Intelligent Detection of Container Surface Damage Based on Scene Self-Adaptation and Cascaded Network

Ruchen Li, Zixin Li, Shucong Liu, Yiru Wang, and Xinyi Yang

***Abstract*—To address the issues of variable lighting, category confusion, and sample imbalance in container surface damage detection under complex port environments, this paper proposes an intelligent detection model integrating scene self-adaptation preprocessing and cascaded deep learning. First, a scene self-adaptation preprocessing module based on low-level visual features is designed to automatically identify four typical scenes (sunny, cloudy, night, rainy) and perform customized image enhancement, thus improving the model's robustness against complex lighting conditions. Second, a cascaded multi-label classification network with ResNet-50 as the backbone is constructed, and lightweight sub-classifiers are added to achieve refined discrimination of easily confused damage categories (scratch-rust-damage and damage-hole); a two-stage dynamic fusion architecture is further proposed to synergize the cascaded classifier and YOLOv8n-seg under a confidence-driven strategy. Experiments verify the model robustness through 5-fold cross-validation and 3 groups of random seed experiments, with results obtained on the public Port Container Surface Damage Detection Dataset containing 1274 images, where the confidence-driven dynamic fusion adopts dual thresholds (τ_h=0.8, τ_l=0.4) and weighted coefficients (w_1=0.6, w_2=0.4) for model complementation. Experimental results show that the classification accuracy reaches 87.60%, the instance segmentation mAP@0.5 reaches 0.744, and the inference speed is 35 FPS, significantly outperforming mainstream methods (e.g., Mask R-CNN, YOLACT) in comprehensive accuracy-efficiency performance.**

***Key Words*—Container Surface Damage Detection, Scene Self-Adaptation Preprocessing, Cascaded Deep Learning, ResNet-50, YOLOv8n-seg**

I. Introduction

As the core carrier of the global logistics chain, the structural integrity of the outer surface of the container directly determines the safety of cargo transportation and the efficiency of port operations [1]. During long-term use, containers are prone to seven common types of damage, such as holes, damage, and rust, due to factors such as mechanical collisions, environmental corrosion, and structural fatigue [2]. Traditional artificial vision detection methods are inefficient and subjective, making it difficult to meet the operational needs of modern port automation and high throughput. The rise of Computer Vision and Deep Learning technology provides an effective solution for intelligent detection of container damage [3]. This paper focuses on the automatic detection and classification of seven types of damage on the outer surface of containers [4]. In response to core challenges such as complex natural lighting, visually similar damage, and non-standard data annotation, it constructs an efficient and intelligent detection system to facilitate the automated upgrade of port inspections [5]. Our work complements recent real-time container tracking and damage detection systems at seaports [6] by providing enhanced classification accuracy through cascaded networks for easily confused damage categories.

In recent years, with the booming development of global trade, containers, as the core carriers of modern logistics, their structural integrity directly affects cargo safety and turnover efficiency [7]. The introduction of intelligent inspection systems is a critical path to improving port automation levels and achieving cost reduction and efficiency improvement [8]. In the intelligent operation and maintenance system of containers, the rapid, accurate automatic detection and classification of various damages on the outer surface are the primary prerequisites for ensuring the safe service of containers and avoiding cargo losses.

However, the actual port operation scenario is complex, and image acquisition is affected by lighting factors such as day-night alternation, sunny, cloudy, rainy, and snowy conditions, resulting in problems such as uneven brightness and noise interference. Moreover, the visual features of the seven types of damage are similar, which easily leads to misjudgment by the model and restricts the improvement of detection accuracy [9]. The core reason is that existing models lack scene self-adaptation ability and have not been optimized for confusing categories.

In response to the above issues, existing research falls into two categories: in terms of illumination optimization, image enhancement or data augmentation methods are mainly used to improve robustness [10, 11]; in terms of category confusion, feature representation is often enhanced by strengthening the backbone network or introducing an attention mechanism [12]. However, most existing methods separate preprocessing from

Manuscript received February 16, 2026; revised February 23 and March 2, 2026; accepted March 20, 2026. This article was recommended for publication by Associate Editor Shujin Qin upon evaluation of the reviewers' comments.

Ruchen Li, Zixin Li and Shucong Liu are with the School of Software, Shangqiu Normal University, Shangqiu 476000, China (e-mail: 2742770300@qq.com, 3065909579@qq.com, 574529713@qq.com).

Yiru Wang is with the School of Economics and Management, Shangqiu Normal University, Shangqiu 476000, China (e-mail: 565339792@qq.com).

