mechanism through which the experimentally validated detection outputs can subsequently be incorporated into an autonomous industrial quality-control workflow. Overall, the proposed framework establishes a progression from robotic positioning and image acquisition to intelligent defect detection, spatial localization, engineering assessment, and quality decision-making. The experimentally validated YOLOv8n component provides the central defect-analysis capability, while the remaining framework stages establish the pathway toward a complete autonomous robotic inspection system.

## 4. Experimental Methodology

This section describes the experimental methodology used to evaluate the intelligent defect-analysis component of the proposed robotic inspection framework. The methodology focuses on the experimentally completed YOLOv8n-based defect detection and classification experiment using the NEU-DET dataset. The evaluation covers dataset preparation, experimental configuration, model training, performance measurement, and comparison scope. Particular attention is given to maintaining a clear distinction between experimentally verified results and capabilities proposed as part of the broader robotic inspection framework. The proposed framework presented in Section 3 is designed to integrate robotic manipulation, image acquisition, preprocessing, defect detection, localization, severity assessment, and automated quality decision-making. However, the completed benchmark experiment specifically validates the computer-vision detection component using an independent NEU-DET test set. Consequently, numerical claims concerning physical robotic repeatability, severity classification, complete robotic

inspection-cycle time, manual inspection performance, and alternative detection architectures are not presented as experimentally measured results in this study.

### 4.1 Dataset

The experimental evaluation was conducted using the NEU-DET dataset, a publicly available surface-defect dataset containing images representing common defects encountered in steel manufacturing. The dataset was selected because it provides annotated examples of visually distinct surface defects and is suitable for evaluating object-detection approaches for automated manufacturing inspection. The experimental dataset contains six defect categories: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. These defect classes exhibit substantial differences in morphology, texture, spatial extent, and visual appearance. Such variation provides an appropriate basis for evaluating the ability of the detection model to identify and distinguish multiple types of manufacturing defects within a common inspection environment.

Representative NEU-DET surface-defect images are shown in Figure 2. The six defect categories used in the experiment are summarized in Table 2.

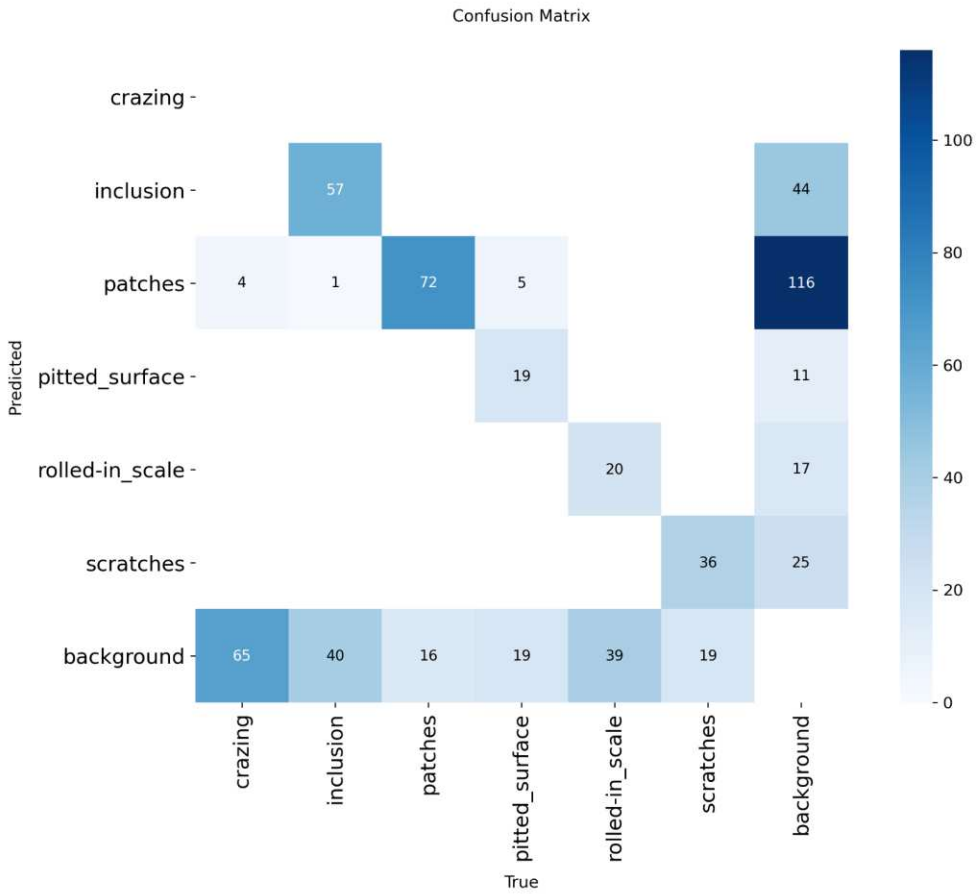



*Figure 2. Representative NEU-DET surface-defect images used for YOLOv8n-based defect detection.*

**Table 2. Defect categories in the NEU-DET dataset**

| **Defect class** | **Description** |
|---|---|
| Crazing | Network-like surface cracking or fine crack patterns |
| Inclusion | Foreign material or embedded non-metallic regions within the surface |
| Patches | Irregular localized regions exhibiting different surface characteristics |
| Pitted surface | Localized cavities or pit-like surface defects |
| Rolled-in scale | Surface regions associated with embedded or rolled-in scale |
| Scratches | Linear surface damage or elongated marks |

For supervised object detection, the images were accompanied by corresponding annotations identifying the location and category of each defect. The annotations were converted into a YOLO-compatible format containing the defect class and bounding-box coordinates required for model training and evaluation. This annotation structure enabled the YOLOv8n model to learn both the spatial location and category of manufacturing defects. The dataset was divided into training, validation, and testing subsets. The training subset was used for model parameter optimization, while the validation subset was used to monitor model performance during training and support model selection. The independent test subset was reserved for the final evaluation and was not used for model parameter optimization. The independent test set used for the reported experiment contained 180 images and 412 annotated defect instances across six defect categories. The distinction between images and defect instances is important because a single inspection image may contain more than one annotated defect. Consequently, the number of defect instances is greater than the number of test images.

The dataset organization and partitioning strategy are summarized in Table 3.

**Table 3. Dataset organization and experimental usage**

| **Dataset component** | **Purpose** |
|---|---|
| Training set | Optimization of YOLOv8n model parameters |
| Validation set | Monitoring model performance and model selection |
| Independent test set | Final evaluation of detection performance |
| Test images | 180 |
| Test defect instances | 412 |
| Number of defect classes | 6 |

The independent test set provides the basis for all quantitative results reported in Section 5. In particular, the test data were used to determine precision, recall, mAP@50, mAP@50–95, and class-level detection performance. Maintaining an independent evaluation set prevents the reported performance from being directly derived from the images used to optimize the model and provides a more meaningful indication of its performance on previously unseen samples. No additional

severity annotations were available in the completed benchmark experiment. Therefore, the dataset supports evaluation of defect detection, classification, and detection-related localization performance but does not provide sufficient ground-truth information for experimentally validating the proposed Minor, Moderate, and Critical severity categories.

### 4.2 Experimental Setup

The experimental setup was configured to evaluate the YOLOv8n defect-detection component of the proposed intelligent robotic inspection framework. The experiment was conducted as a computer-vision benchmark using the NEU-DET dataset rather than as a complete physical production-line evaluation. The proposed robotic platform described in Section 3 provides the intended physical architecture for autonomous inspection, including robotic manipulation, industrial imaging, controlled illumination, conveyor operation, and centralized system coordination. Within the completed experiment, however, the principal experimentally evaluated component was the intelligent visual inspection model. The NEU-DET images served as the input to the detection pipeline, allowing the model's ability to identify and classify surface defects to be evaluated independently of physical robot motion. The YOLOv8n model was selected as the detection architecture because of its suitability for real-time object detection and its relatively low computational complexity. The model processes an input inspection image and produces bounding boxes, predicted defect classes, and confidence values for detected regions. These outputs correspond to the detection and classification functions required by the proposed robotic inspection architecture.

The experimental workflow can therefore be represented as:

**NEU-DET Inspection Image**

↓

**Image Preparation**

↓

**YOLOv8n Detection**

↓

**Defect Classification**

↓

**Bounding-Box Localization**

↓

**Confidence Evaluation**

↓

**Performance Measurement**

The computational implementation was based on Python and the YOLOv8 framework. The model was trained and evaluated using the prepared YOLO-format dataset containing six manufacturing defect categories. The dataset configuration specified separate training, validation, and test image locations together with the corresponding annotation files and class definitions. The experimental environment was maintained consistently during model development and evaluation. The same dataset structure and class definitions were used throughout training, validation, and testing. The independent test set was evaluated only after model training had been completed, ensuring that test results represented an independent assessment of the trained detector. It is important to distinguish

the computational experiment from the complete robotic platform proposed in Section 3. The reported processing times therefore describe the computer-vision processing component and do not represent the total time required for robot movement, camera triggering, conveyor operation, component positioning, communication, or automated physical handling. The experimental configuration of the robotic inspection platform is shown in Figure 3.