Xinyi Yang is with the Faculty of Business, Lingnan University, Hong Kong, 000000, China (e-mail: yxy13022870968@163.com).

Corresponding Author: Xinyi Yang.

model design, lack targeted optimization, and struggle to balance accuracy and efficiency [13].

To address the aforementioned bottlenecks, this paper proposes an intelligent detection model for container surface damage, with the core innovation points as follows:

1) Design a scene Self-Adaptation preprocessing module based on low-level visual features to automatically recognize four types of scenes, namely "sunny, cloudy, night, and rainy", and apply customized enhancement .
2) Construct a cascaded multi-label classification network (hereinafter referred to as CCN, Cascaded Multi-label Classification Network), and add sub-classifiers focusing on the confusion sets of "scratch-rust-damage" and "damage-hole" to achieve refined discrimination.
3) Propose a two-stage dynamic fusion architecture to synergistically leverage the positioning advantage of YOLOv8n-seg and the classification advantage of the cascaded classifier, thereby simultaneously improving the accuracy of detection and segmentation.

This research is not an independent innovation of a single module, but aims at the industrial needs of container damage detection in complex port scenarios. It adopts an end-to-end integrated design of scene adaptive preprocessing, cascaded classification, and dynamic fusion detection-segmentation architecture, solving three core industrial pain points: variable lighting, category confusion, and sample imbalance, and realizing the coordinated optimization of detection accuracy and real-time performance.

The subsequent structure of this paper is as follows: Chapter 2 provides a detailed description and analysis of the multi-label recognition problem of container damage; Chapter 3 elaborates on the overall architecture and core module design of the proposed model; Chapter 4 validates the effectiveness of the model through experiments; Chapter 5 summarizes the entire paper and looks ahead to future research directions.

II. Problem Description And Preliminaries

In the context of the rapid development of global trade, intelligent ports and automated container logistics, the real-time and accurate perception of container surface damage has become an important part of ensuring transportation safety, improving operational efficiency and reducing maintenance costs [14]. Traditional container damage detection mainly relies on manual inspection, which is easily affected by human factors such as fatigue, experience difference and subjective judgment, resulting in problems such as low efficiency, high cost, high missed detection rate and false detection rate. With the wide application of computer vision and deep learning in industrial inspection, visual-based intelligent damage detection has gradually become a research hotspot in the field of container transportation safety. However, in actual port scenarios, the collected container images are often accompanied by complex and changeable background interference, large differences in damage scales and shapes, unbalanced distribution of positive and negative samples and various damage categories, as well as high similarity between different damage types, which bring great challenges to the accuracy and robustness of damage detection models.

Aiming at the actual engineering requirements of container damage detection in the process of transportation and handling, this paper focuses on a series of typical problems existing in real port scene images, including complex and cluttered background environment, large scale variation of damage areas from tiny scratches to large-area deformation, extremely unbalanced sample number among different damage categories, and high inter-class similarity that easily leads to confusion in category recognition. On the basis of the preliminary competition stage, in which the basic tasks including binary classification of container damage existence, coarse localization of damage regions, recognition of damage categories and multi-dimensional quantitative evaluation of model performance have been completed, this paper further carries out in-depth research on model optimization and reconstruction strategies based on a newly constructed and more comprehensive dataset. The goal is to realize the integrated intelligent identification of damage existence classification, high-precision spatial localization and fine-grained category detection for seven typical types of damages on the outer surface of containers, including cracks, dents, perforations, rust and other common defects, and complete a comprehensive and systematic re-evaluation and verification of the optimized model.

The intelligent visual detection task of container surface damage can be formally defined as a multi-task learning problem in the field of computer vision, which integrates multi-label classification, object detection and instance segmentation, and mainly includes the following two interrelated and mutually promoted sub-tasks:

1) Multi-label image classification task: Given an input container image $I \in \mathbb{R}^{H\times W\times 3}$ (I: container surface image, H: image height, W: image width) with height H, width W and three RGB color channels, the model is required to output a 7-dimensional binary vector $y \in \{0,1\}^7$ (y: binary label vector, 1=damage exists, 0=damage absent). Each dimension in the vector corresponds to one type of damage, where the value 1 means that the damage exists and 0 means that the damage does not exist, so as to realize the simultaneous judgment of the presence or absence of seven kinds of damages.
2) Instance segmentation task: On the basis of judging the existence of damage, if one or more types of damages appear in the image, the model needs to further output complete instance-level information for each independent damage, including the bounding box $B_i = (x_1, y_1, x_2, y_2)$ (B_i: bounding box of the i-th damage, pixel coordinates) that accurately locates the spatial position of the damage area, the class label $c_i \in \{1, 2, ..., 7\}$ (c_i: class label of the i-th damage instance) corresponding to the damage category, and the pixel-level binary segmentation mask $M_i \in \{0,1\}^{H\times W}$ (M_i: segmentation mask of the i-th damage, 1=damage area, 0=background) that depicts the complete contour and shape of the damage.