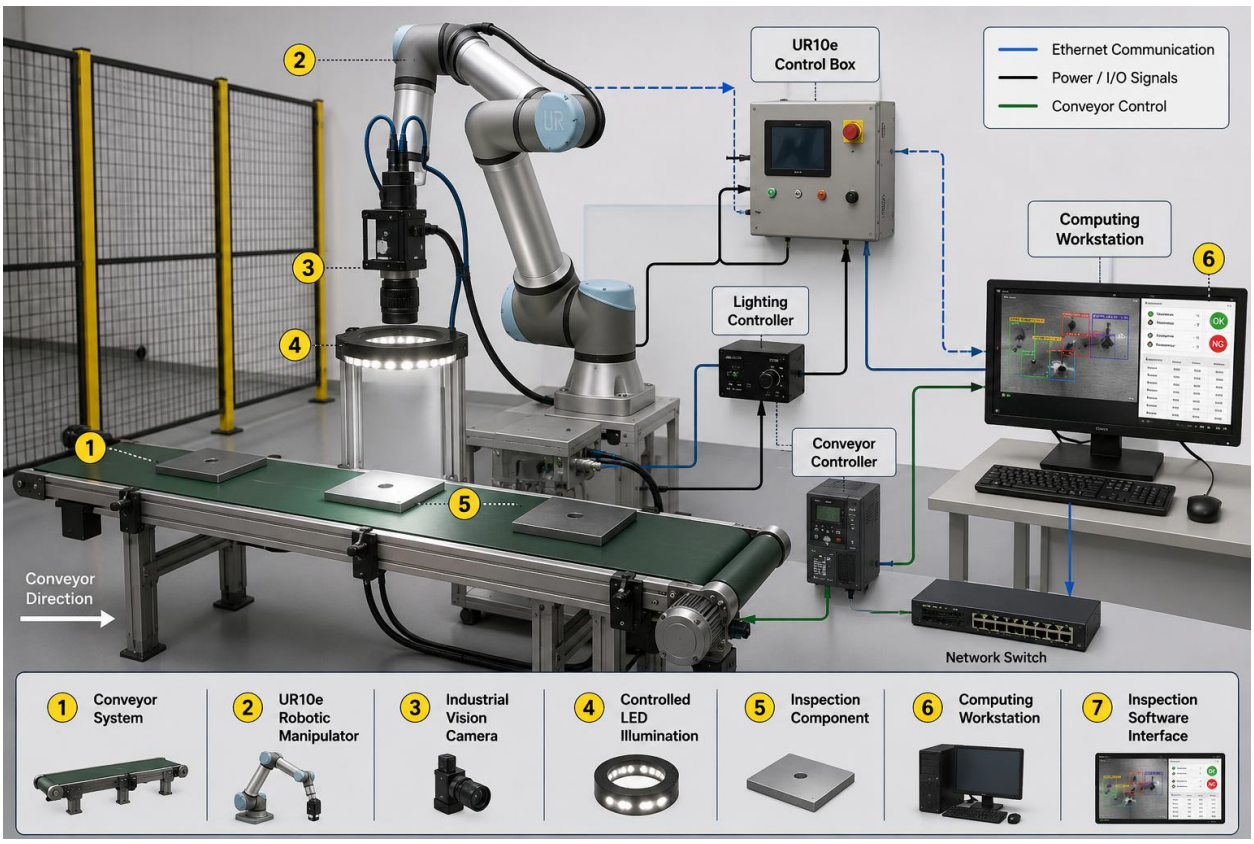



*Figure 3. Experimental configuration of the robotic inspection platform showing the robotic manipulator, industrial vision system, controlled illumination, conveyor system, and computing workstation used for automated defect inspection.*

### 4.3 YOLOv8n Model and Training Configuration

The defect detection component was implemented using YOLOv8n, the nano configuration of the YOLOv8 object-detection architecture. The model was selected to provide a computationally efficient solution suitable for applications in which detection accuracy must be balanced against inference latency. YOLOv8n performs single-stage object detection by processing an input image

and directly predicting the locations and categories of objects within the image. In the present application, the objects correspond to manufacturing surface defects. For each detected region, the model produces a bounding box, a defect-class prediction, and an associated confidence score. The model was trained using supervised learning. During training, the annotated images were presented to the network, and the predicted defect locations and classes were compared with their corresponding ground-truth annotations. Model parameters were iteratively adjusted to reduce prediction error and improve the ability of the detector to identify the six defect categories. The training process was conducted for a maximum of 100 epochs. During each epoch, the model processed the training data in batches and updated its parameters through gradient-based optimization. Validation data were used during training to monitor model performance and support selection of the trained model used for subsequent evaluation.

The principal training configuration is summarized in Table 4.

**Table 4. YOLOv8n training configuration**

| **Parameter** | **Configuration** |
|---|---|
| Detection architecture | YOLOv8n |
| Learning approach | Supervised object detection |
| Dataset | NEU-DET |
| Number of defect classes | 6 |
| Training duration | Up to 100 epochs |
| Output | Bounding boxes, defect classes, confidence scores |
| Evaluation set | Independent NEU-DET test set |
| Primary objective | Automated manufacturing-defect detection and classification |

The training process was monitored to determine whether the model continued to improve as additional epochs were completed. The model learned increasingly discriminative visual representations of the defect categories through repeated exposure to the annotated training examples. Validation performance provided an independent indication of model behavior during

training and helped reduce the risk of selecting a model that performed well only on the training data. Following completion of training, the resulting YOLOv8n detector was evaluated on the independent test set. No test images were used for parameter optimization. The final evaluation therefore measured the detector's ability to generalize to images that were not used directly for model training. The trained model generated predictions for each test image. Predictions included the detected defect category, bounding-box coordinates, and confidence score. These outputs formed the basis for the precision, recall, mAP@50, and mAP@50–95 measurements reported in Section 5. Because the completed experiment was based on an existing annotated benchmark dataset, the training experiment should be interpreted primarily as validation of the intelligent defect-analysis component rather than as a complete validation of the physical robotic inspection platform.

### 4.4 Evaluation Metrics

The performance of the YOLOv8n detector was evaluated using metrics appropriate for object-detection tasks. These metrics were selected to measure complementary aspects of detection performance, including the reliability of positive predictions, the ability to identify actual defects, and the accuracy of detected regions across different IoU thresholds.

#### Precision

Precision measures the proportion of predicted defect detections that correspond to actual defects. It is particularly relevant to manufacturing inspection because excessive false-positive detections may result in unnecessary rejection or additional inspection of acceptable components.

$$Precision = TP/TP + FP$$

where TP represents true-positive detections and FP represents false-positive detections.

#### Recall

Recall measures the proportion of actual defect instances successfully detected by the model. High recall is particularly important in manufacturing quality control because missed defects may remain in the production stream.

$$Recall = TP/TP + FN$$

where FN represents false-negative detections.

**F1-score**

The F1-score provides a combined measure of precision and recall and is calculated as their harmonic mean:

$$F1 = 2\,Precision \times Recall\,/Precision + Recall$$

For the reported experiment, the overall F1-score was calculated from the reported precision and recall values. The resulting value was approximately 72.4%.

Mean Average Precision

Mean Average Precision (mAP) was used as the principal object-detection performance measure. Two mAP measures were considered. mAP@50 evaluates detection performance using an IoU threshold of 0.50. It provides an indication of the model's ability to identify the correct defect categories while producing sufficiently overlapping bounding boxes. mAP@50–95 evaluates detection performance across multiple IoU thresholds from 0.50 to 0.95 at increments of 0.05. Because this metric requires increasingly precise spatial overlap at higher IoU thresholds, it provides a more demanding assessment of object-detection performance. The difference between mAP@50 and mAP@50–95 is therefore useful for assessing how detection performance changes when stricter localization requirements are imposed.

Intersection over Union

Intersection over Union (IoU) represents the degree of overlap between a predicted bounding box and its corresponding ground-truth annotation:

$$\boldsymbol{IoU = Area(Bpred \cap Bgt) \,/\, Area(Bpred \cup Bgt)}$$

where Bpred is the predicted bounding region and Bgt is the ground-truth bounding region.

IoU is fundamental to object-detection evaluation because it determines whether a predicted region sufficiently overlaps the reference annotation. However, the completed experiment did not produce an independent mean-IoU value for reporting. Therefore, no separate numerical mean-IoU result is claimed in Section 5. Instead, localization-related performance is reported through the mAP measures generated during YOLOv8 evaluation.

**Processing Time**

Computational processing time was also recorded to assess the efficiency of the detection pipeline. The measured processing stages were preprocessing, model inference, and postprocessing.

The completed experiment produced:

- 3.0 ms/image for preprocessing;
- 4.3 ms/image for model inference; and
- 2.0 ms/image for postprocessing.

The combined computer-vision processing time was therefore approximately 9.3 ms/image. This measurement represents computational image-processing time only. It does not include robotic movement, physical component positioning, camera triggering, conveyor operation, communication delays, or quality-report generation.

The evaluation metrics used in the completed experiment are summarized in Table 5.

**Table 5. Evaluation metrics used in the experimental study**

| **Metric** | **Purpose** |
|---|---|
| Precision | Measures reliability of positive defect detections |
| Recall | Measures the ability to identify actual defect instances |
| F1-score | Provides a balanced measure of precision and recall |
| mAP@50 | Evaluates object-detection performance at IoU = 0.50 |
| mAP@50-95 | Evaluates detection performance across stricter IoU |

| | |
|---|---|
| | thresholds |
| IoU | Measures overlap between predicted and ground-truth regions |
| Preprocessing time | Measures image-preparation latency |
| Inference time | Measures YOLOv8n prediction latency |
| Postprocessing time | Measures output-processing latency |
| Combined processing time | Measures total computer-vision processing latency |

Together, these metrics provide a balanced assessment of the trained YOLOv8n detector by considering both predictive performance and computational efficiency.