To facilitate the understanding of the subsequent model, the core symbols are explained in advance (see Appendix for detailed definitions): B_{avg} (average brightness), C_{std} (contrast standard deviation), τ_h (high confidence threshold), τ_l (low confidence threshold), w_1 (segmentation network weight), w_2 (CCN weight).

In practical port application scenarios, the above multi-task damage detection task is confronted with two prominent and critical challenges, which seriously restrict the improvement of model performance [15].

First, the lighting conditions in the port scene are complex and highly variable. In the actual collection process, container images are affected by natural lighting, weather changes and shooting angles, resulting in various interferences such as local overexposure under strong direct sunlight, insufficient image details under low illumination at night, and obvious reflection interference on the ground and container surface in rainy and humid days [16]. These unstable environmental factors often submerge or distort the key visual features that are crucial for damage recognition, such as the unique texture features of rust, the clear edge contours of scratches, the structural characteristics of deformation and the boundary information of holes, which further lead to the offset and distortion of feature distribution in the feature space, reduce the feature extraction ability of the model, and greatly affect the generalization performance in real scenes.

Second, the visual features between different damage categories are highly similar and easily confused. For example, scratches, rust and surface damage have strong ambiguity in color distribution, local texture and morphological structure; deformation and holes are similar in local shape, edge continuity and area distribution [17, 18]. These similarities make the decision boundary of the model between similar categories very blurred, and it is easy to produce a large number of misjudgments in the process of classification and detection, which directly leads to the decline of classification accuracy, detection precision and recall rate, and makes it difficult to meet the high-precision requirements of actual port inspection.

Therefore, aiming at the above practical application requirements and key technical bottlenecks, the core research objective of this paper is to design a lightweight, efficient and end-to-end multi-task damage detection model suitable for port scenes [19]. Our approach complements recent semantic segmentation techniques for container corrosion detection [20] by integrating scene-aware preprocessing for multi-category damage classification. On the premise of ensuring real-time inference performance with frame rate higher than 30 FPS to meet the needs of actual online detection, the model can significantly enhance the feature robustness under complex and changeable lighting conditions, effectively suppress the interference of similar features between categories, and improve the recognition accuracy of easily confused damage types. Ultimately, the model is expected to achieve better comprehensive detection performance than the existing mainstream object detection and segmentation methods, with mAP@0.5 as the main evaluation index for detection tasks and classification F1 score as the key index for classification tasks, so as to provide a feasible and effective technical scheme for intelligent damage detection in actual container transportation.

III. Model Design

A. Overall Architecture and Base Line Model

The model uses ResNet-50[21] and YOLOv8n-seg[22] as the basic framework, and the core improvement is the introduction of SAPM (Scene Adaptive Preprocessing Module), CCN (Cascaded Multi-Label Classification Network) and Two-stage Dynamic Fusion Detection Segmentation Module.

A multi-label classification base line model is constructed using pre-trained ResNet-50 as the backbone network. Its residual connection structure can effectively alleviate the problem of layer disappearance in deep networks and adapt to the complex classification requirements of seven types of defects (holes, damages, rust, scratches, surface deformation, beam deformation, corner column deformation). The specific structure of the network is shown in Table I.

TABLE I Network structure of damage classification based on ResNet-50

Layer	Output size	Param. config	Remarks
Input	$224 \times 224 \times 3$	-	Preprocessed img
Conv	$112 \times 112 \times 64$	7×7 conv, s=2	Initial feature
Pooling	$56 \times 56 \times 64$	3×3 max pool	Down sampling
Res block 1	$56 \times 56 \times 256$	3 bottleneck blocks	Shallow feature
Res block 2	$28 \times 28 \times 512$	4 bottleneck blocks	Middle feature
Res block 3	$14 \times 14 \times 1024$	6 bottleneck blocks	Deep feature
Res block 4	$7 \times 7 \times 2048$	3 bottleneck blocks	Advanced semantic
Global pooling	$1 \times 1 \times 2048$	Adaptive avg pool	Feature vector
FC layer	C	Dropout(0.5) + Linear	Multi-label cls

During the model training process, L2 regularization and a dropout layer (dropout rate 0.2) are introduced, and an early stopping strategy is adopted based on the validation set performance to prevent model overfitting.