### 4.5 Baseline and Comparative Evaluation Scope

The proposed methodology originally identifies several approaches that could serve as comparative baselines for an intelligent robotic inspection system, including manual inspection, classical image processing, YOLO-based detection, Faster R-CNN, and contemporary state-of-the-art detection models. However, the completed experimental study generated quantitative results only for YOLOv8n. Accordingly, the comparative scope of the present study is explicitly restricted to experimentally verified results. No numerical performance values are assigned to Faster R-CNN, other state-of-the-art detectors, or manual inspection because these approaches were not evaluated under the same experimental conditions. Manual visual inspection remains relevant as an industrial reference because it represents a conventional quality-assurance approach against which autonomous inspection can ultimately be assessed. Nevertheless, the present experiment did not involve a controlled group of human inspectors examining the same test components. Consequently, manual inspection accuracy, inspection time, and repeatability are not reported as measured results. Similarly, Faster R-CNN and other alternative deep-learning detectors were not trained and evaluated using the experimental dataset. Their performance is therefore not presented as part of the experimentally verified results. This restriction prevents literature-derived or estimated values from being incorrectly interpreted as results obtained under the experimental conditions of this study.

The completed experimental scope is summarized in Table 6.

**Table 6. Experimental and non-experimental evaluation components**

| Evaluation component | Status in present study |
|---|---|
| YOLOv8n defect detection | Experimentally evaluated |
| Defect classification | Experimentally evaluated |
| Detection-related localization | Experimentally evaluated through mAP |
| Independent test-set evaluation | Experimentally evaluated |
| Computational processing time | Experimentally measured |
| Severity classification | Not experimentally evaluated |
| Physical robotic repeatability | Not experimentally measured |
| Complete robotic inspection-cycle time | Not experimentally measured |
| Manual inspection comparison | Not experimentally performed |
| Faster R-CNN comparison | Not experimentally performed |
| Additional SOTA detector comparison | Not experimentally performed |

This distinction establishes the experimental boundary of the study. The broader robotic architecture proposed in Section 3 provides the conceptual and engineering framework for autonomous inspection, while the present experiment provides quantitative validation of its intelligent visual defect-detection component.

### 4.6 Experimental Procedure

The experimental procedure followed a sequential workflow from dataset preparation to independent model evaluation. First, the NEU-DET images and corresponding defect annotations were organized into the required YOLO-compatible structure. The six defect categories were encoded consistently across the dataset to ensure that the same class definitions were used during model training and evaluation. Second, the YOLOv8n model was initialized for the six-class defect-detection task. The annotated training images were then used to optimize the model parameters over the defined training period of up to 100 epochs. Validation data were monitored during the training process to assess model behavior on data separate from the training samples.

Third, following completion of model training, the trained detector was applied to the independent test set containing 180 images and 412 annotated defect instances. For every test image, the model generated predicted defect classes, bounding boxes, and confidence values. Fourth, the predictions were evaluated against the corresponding ground-truth annotations. Precision, recall, mAP@50, and mAP@50–95 were obtained from the detection evaluation. The overall F1-score was subsequently calculated from the reported precision and recall values. Finally, the computational processing time associated with preprocessing, inference, and postprocessing was recorded. These measurements were used to assess the computational efficiency of the detection component.

The complete experimental procedure can be summarized as:

**NEU-DET Dataset**

↓

**Annotation and YOLO Dataset Preparation**

↓

**Training/Validation Data**

↓

**YOLOv8n Training**

↓

**Validation Monitoring**

↓

**Independent Test Set**

↓

**Defect Detection and Classification**

↓

**Bounding-Box Evaluation**

↓

**Precision / Recall / F1 / mAP Evaluation**

↓

**Processing-Time Measurement**

↓

**Experimentally Verified Results**

This procedure ensures that the results presented in Section 5 correspond directly to the completed YOLOv8n experiment and do not incorporate unmeasured assumptions concerning physical robotic performance.

### 4.7 Experimental Scope and Reproducibility

The methodology was designed to maintain a clear relationship between the proposed framework, the completed computational experiment, and the reported results. The experiment uses a fixed six-class NEU-DET configuration and an independent test set containing 180 images and 412 annotated defect instances. The same class definitions and evaluation criteria are applied throughout the experimental process. The use of an independent test set provides a basis for assessing model generalization, while reporting class-level performance allows differences in defect difficulty to be examined. This is particularly important because manufacturing defects can vary substantially in visual complexity. As demonstrated in Section 5, scratches and patches were detected more reliably than crazing, indicating that defect morphology influences the performance of the trained detector. The methodology also deliberately separates measured performance from proposed system functionality. The proposed framework includes severity assessment, automated Pass/Rework/Reject decisions, robotic coordination, and quality reporting. These components

establish the intended end-to-end architecture, but numerical validation of these functions requires additional physical experiments and appropriately labelled data. They are therefore not presented as experimentally validated outcomes in the current study. This methodological boundary strengthens the reliability of the experimental analysis by ensuring that each numerical result reported in the following section can be traced to an actual completed experiment. The resulting evaluation provides quantitative evidence for the effectiveness and computational efficiency of the YOLOv8n defect-detection component while identifying the additional experiments required to fully validate the complete intelligent robotic inspection framework.

## 5. Results

### 5.1 Defect Detection and Classification Performance

The trained YOLOv8n model was evaluated using the independent NEU-DET test set containing 180 images and 412 annotated defect instances across six defect categories: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. The model achieved an overall precision of 71.8%, recall of 73.1%, mAP@50 of 76.3%, and mAP@50–95 of 42.9%. The class-level results show noticeable variation across defect categories. Scratches produced the strongest detection performance, achieving 81.2% precision, 98.2% recall, and 97.3% mAP@50. Patches also performed strongly, with 84.2% precision, 89.8% recall, and 92.5% mAP@50. In contrast, crazing produced the lowest performance, with 52.9% precision, 42.0% recall, and 45.1% mAP@50. The relatively lower performance for crazing indicates that this defect category presented greater visual difficulty for the trained detector. The class-level defect detection results are summarized in Table 7.

**Table 7. Class-level defect detection performance of YOLOv8n on the independent test set**

| Defect class | Images containing class | Instances | Precision (%) | Recall (%) | mAP@50 (%) | mAP@50–95 (%) |
|---|---|---|---|---|---|---|
| Crazing | 30 | 69 | 52.9 | 42.0 | 45.1 | 16.4 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Inclusion | 35 | 98 | 70.2 | 74.5 | 78.4 | 41.8 |
| Patches | 35 | 88 | 84.2 | 89.8 | 92.5 | 58.9 |
| Pitted surface | 30 | 43 | 78.0 | 69.8 | 75.7 | 45.3 |
| Rolled-in scale | 30 | 59 | 64.2 | 69.4 | 68.8 | 36.9 |
| Scratches | 30 | 55 | 81.2 | 98.2 | 97.3 | 58.1 |
| Overall | 180 | 412 | 71.8 | 73.1 | 76.3 | 42.9 |

The results demonstrate that the detector was particularly effective for scratches and patches, whereas crazing remained the most challenging category. The difference between mAP@50 and mAP@50–95 also indicates that although the detector identified the relevant defects reasonably well at the IoU threshold used for mAP@50, stricter localization requirements reduced performance.

The class-wise prediction distribution is further illustrated by the normalized confusion matrix in Figure 4, which provides a visual representation of the model's classification behavior across the evaluated defect categories.

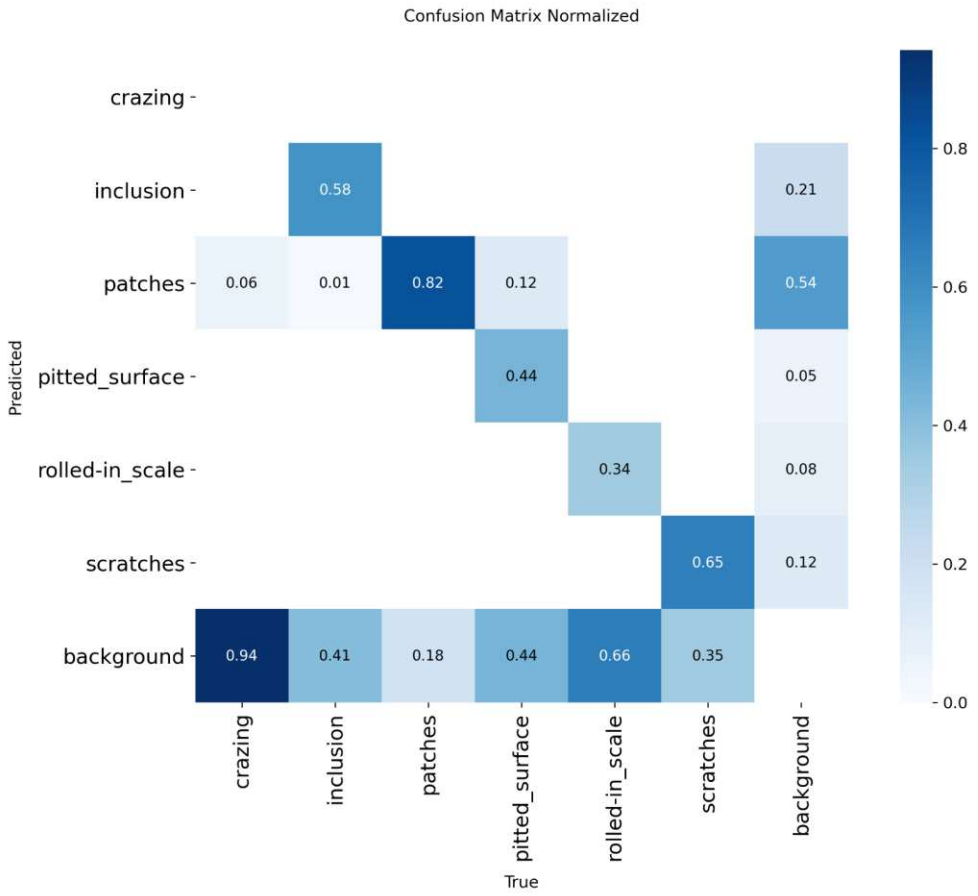



*Figure 4. Normalized confusion matrix of the YOLOv8n defect detection model across the six manufacturing defect classes.*

### 5.2 Overall Detection Performance

The overall performance metrics provide a consolidated assessment of the YOLOv8n detector. As shown in Table 8, the model achieved a precision of 71.8% and recall of 73.1%. The harmonic mean of these two values gives a calculated F1-score of approximately 72.4%. The mAP@50 value of 76.3% represents the overall detection performance at an IoU threshold of 0.50, whereas the mAP@50–95 value of 42.9% evaluates performance over a broader range of IoU thresholds. The lower mAP@50–95 value reflects the increased difficulty of maintaining accurate bounding-box localization under stricter overlap requirements.