The proposed model is trained in an end-to-end manner with a multi-label cross-entropy loss function, which serves as the objective function to optimize the model parameters. The loss function is defined mathematically as (1):

$$L_{multi} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\Big[y_{i,c}\log\left(\sigma(f(x_i)_c)\right) + (1-y_{i,c})\log\left(1-\sigma(f(x_i)_c)\right)\Big]. \tag{1}$$

Among them, $\sigma(\cdot)$ ($\sigma(\cdot)$: Sigmoid activation function, maps output to $[0,1]$) is used to map the original output of the c-th category $f(x_i)_c$ ($f(x_i)_c$: raw network output of the i-th sample for the c-th class) to the $[0,1]$ interval, representing the probability that the sample belongs to this category; $y_{i,c} \in \{0,1\}$ ($y_{i,c}$: ground-truth label of the i-th sample for the c-th class) is the real label of sample x_i on category c, N (N: total number of training samples) is the number of samples, and C (C: total number of damage categories) is the total number of damaged categories ($C = 7$).

To improve the accuracy and robustness of instance segmentation, especially for the segmentation of defects in complex scenarios, we adopt the improved YOLOv8n-seg model for instance segmentation tasks. Specifically, we introduce more

diverse data augmentation strategies to enhance the model's generalization ability and adaptability to defects under different lighting conditions, angles, and scales; meanwhile, a weighted loss function is adopted to alleviate the sample imbalance problem during training and optimize the model's learning effect on small-target defects and hard-to-segment regions. For the instance segmentation training of this improved model, we set reasonable training parameter configurations, and the specific values, setting methods of each parameter as well as their corresponding action mechanisms are detailed in Table II.

TABLE II YOLOv8n-seg enhanced training parameter configuration

Parameter category	Specific configuration	Mechanism of action
Basic settings	epochs=30, batch=16, imgsz=640	Control training scale
Optimizer	AdamW, lr0=0.001, lrf=0.01	Adaptive learning rate adjustment
Spatial augmentation	degrees=10.0, translate=0.1, scale=0.5	Improve geometric robustness
Color augmentation	$hsv_h = 0.015, hsv_s = 0.7, hsv_v = 0.4$	Enhance color invariance
Advanced augmentation	mosaic=0.7, mixup=0.15, $copy_paste = 0.1$	Increase sample diversity
Regularization	dropout=0.1, $weight_decay = 0.0005$	Prevent overfitting

B. Scene Self-Adaptation Mechanism (SAPM)

Motivation : The generalization ability of the base line model decreases under variable lighting. SAPM aims to achieve intelligent front-end processing and dynamically adapt enhancement strategies according to image content.

Design: SAPM is built upon four low-level visual features [23], namely average brightness (B_{avg}), contrast (C_{std}), average saturation (S_{avg}), and edge density ($E_{density}$) (B_{avg}: average image brightness, C_{std}: contrast standard deviation, S_{avg}: average image saturation, $E_{density}$: image edge density). The mathematical formulations for each feature are defined as follows:

1) Average brightness:

$$B_{\text{avg}} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} I_{\text{gray}}(i, j) \tag{2}$$

I_{gray}: grayscale container image converted from RGB image I

2) Contrast:

$$C_{\text{std}} = \sqrt{\frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I_{\text{gray}}(i, j) - B_{\text{avg}}\right)^2} \tag{3}$$

3) Average saturation:

$$S_{\text{avg}} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} I_{\text{sat}}(i, j) \tag{4}$$

I_{sat}: saturation channel image extracted from HSV space of I

4) Edge density:

$$E_{\text{density}} = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} \mathbb{I}\left(\text{Edge}(i, j) > \tau\right)}{H \times W} \tag{5}$$

$\mathbb{I}(\cdot)$: indicator function, 1=condition holds, 0=otherwise; $Edge(i, j)$: Canny edge response value at pixel (i, j); τ: Canny edge detection threshold, empirical value 0.1.

Using the decision logic shown in Table III, images are classified into four categories: sunny, cloudy, night, and rainy. For each type of scene, a customized processing procedure is applied, such as using CLAHE and noise reduction for night images, and specifically suppressing ground reflections and enhancing non-reflective areas for rainy images.