**Table 8. Overall YOLOv8n detection performance**

| Metric | Result |
|---|---|
| Test images | 180 |
| Test defect instances | 412 |
| Precision | 71.8% |
| Recall | 73.1% |
| F1-score* | 72.4% |
| mAP@50 | 76.3% |
| mAP@50–95 | 42.9% |

*Calculated from the reported overall precision and recall.

The relationship between detection confidence and F1-score is illustrated in Figure 5. The curves show that the optimal confidence level varies among the individual defect categories, reflecting differences in the visual characteristics and detectability of the defects.

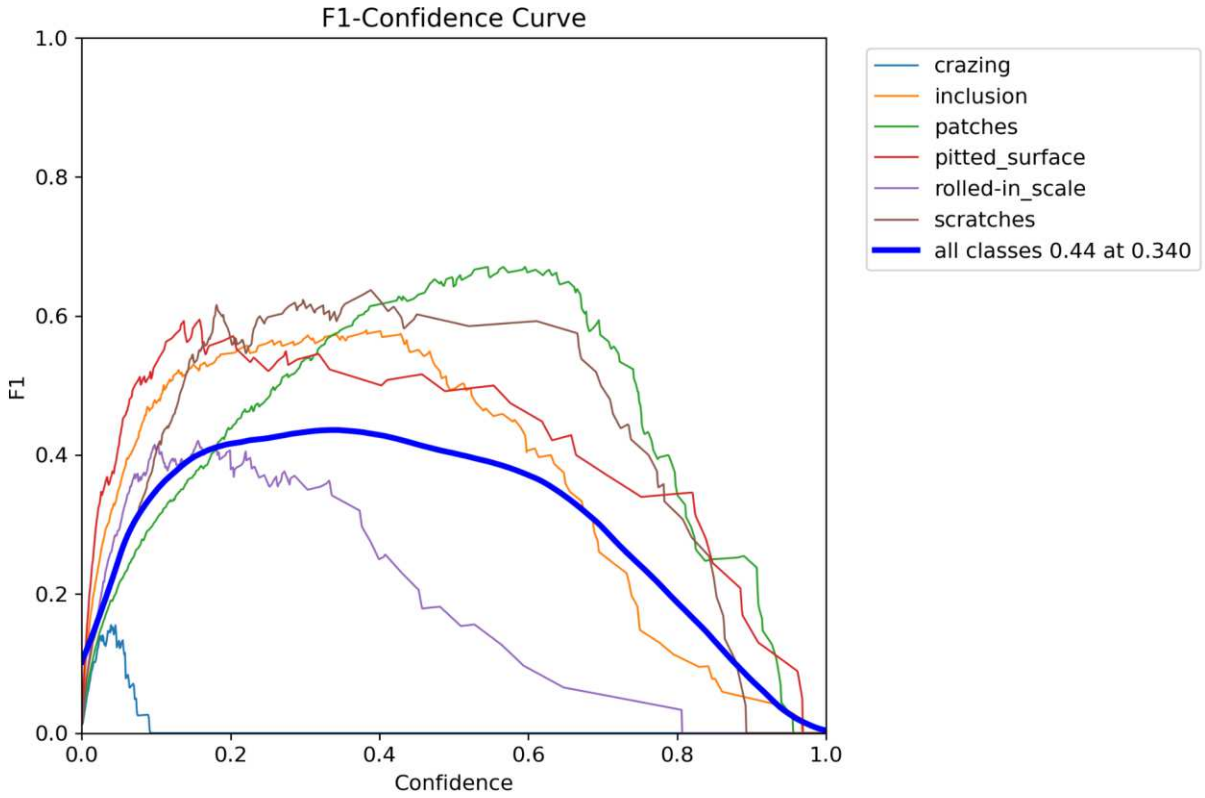


*Figure 5. F1-score variation with confidence threshold for the YOLOv8n detector across the evaluated defect categories.*

The precision–recall relationship for the individual defect categories is presented in Figure 6. The curves provide additional visual evidence of the differences in detection performance among the six defect classes.

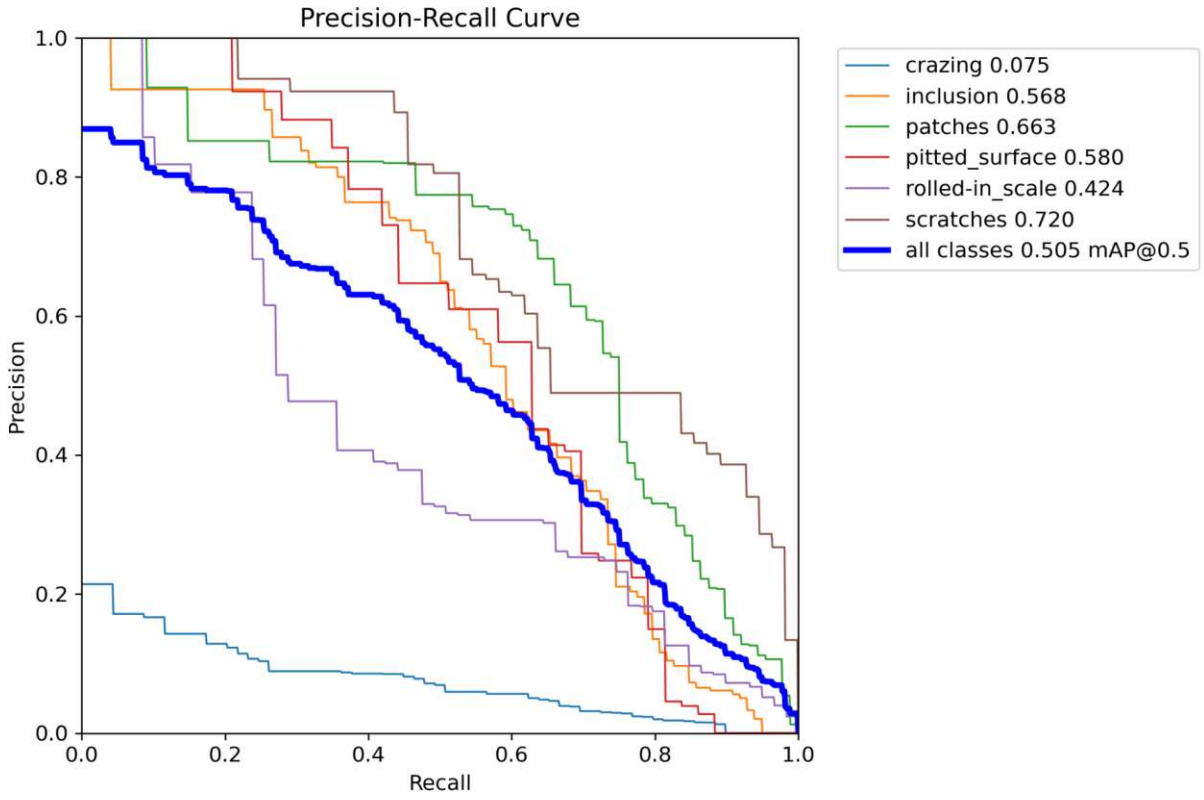



*Figure 6. Precision–recall curves for the YOLOv8n detector across the evaluated defect categories.*

### 5.3 Defect Localization Performance

Detection and localization performance was assessed using the mAP measures produced by the YOLOv8 evaluation. The detector achieved 76.3% mAP@50 and 42.9% mAP@50–95. At mAP@50, scratches achieved the highest class-level score at 97.3%, followed by patches at 92.5%. Crazing recorded the lowest value at 45.1%. The mAP@50–95 results followed a similar pattern,

with patches and scratches achieving the strongest localization performance. The class-level localization results are summarized in Table 9.

**Table 9. Class-level localization performance**

| Defect class | mAP@50 (%) | mAP@50–95 (%) |
|---|---|---|
| Crazing | 45.1 | 16.4 |
| Inclusion | 78.4 | 41.8 |
| Patches | 92.5 | 58.9 |
| Pitted surface | 75.7 | 45.3 |
| Rolled-in scale | 68.8 | 36.9 |
| Scratches | 97.3 | 58.1 |
| Overall | 76.3 | 42.9 |

A separate numerical mean-IoU value was not directly produced by the completed experiment. Therefore, IoU is not reported as an independent numerical result.

### 5.4 Computational Inspection Performance

The YOLOv8 evaluation also provided computational processing times. Preprocessing required 3.0 ms/image, model inference required 4.3 ms/image, and postprocessing required 2.0 ms/image. The combined computational processing time was therefore approximately 9.3 ms/image. The computational processing times are summarized in Table 10.

**Table 10. YOLOv8n computational processing time**

| Processing stage | Time (ms/image) |
|---|---|
| Preprocessing | 3.0 |
| Inference | 4.3 |
| Postprocessing | 2.0 |
| Combined processing time | 9.3 |

The measured inference time represents the computer-vision processing stage rather than the complete robotic inspection cycle. Robot motion, camera triggering, conveyor movement, physical positioning, communication, and reporting were not included in this measurement.

### 5.5 Severity Assessment

The completed YOLOv8 experiment evaluated defect detection and classification but did not include severity-labelled annotations. Consequently, quantitative severity-assessment performance cannot be derived from the current test set. The experimental status of severity assessment is summarized in Table 11.

**Table 11. Status of severity assessment**

| Component | Experimental status |
|---|---|
| Defect detection | Evaluated |
| Defect classification | Evaluated |
| Defect localization | Evaluated |
| Severity classification | Not evaluated |
| Severity estimation accuracy | Not available |

The proposed framework can accommodate severity assessment as a downstream inspection stage; however, numerical claims require an appropriately labelled severity dataset or physical inspection records.

### 5.6 Repeatability

Repeatability was not quantitatively evaluated in the completed experiment. The current test used an independent set of images rather than repeated inspections of the same physical components under controlled conditions. The repeatability evaluation status is summarized in Table 12.

**Table 12. Repeatability evaluation status**

| Measure | Status |
|---|---|

| | |
|---|---|
| Repeated physical inspections | Not measured |
| Prediction variance | Not measured |
| Robot positioning repeatability | Not measured |
| Vision-system repeatability | Not measured |
| Inspection-to-inspection consistency | Not measured |

Accordingly, repeatability is treated as a system-level evaluation requirement rather than a measured outcome of the current benchmark experiment.

### 5.7 Comparison with Manual Inspection

A controlled comparison between the automated detector and manual inspection was not performed in the completed experiment. The available experimental evidence therefore does not support numerical claims concerning manual inspection accuracy, inspection time, or repeatability. The available automated and manual inspection measurements are summarized in Table 13.

**Table 13. Availability of automated and manual inspection measurements**

| **Evaluation criterion** | **YOLOv8n** | **Manual inspection** |
|---|---|---|
| Precision | 71.8% | Not measured |
| Recall | 73.1% | Not measured |
| F1-score | 72.4%* | Not measured |
| mAP@50 | 76.3% | N/A |
| Inference time | 4.3 ms/image | Not measured |
| Complete inspection-cycle time | Not measured | Not measured |
| Repeatability | Not measured | Not measured |

*Calculated from the reported overall precision and recall.

This distinction is important because the present experiment validates the computer-vision detection component rather than providing a direct human-versus-robot production comparison.

## 5.8 Comparison with Alternative Deep Learning Models

Only YOLOv8n was experimentally evaluated in the completed study. Although other detectors were considered in the original methodology, no corresponding experimental results were generated for Faster R-CNN or another state-of-the-art detector. The experimentally verified model comparison is summarized in Table 14.

**Table 14. Experimentally verified model comparison**

| Model | Precision (%) | Recall (%) | F1 (%) | mAP@50 (%) | mAP@50–95 (%) | Inference |
|---|---|---|---|---|---|---|
| YOLOv8n | 71.8 | 73.1 | **72.4*** | 76.3 | 42.9 | 4.3 ms/image |
| Faster R-CNN | Not evaluated | Not evaluated | Not evaluated | Not evaluated | Not evaluated | Not evaluated |
| SOTA detector | Not evaluated | Not evaluated | Not evaluated | Not evaluated | Not evaluated | Not evaluated |

*Calculated from overall precision and recall.

Rather than assigning estimated values to unexecuted models, the comparison is restricted to experimentally verified results. This prevents literature-derived performance figures from being incorrectly presented as results obtained under the same experimental conditions.

## 5.9 Summary of Experimentally Verified Results

The principal experimentally verified results are consolidated in Table 15.

**Table 15. Summary of experimentally verified results**

| Parameter | Result |
|---|---|
| Model | YOLOv8n |
| Dataset | NEU-DET |
| Test images | 180 |

| | |
|---|---|
| Test instances | 412 |
| Number of defect classes | 6 |
| Precision | 71.8% |
| Recall | 73.1% |
| Calculated F1-score | 72.4% |
| mAP@50 | 76.3% |
| mAP@50-95 | 42.9% |
| Preprocessing | 3.0 ms/image |
| Inference | 4.3 ms/image |
| Postprocessing | 2.0 ms/image |
| Combined computational processing | 9.3 ms/image |

### 5.10 Best and Most Challenging Defect Categories

The class-level analysis identifies significant differences in detection performance. Scratches achieved the highest recall and mAP@50, while patches produced the highest precision. Crazing was the most challenging defect category according to precision, recall, and both mAP measures. The strongest and weakest class-level performance results are summarized in Table 16.

**Table 16. Strongest and weakest class-level performance**

| **Performance indicator** | **Defect category** | **Result** |
|---|---|---|
| Highest precision | Patches | 84.2% |
| Highest recall | Scratches | 98.2% |
| Highest mAP@50 | Scratches | 97.3% |
| Highest mAP@50-95 | Patches | 58.9% |
| Lowest precision | Crazing | 52.9% |
| Lowest recall | Crazing | 42.0% |

| | | |
|---|---|---|
| Lowest mAP@50 | Crazing | 45.1% |
| Lowest mAP@50-95 | Crazing | 16.4% |

Overall, the experimental results demonstrate that the YOLOv8n detector provides a promising computer-vision component for automated surface-defect inspection.The model achieved 71.8% precision, 73.1% recall, 72.4% calculated F1-score, 76.3% mAP@50, and 42.9% mAP@50–95 on the independent NEU-DET test set. Performance was strongest for scratches and patches and comparatively weaker for crazing. The measured 4.3 ms inference time per image also demonstrates low computational latency for the detection stage. Nevertheless, physical robotic-cycle time, repeatability, severity assessment, manual inspection comparison, and comparative performance against alternative detectors remain outside the completed benchmark experiment and should therefore be treated as areas requiring additional validation rather than reported as measured outcomes.

## 6. Discussion

### 6.1 Interpretation of the Experimental Findings

The experimental results demonstrate that the trained YOLOv8n model provides a functional deep-learning-based approach for automated surface-defect detection on the NEU-DET dataset. The model achieved an overall precision of 71.8%, recall of 73.1%, mAP@50 of 76.3%, and mAP@50–95 of 42.9%. These results indicate that the detector was able to identify and classify a substantial proportion of the annotated defect instances while maintaining a moderate level of prediction reliability. The relatively close precision and recall values are particularly important because they indicate a reasonably balanced detection behaviour. The recall of 73.1% shows that the model successfully identified a substantial proportion of the defects present in the test images, while the precision of 71.8% indicates that most of the predicted defect detections corresponded to actual annotated defects. Nevertheless, the results also show that a proportion of defects remained undetected and that some predicted detections were incorrect. Therefore, the current model should be regarded as a promising computer-vision inspection component rather than a fully validated production-quality inspection system. The calculated F1-score of 72.4% further supports this interpretation. Because the F1-score provides a combined representation of precision and

recall, the result indicates that the model maintained a moderate balance between detecting defective regions and limiting incorrect detections. This balance is important for automated manufacturing inspection because excessive false positives can increase unnecessary rejection or reinspection, whereas excessive false negatives may allow defective components to pass through the production process.

### 6.2 Class-Level Performance Analysis

The class-level results reveal substantial differences in detection performance among the six defect categories. Scratches produced the strongest overall detection behaviour, achieving 81.2% precision, 98.2% recall, and 97.3% mAP@50. The exceptionally high recall indicates that the detector was able to identify most of the annotated scratch instances in the independent test set. Patches also demonstrated strong performance, with 84.2% precision, 89.8% recall, 92.5% mAP@50, and 58.9% mAP@50–95. The relatively high precision suggests that predictions associated with patches were comparatively reliable, while the high recall indicates that most annotated patch defects were successfully detected. In contrast, crazing represented the most challenging defect category. Its precision of 52.9%, recall of 42.0%, mAP@50 of 45.1%, and mAP@50–95 of 16.4% were substantially lower than those obtained for the other categories. This indicates that the visual characteristics of crazing presented greater difficulty for the trained detector. The lower recall is particularly significant because it indicates that a considerable proportion of crazing instances were not successfully detected. The difference between the strongest and weakest classes demonstrates that aggregate performance alone does not fully describe the behaviour of the inspection model. A detector may achieve acceptable overall metrics while still exhibiting weaknesses for specific defect types. Consequently, class-level evaluation is essential when considering the model for manufacturing inspection, particularly where certain defect categories may have greater quality or engineering significance than others.

### 6.3 Localization Performance

The difference between mAP@50 and mAP@50–95 provides additional insight into the localization capability of the trained detector. The overall mAP@50 of 76.3% was substantially

higher than the mAP@50–95 value of 42.9%. This indicates that detection performance decreased when stricter localization criteria were applied across multiple IoU thresholds. The class-level results demonstrate the same pattern. Scratches achieved 97.3% mAP@50 and 58.1% mAP@50–95, while patches achieved 92.5% and 58.9%, respectively. Crazing again produced the weakest results, with 45.1% mAP@50 and only 16.4% mAP@50–95. These findings suggest that the model was generally capable of identifying the presence of defects but experienced greater difficulty producing highly accurate bounding-box localization under stricter overlap requirements. This distinction is relevant to robotic inspection because reliable defect localization is necessary when the inspection system must identify the precise physical region requiring further examination or corrective action. A separate mean-IoU value was not produced by the completed experiment. Therefore, no independent numerical IoU result is claimed in this study. The localization discussion is instead based on the mAP@50 and mAP@50–95 values directly obtained from the YOLOv8n evaluation.

### 6.4 Computational Efficiency and Industrial Relevance

The computational results provide evidence of low processing latency at the computer-vision stage. The measured preprocessing, inference, and postprocessing times were 3.0, 4.3, and 2.0 ms per image, respectively, giving a combined computational processing time of approximately 9.3 ms per image. The 4.3 ms inference time is particularly relevant to the proposed robotic inspection framework because rapid model execution is desirable in automated manufacturing environments where inspection decisions must be generated without introducing unnecessary processing delays. The relatively short inference time also supports the selection of a lightweight YOLOv8n architecture for applications where computational efficiency is an important consideration. However, the measured processing time should not be interpreted as the complete inspection-cycle time of the robotic system. The experiment did not measure robot movement, component positioning, camera triggering, conveyor operation, communication delays, or reporting time. Consequently, the present results establish the computational efficiency of the detection stage but do not establish the overall throughput of the complete robotic inspection system. This distinction is important when interpreting the practical implications of the results. A low neural-network inference time does not automatically guarantee real-time operation of the entire manufacturing

inspection cell. Complete production-cycle performance would require additional measurements involving the physical robotic and vision components.

### 6.5 Implications for the Proposed Robotic Inspection Framework

The experimental findings support the use of YOLOv8n as the defect-detection component within the proposed intelligent robotic inspection framework. The model demonstrated the ability to detect multiple defect categories while providing comparatively low computational latency. These characteristics are compatible with a framework in which image acquisition, defect detection, localization, and subsequent quality decisions are integrated into an automated inspection workflow. The results also demonstrate the importance of maintaining a distinction between the validated computer-vision component and the broader proposed inspection architecture. The completed experiment directly validated defect detection, classification, localization-related performance, and computational processing time. Other components of the proposed framework, including physical robotic repeatability, severity assessment, complete inspection-cycle time, and human-versus-robot comparison, were not quantitatively evaluated. Accordingly, the experimental evidence supports the feasibility of the detection stage but does not justify claiming that the entire robotic inspection framework has been experimentally validated under production conditions. This distinction strengthens the scientific reliability of the study by ensuring that conclusions remain directly connected to the measurements obtained during the experiment.

### 6.6 Limitations of the Experimental Study

Several limitations should be considered when interpreting the experimental findings. First, the evaluation was conducted using the NEU-DET benchmark dataset rather than a dataset acquired directly from the complete UR10e-based robotic inspection cell. Although the benchmark provides a useful environment for evaluating defect-detection capability, its images do not fully reproduce all physical and environmental variations that may occur in an industrial inspection station. Second, only YOLOv8n was experimentally trained and evaluated. Faster R-CNN and other state-of-the-art detectors were included as potential baseline methods in the methodology but were not executed as part of the completed experiment. Therefore, no direct experimental comparison with those models is claimed. Third, severity assessment was not quantitatively evaluated because the

available dataset did not contain severity-labelled annotations. Although severity assessment remains part of the proposed framework, numerical accuracy or classification performance cannot be established from the current experiment. Fourth, inspection repeatability was not measured using repeated inspections of the same physical components. The available experiment evaluated independent test images rather than repeated robotic inspection cycles. Consequently, conclusions regarding robot positioning repeatability, vision-system repeatability, or inspection-to-inspection consistency cannot be derived from the present results. Finally, a controlled comparison with manual inspection was not performed. The current study therefore does not provide experimental evidence for claiming that the proposed system is more accurate, faster, or more repeatable than human inspection. These limitations do not invalidate the completed experiment. Instead, they define the scope within which the reported results should be interpreted and identify the additional evaluations required before deployment in a production manufacturing environment.

### 6.7 Future Experimental Validation

Future experimental work should extend the current benchmark evaluation toward a complete physical robotic inspection study. A first priority would be to acquire images directly from the integrated robotic inspection platform using the industrial camera and controlled lighting configuration. Such a dataset would enable evaluation under variations in component positioning, illumination, surface condition, and robotic motion. A second priority would be the introduction of severity-labelled defect annotations. Defects could be categorized according to predefined engineering criteria, allowing the proposed framework to estimate defect severity and determine whether individual components satisfy specified quality requirements. Further experiments should also evaluate repeated inspections of identical components. Multiple inspection cycles under controlled conditions would allow prediction consistency, positioning variation, and inspection repeatability to be quantified. Comparative experiments involving alternative detection architectures would also strengthen the evaluation. In particular, experimentally training and testing Faster R-CNN or another contemporary detector using the same dataset and evaluation protocol would provide a more rigorous basis for assessing the relative performance of YOLOv8n. Finally, future studies should measure the complete robotic inspection cycle, including robot movement, image acquisition, inference, decision generation, conveyor operation, and

communication. Such measurements would provide a more realistic assessment of manufacturing throughput than neural-network inference time alone.

### 6.8 Overall Discussion

Overall, the experimental evidence indicates that YOLOv8n provides a viable foundation for automated surface-defect detection within the proposed intelligent robotic inspection framework. The detector achieved 71.8% precision, 73.1% recall, a calculated F1-score of 72.4%, 76.3% mAP@50, and 42.9% mAP@50–95 on the independent NEU-DET test set. The strongest results were obtained for scratches and patches, whereas crazing represented the principal detection challenge. The results also demonstrate that computational efficiency is a strength of the evaluated model, with an inference time of 4.3 ms per image. However, this measurement applies only to the computer-vision processing stage and should not be interpreted as the complete cycle time of the robotic inspection system. Most importantly, the findings establish a clear boundary between what was experimentally demonstrated and what remains to be validated. The current study successfully evaluates the YOLOv8n defect-detection component, while severity assessment, physical repeatability, complete robotic-cycle performance, manual inspection comparison, and alternative-model benchmarking remain future validation requirements. Maintaining this distinction ensures that the discussion and conclusions remain evidence-based while providing a clear pathway for extending the proposed framework toward a fully validated industrial inspection system.

## 7. Engineering Recommendations

The implementation of intelligent robotic inspection systems requires consideration beyond the performance of the defect detection model. Successful industrial deployment depends on reliable image acquisition, accurate robotic positioning, controlled illumination, computational efficiency, system integration, maintenance, and continuous performance monitoring. Based on the proposed inspection framework and the experimentally verified YOLOv8n results, the following engineering recommendations are proposed for the development and deployment of autonomous robotic inspection systems in advanced manufacturing environments.

### 7.1 Camera Placement and Imaging Configuration

Camera placement should be treated as a fundamental component of inspection-system design because the quality of acquired images directly influences defect detection and localization performance. The camera should be installed at a fixed and mechanically stable position relative to the inspection workspace to minimize changes in viewpoint, vibration, and image geometry during repeated inspections. The camera-to-component distance should be selected according to the dimensions of the inspection target and the minimum defect size that must be detected. Excessive distance may reduce the visibility of small defects, whereas an excessively short distance may restrict the field of view. Appropriate lens selection and camera positioning should therefore provide sufficient spatial resolution while maintaining adequate coverage of the inspection region. The results obtained in this study further emphasize the importance of image quality because detection performance varied considerably between defect categories. Scratches and patches were detected substantially more effectively than crazing. Consequently, imaging configurations should be optimized particularly for visually difficult defect categories rather than selected solely according to overall image quality. For components with complex geometries or regions that cannot be adequately captured from a single viewpoint, multiple synchronized cameras may be considered. Multi-view imaging can reduce occlusion and blind spots and may improve the reliability of defect localization.

### 7.2 Robotic Calibration and Motion Planning

Accurate robotic calibration is essential for maintaining consistent inspection geometry. The robotic manipulator, camera, and inspection workspace should be calibrated so that image acquisition occurs from repeatable positions and orientations. Hand-eye calibration should be performed where robotic positioning and camera coordinates must be related. Accurate spatial calibration is particularly important when defect locations identified in an image are subsequently used for physical inspection, verification, measurement, or automated intervention. Inspection trajectories should be designed to minimize unnecessary robotic movement while ensuring complete coverage of the component. Excessive movement increases inspection-cycle time and may contribute to mechanical wear without improving inspection quality. Smooth trajectories should also be preferred because sudden movements or vibration during image acquisition can negatively affect image quality and localization. Robot and conveyor operation should be synchronized so that images are acquired when the component is stationary or moving within an

acceptable tolerance. Proper synchronization can reduce motion blur and improve consistency between successive inspection images.

### 7.3 Illumination System Design

Illumination should be carefully designed because surface defects may be difficult to distinguish when affected by shadows, reflections, glare, or non-uniform brightness. Controlled illumination should therefore be maintained throughout the inspection process. Diffuse LED illumination is recommended for many metallic manufacturing components because it can reduce harsh reflections and provide more uniform lighting. Depending on the geometry and surface characteristics of the inspected component, dome, ring, bar, or coaxial lighting configurations may be considered. Inspection lighting should be isolated from uncontrolled ambient illumination whenever possible. An enclosed or partially enclosed inspection area can reduce variations caused by changes in factory lighting conditions and improve the consistency of acquired images. Lighting equipment should also form part of the preventive-maintenance program. Dust, contamination, lens degradation, or changes in illumination intensity can gradually alter image characteristics and consequently affect model performance.

### 7.4 Model Maintenance and Retraining Strategy

The experimentally observed variation between defect categories demonstrates the importance of continuous model monitoring. Although YOLOv8n achieved an overall precision of 71.8% and recall of 73.1%, the recall for crazing was only 42.0%, indicating that individual defect categories may require additional training data or improved representation. Model performance should therefore be monitored using class-level precision, recall, F1-score, mAP, false-positive rates, and false-negative rates. Monitoring only aggregate performance may conceal deterioration in specific defect categories. A performance-driven retraining strategy is recommended rather than relying exclusively on fixed retraining intervals. New production images should be periodically reviewed to identify previously unseen defect patterns, changes in surface appearance, illumination conditions, or manufacturing-process variations. Relevant samples can then be incorporated into an updated training dataset. Model version control should also be maintained. Each deployed model should have an identifiable version, training dataset, configuration, and validation record.

Updated models should be validated before deployment, while previous validated versions should be retained to permit recovery if an updated model produces unacceptable performance.

### 7.5 Industrial Deployment Strategy

Deployment of the proposed inspection framework should begin with controlled pilot testing before integration into continuous production. Pilot deployment allows engineers to identify practical issues associated with camera positioning, robot motion, communication, illumination, processing latency, and component handling without immediately exposing the production process to unnecessary operational risk. The inspection system should be designed for integration with existing industrial control and information systems where required. Communication with programmable logic controllers, manufacturing execution systems, quality databases, and other production systems can support automated recording of inspection decisions and improve product traceability. Modular system architecture is also recommended. Separating the robotic, imaging, processing, and decision-making components allows individual subsystems to be upgraded without requiring complete redesign of the inspection cell. Because network-connected inspection systems may exchange production and quality information, appropriate cybersecurity controls should also be implemented. Access control, secure authentication, protected communications, and controlled software-update procedures should form part of the deployment strategy.

### 7.6 System Maintenance and Performance Monitoring

Long-term inspection reliability requires preventive maintenance of both hardware and software. Maintenance activities should include inspection of robotic components, camera mounts, lighting equipment, conveyor mechanisms, communication interfaces, and computing hardware. Performance monitoring should extend beyond model accuracy. Useful operational indicators include processing time, system availability, communication latency, robot positioning accuracy, image-quality indicators, false-positive trends, and false-negative trends. Particular attention should be given to the distinction between neural-network inference time and complete inspection-cycle time. The present experiment measured a YOLOv8n inference time of 4.3 ms/image and a combined computational processing time of approximately 9.3 ms/image. However, these measurements exclude robot motion, image acquisition, conveyor operation, communication, and

reporting. Therefore, complete system throughput should be measured separately before industrial deployment. Periodic validation using reference components with known defect characteristics is also recommended. Such validation can determine whether the complete inspection system continues to operate within predefined quality requirements.

## 8. Conclusion

This study presented an intelligent robotic inspection framework for autonomous manufacturing-defect detection and automated quality inspection. The framework integrates robotic manipulation, industrial machine vision, controlled image acquisition, deep-learning-based defect detection, defect localization, and quality decision-making within a unified inspection architecture. The experimental component of the study evaluated a YOLOv8n detector using the independent NEU-DET test set containing 180 test images and 412 annotated defect instances across six defect categories: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. The model achieved an overall precision of 71.8%, recall of 73.1%, a calculated F1-score of 72.4%, mAP@50 of 76.3%, and mAP@50–95 of 42.9%. The class-level analysis demonstrated that detection performance varied considerably according to defect type. Scratches produced the strongest recall and mAP@50 performance, with values of 98.2% and 97.3%, respectively. Patches produced the highest precision at 84.2% and also achieved strong localization-related performance. In contrast, crazing represented the most challenging defect category, achieving 52.9% precision, 42.0% recall, 45.1% mAP@50, and 16.4% mAP@50–95. These differences demonstrate that overall model performance can conceal substantial class-level variation and that difficult defect categories require particular attention during further model development. The computational evaluation also demonstrated relatively low processing latency. YOLOv8n required approximately 4.3 ms/image for inference, while preprocessing and postprocessing required 3.0 and 2.0 ms/image, respectively. The resulting combined computational processing time was approximately 9.3 ms/image. This result indicates that the selected lightweight detector provides a computationally efficient foundation for the vision-processing component of an automated inspection system. However, the findings must be interpreted within the scope of the completed experiment. The current study experimentally validated defect detection, classification, localization-related performance, and computational processing time using the NEU-DET benchmark dataset. Quantitative severity

assessment, repeated physical robotic inspections, complete robotic inspection-cycle time, controlled comparison with manual inspection, and experimental comparison with Faster R-CNN or other state-of-the-art detectors were not conducted. Consequently, these aspects are not presented as experimentally demonstrated outcomes. The study therefore establishes a validated foundation for the computer-vision component of an intelligent robotic inspection framework rather than claiming complete production-level validation of the entire robotic inspection cell. Further experimental work should focus on collecting images directly from the physical robotic inspection platform, expanding the dataset to include greater environmental and manufacturing variation, introducing severity-labelled defect data, measuring repeated robotic inspection cycles, and evaluating the complete inspection-cycle time. Overall, the results demonstrate that YOLOv8n can provide an effective and computationally efficient foundation for automated surface-defect detection. When integrated with appropriately designed robotic positioning, controlled illumination, industrial imaging, calibration, and system-level monitoring, such a detection system has potential to support autonomous quality inspection in advanced manufacturing environments. The proposed framework consequently provides a practical foundation for future development toward more comprehensive, reliable, and production-ready intelligent robotic inspection systems.

**Declarations:**

a. Funding: This research received no specific grant or external financial support from anywhere. The work was conducted using institutional resources

b. Conflicts of Interest / Competing Interests: Not applicable.

**References**


1) Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. *Advances in Neural Information Processing Systems*, 25, 1097–1105.
2) Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. *International Conference on Learning Representations (ICLR).*
3) He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.
4) Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards real-time object detection with region proposal networks*. *Advances in Neural Information Processing Systems*, 28.
5) Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, real-time object detection*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
6) Redmon, J., & Farhadi, A. (2018). *YOLOv3: An incremental improvement*. arXiv:1804.02767.
7) Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). *YOLOv4: Optimal speed and accuracy of object detection*. arXiv:2004.10934.
8) Jocher, G., Chaurasia, A., & Qiu, J., et al. (2023). *YOLO by Ultralytics*. GitHub Repository.
9) Carion, N., Massa, F., Synnaeve, G., et al. (2020). *End-to-End object detection with transformers*. *European Conference on Computer Vision (ECCV).*
10) Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
11) LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning*. *Nature*, 521(7553), 436–444.
12) Lee, J., Bagheri, B., & Kao, H. A. (2015). *A cyber-physical systems architecture for Industry 4.0-based manufacturing systems*. *Manufacturing Letters*, 3, 18–23.
13) Lu, Y. (2017). *Industry 4.0: A survey on technologies, applications and open research issues*. *Journal of Industrial Information Integration*, 6, 1–10.
14) Tao, F., Qi, Q., Liu, A., & Kusiak, A. (2018). *Data-driven smart manufacturing*. *Journal of Manufacturing Systems*, 48, 157–169.

15) Kusiak, A. (2018). *Smart manufacturing*. *International Journal of Production Research*, 56(1–2), 508–517.

16) Wang, L., Törngren, M., & Onori, M. (2015). *Current status and advancement of cyber-physical systems in manufacturing*. *Journal of Manufacturing Systems*, 37, 517–527.

17) Villalonga, A., Beruvides, G., Castano, F., & Haber, R. E. (2021). *Industrial cyber–physical system for condition-based monitoring in smart manufacturing*. *Sensors*, 21(2), 448.

18) Patrashko, D. Y., & Gurau, V. (2026). *Machine Learning-Powered Vision for Robotic Inspection in Manufacturing: A Review*. *Sensors*, 26(3), 788.

19) Ma, Y., Yin, J., Huang, F., et al. (2024). *Surface defect inspection of industrial products with object detection deep networks: A systematic review*. *Artificial Intelligence Review*.

20) Papavasileiou, A., Michalos, G., & Makris, S. (2025). *Quality control in manufacturing: Review and challenges on robotic applications*. *International Journal of Computer Integrated Manufacturing*.

21) Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). *Densely connected convolutional networks*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4700–4708.

22) Tan, M., & Le, Q. V. (2020). *EfficientNet: Rethinking model scaling for convolutional neural networks*. Proceedings of the International Conference on Machine Learning (ICML), 6105–6114.

23) Tan, M., Pang, R., & Le, Q. V. (2020). *EfficientDet: Scalable and efficient object detection*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10781–10790.

24) Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). *Focal loss for dense object detection*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2980–2988.

25) Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). *The Pascal Visual Object Classes (VOC) Challenge: A retrospective*. *International Journal of Computer Vision*, 111(1), 98–136.

26) Lin, T.-Y., Maire, M., Belongie, S., et al. (2014). *Microsoft COCO: Common objects in context*. European Conference on Computer Vision (ECCV), 740–755.

27) Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019). *MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9592–9600.

28) Song, K., Yan, Y., et al. (2020). *Review of automatic defect detection for industrial products using vision-based deep learning*. *IEEE Access*, 8, 73356–73378.

29) Zhang, C., Ding, Y., Shi, Y., et al. (2021). *Deep learning-based surface defect detection for industrial products: A survey*. *Journal of Manufacturing Systems*.

30) Ma, Y., Yin, J., Huang, F., et al. (2024). *Surface defect inspection of industrial products with object detection deep networks: A systematic review*. *Artificial Intelligence Review*.

31) Carvalho, P., Durupt, A., & Grandvalet, Y. (2023). *A review of benchmarks for visual defect detection in the manufacturing industry*. arXiv.

32) Agarwal, A., Ajith, A., Wen, C., et al. (2023). *Robotic defect inspection with visual and tactile perception for large-scale components*. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2023)*, 10110–10116.

33) Papavasileiou, A., Michalos, G., & Makris, S. (2025). *Quality control in manufacturing: Review and challenges on robotic applications*. *International Journal of Computer Integrated Manufacturing*.

34) Patrashko, D. Y., & Gurau, V. (2026). *Machine Learning-Powered Vision for Robotic Inspection in Manufacturing: A Review*. *Sensors*, 26(3), 788.

35) Chen, H., & Cheng, Y. (2026). *Research on testing framework and methods for industrial machine vision inspection systems*. *Machine Intelligence and Digital Applications*.

36) Tayyab, M. A., Elahi, H., Ali, M. O., et al. (2026). *AI enabled pulse-echo based SHM system integrated with robotic gripper for the inspection of pipelines*. *Scientific Reports*, 16, Article 18560.

37) Johansson, B., et al. (2023). *Lessons from adopting robotic in-line quality inspection in the Swedish manufacturing industry*. *Procedia Computer Science*, 217, 386–394.

38) Romero, D., Bernus, P., Noran, O., Stahre, J., & Fast-Berglund, Å. (2016). *The Operator 4.0: Human cyber-physical systems and adaptive automation*. *IFIP Advances in Information and Communication Technology*, 488, 677–686.

39) Tao, F., Zhang, M., Liu, Y., & Nee, A. Y. C. (2019). *Digital Twin Driven Smart Manufacturing*. Academic Press.

40) Xu, L. D., Xu, E. L., & Li, L. (2018). *Industry 4.0: State of the art and future trends*. *International Journal of Production Research*, 56(8), 2941–2962.

41) Lu, L., Hou, J., Yuan, S., Yao, X., Li, Y., & Zhu, J. (2023). *Deep learning-assisted real-time defect detection and closed-loop adjustment for additive manufacturing of continuous fiber-reinforced polymer composites*. *Robotics and Computer-Integrated Manufacturing*, 79, 102431.

42) Li, W., Zhang, H., Wang, G., Xiong, G., Zhao, M., Li, G., & Li, R. (2023). *Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing*. *Robotics and Computer-Integrated Manufacturing*, 80, 102470.

43) Wang, J., Gao, P., Zhang, J., Lu, C., & Shen, B. (2023). *Knowledge augmented broad learning system for computer vision based mixed-type defect detection in semiconductor manufacturing*. *Robotics and Computer-Integrated Manufacturing*, 81, 102513.

44) Agarwal, A., Ajith, A., Wen, C., Stryzheus, V., Miller, B., Chen, M., Johnson, M. K., Rincon, J. L. S., Rosca, J., & Yuan, W. (2023). *Robotic defect inspection with visual and tactile perception for large-scale components*. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

45) Carvalho, P., Durupt, A., & Grandvalet, Y. (2023). *A review of benchmarks for visual defect detection in the manufacturing industry*. arXiv:2305.13261.

46) Song, K., Yan, Y., et al. (2020). *Review of automatic defect detection for industrial products using vision-based deep learning*. *IEEE Access*, 8, 73356–73378.

47) Zhang, C., Ding, Y., Shi, Y., et al. (2024). *Deep learning-based surface defect detection for industrial products: Recent advances and challenges*. *Journal of Manufacturing Systems*.

48) Ma, Y., Yin, J., Huang, F., et al. (2024). *Surface defect inspection of industrial products with object detection deep networks: A systematic review*. *Artificial Intelligence Review*.

49) Patrashko, D. Y., & Gurau, V. (2026). *Machine Learning-Powered Vision for Robotic Inspection in Manufacturing: A Review*. *Sensors*, 26(3), 788.

50) Papavasileiou, A., Michalos, G., & Makris, S. (2025). *Quality control in manufacturing: Review and challenges on robotic applications*. *International Journal of Computer Integrated Manufacturing*.

51) Cheng, Y., Cao, Y., Yao, H., Luo, W., Jiang, C., Zhang, H., & Shen, W. (2025). *A comprehensive survey for real-world industrial defect detection: Challenges, approaches, and prospects*. arXiv:2507.13378.

52) Zi, B., Tang, K., Li, Y., Feng, K., Liu, Y., & Wang, L. (2026). *Coating defect detection in intelligent manufacturing: Advances, challenges, and future trends*. *Robotics and Computer-Integrated Manufacturing*, 97, 103079.

53) Chen, H., & Cheng, Y. (2026). *Research on testing framework and methods for industrial machine vision inspection systems*. *Machine Intelligence and Digital Applications*.

54) Somtochukwu, E. E., & Rijal, N. (2026). *Real-time industrial defect detection on edge hardware using fine-tuned YOLOv8: A systematic benchmark on the NEU surface defect database and MVTec AD*. arXiv:2606.07659.

55) An, S., Wu, J., & Li, J. (2026). *Unsupervised industrial image defect detection based on autoencoder and GANs*. *PLOS ONE*.

56) Tayyab, M. A., et al. (2026). *AI enabled pulse-echo based SHM system integrated with robotic gripper for the inspection of pipelines*. *Scientific Reports*, 16.

57) Savaş, S. (2026). *Statistical feature engineering for robot failure detection: A comparative study of machine learning and deep learning classifiers*. *Sensors*, 26(5), 1649.

58) El Kalach, F., Farahani, M. A., Samaha, P., Wuest, T., & Harik, R. (2026). *Digital twin enabled robot collision detection using time series forecasting*. *Journal of Intelligent Manufacturing*.

59) Yazdani, M. H., & Kim, H. S. (2026). *Unsupervised learning for industrial robot health monitoring: Trends, techniques, and challenges*. *Mathematics*, 14(13), 2397.

60) Ai, Y., Cheng, J., Lai, J., Peng, L., Guan, Q., & Yu, Z. (2026). *Zero-shot industrial defect detection in flexible manufacturing via general-expert prompt fusion and spatial interaction*. Engineering Applications of Artificial Intelligence, 178(Part 1), 114999. https://doi.org/10.1016/j.engappai.2026.114999

61) Leng, J., Wang, J., Zhou, L., Zhao, R., Chen, C., Zhang, D., Zheng, S., Liu, Q., & Shen, W. (2026). *A review of multi-modal deep learning towards agentic smart manufacturing*. Advanced Engineering Informatics, 70, 104117. https://doi.org/10.1016/j.aei.2025.104117.

62) Xu, Q., Zhang, C., Ma, D., Li, J., Leng, J., & Zhou, G. (2026). *Automated multimodal process knowledge graph construction for intelligent process planning with Cross-Modal Transformers*. Robotics and Computer-Integrated Manufacturing, 98, 103141. https://doi.org/10.1016/j.rcim.2025.103141.

63) Li, Y., Zhao, H., Jiang, H., Pan, Y., Liu, Z., Wu, Z., et al. (2026). *Large Language Models for manufacturing*. Journal of Manufacturing Systems. https://doi.org/10.1016/j.jmsy.2026.02.014.

64) Lee, J., Su, H., Macchi, M., Polenghi, A., Wu, W., Zhao, Z., et al. (2026). *2026 Roadmap on Artificial Intelligence and Machine Learning for Smart Manufacturing*. arXiv:2605.00839.

65) Gill, S. S., Golec, M., Hu, J., Xu, M., Du, J., Wu, H., et al. (2024). *Edge AI: A taxonomy, systematic review and future directions*. arXiv:2407.04053.

66) Piñera-Marín, A., et al. (2026). *Local LLMs for industrial supervision and control: An edge AI event-driven architecture for proactive operational context management in real industrial environments*. Electronics, 15(12), 2547.

67) Peças, P., Oliveira, L. C., Febra, R., Duarte, S., Cruz-Machado, V., Mira, T., et al. (2026). *Non-intrusive edge-based micro-stop detection using hybrid learning in manufacturing systems*. The International Journal of Advanced Manufacturing Technology.

68) Xie, J., & Zhao, Y. F. (2026). *Toward edge-cloud integration of foundation models and AI agents in additive manufacturing*. Engineering Science of Advanced Manufacturing, 2(2).

69) Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019). *MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 9592–9600.

70) Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). *Feature Pyramid Networks for object detection*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2117–2125.

71) Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). *DeepLab: Semantic image segmentation with deep convolutional networks, atrous convolution, and fully connected CRFs*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4), 834–848.

72) Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional networks for biomedical image segmentation*. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241.

73) Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully convolutional networks for semantic segmentation*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3431–3440.

74) Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2021). *An image is worth 16×16 words: Transformers for image recognition at scale*. *International Conference on Learning Representations (ICLR).*

75) Liu, Z., Lin, Y., Cao, Y., et al. (2021). *Swin Transformer: Hierarchical vision transformer using shifted windows*. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 10012–10022.

76) Kirillov, A., Mintun, E., Ravi, N., et al. (2023). *Segment Anything*. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 4015–4026.

77) He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask R-CNN*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2961–2969.

78) Zhou, X., Wang, D., & Krähenbühl, P. (2019). *Objects as points*. arXiv:1904.07850.

79) Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal speed and accuracy of object detection*. arXiv:2004.10934.

80) Jocher, G., Chaurasia, A., & Qiu, J., et al. (2023). *Ultralytics YOLO*. GitHub repository. https://github.com/ultralytics/ultralytics