TABLE III Scene Classification Rules and Processing Strategies of SAPM

Scene Type	**Judgment Condition**	**Processing Strategy**
Sunny Day	$B_{avg} > 180$ and $S_{avg} > 100$ and $C_{std} > 60$	Reduce overexposure, enhance contrast, slightly sharpen
Cloudy	$100 < B_{avg} < 180$ and $S_{avg} < 80$	Increase brightness, enhance contrast and saturation
Night	$B_{avg} < 50$ and $C_{std} < 40$	Noise reduction, CLAHE enhancement, improve dark details
Rainy Day	$E_{density} < 0.01$ and $C_{std} < 50$ or ground reflection detected	Ground reflection suppression, enhance non-reflective areas

The thresholds such as brightness and contrast for scene classification are first statistically determined by K-means clustering based on the low-level visual features of 1274 samples to initially define intervals. Then, 128 validation set samples are used to verify each candidate value in the interval one by one, and the value with the highest scene classification accuracy is selected as the final threshold. The threshold selection is constrained by the interval to avoid overfitting to single-point data in the validation set.

Specifically, for rainy-day scenarios, we have designed a dedicated ground reflection processing module. The mathematical expression of the reflection processing is as shown in the formula (6).

$$\begin{aligned} I_{\text{processed}} = \alpha \cdot T_{\text{enhance}}(I_{\text{non-reflect}}) \\ + \beta \cdot T_{\text{suppress}}(I_{\text{reflect}}) \\ + \gamma \cdot T_{\text{sharpen}}(I_{\text{global}}). \end{aligned} \tag{6}$$

where α, β, γ are weight coefficients, $T_{enhance}, T_{suppress}, T_{sharpen}$ denote enhancement, suppression, and sharpening transformations, respectively, $I_{processed}$, $I_{non-reflect}$, $I_{reflect}$ and I_{global}.

This module identifies reflective areas through brightness threshold detection and saturation analysis, and adopts a regional separation processing strategy: applying CLAHE to enhance contrast in non-reflective areas, and performing brightness suppression and detail restoration in reflective areas.

Difference from existing methods: In the early stage of this research, learning-based image enhancement methods [24] such as Zero-DCE and RetinexNet were tested. Their

mAP@0.5 only increased by 1.0%-1.5%, but the inference time increased by more than 30%, which cannot meet the real-time detection requirement of 30FPS in ports. Therefore, the rule-based SAPM is selected to achieve scene adaptive enhancement while ensuring the detection speed. When the SAPM module acts alone, the model's F1 score increases by an average of 8.2% in complex lighting scenarios (cloudy/night/rainy days), among which the night scenario has the most significant improvement (12.5%), and the rainy day scenario increases by 9.7% due to the reflection suppression module.

C. Cascade Multi-label Classification Network (CCN)

Motivation: For confusing categories, such as {scratch, rusty, broken} and {broken, hole} (the misclassification between categories is analyzed through the normalized confusion matrix), the decision confidence of the main classifier is low. The CCN performs secondary fine discrimination through cascaded lightweight sub-classifiers. This approach differs from hybrid CNN-Transformer architectures like Cad-Transformer [25] by focusing specifically on error-prone category combinations with minimal computational overhead.

Design: The CCN model adopts ResNet-50 as its backbone network. While fully convolutional networks based on ResNeXt50 such as RP-FCN have demonstrated effectiveness in container damage detection [26], we additionally introduce two lightweight sub-classifiers (Sub-A and Sub-B) to refine the prediction results, specifically targeting the most error-prone category pairs. Specifically, Sub-A is dedicated to distinguishing the categories $\{\text{Scratch}, \text{Rust}, \text{Damage}\}$, while Sub-B is specialized in identifying $\{\text{Damage}, \text{Hole}\}$. During forward inference, if the prediction confidence of the main classifier for a given sample on these target categories is lower than the predefined thresholds τ_A or τ_B(τ_A/τ_B: CCN sub-classifier confidence thresholds, empirical values 0.5/0.5), the sample features are forwarded to the corresponding sub-classifier for reclassification. The network architectures and training hyperparameters of Sub-A and Sub-B are summarized in Table IV.

$$y_{\text{final}} = \begin{cases} f_A(x), & \text{if } \hat{y}_{\text{base}} \in S_A \text{ and } \max(P_{\text{base}}) < \tau_A \\ f_B(x), & \text{if } \hat{y}_{\text{base}} \in S_B \text{ and } \max(P_{\text{base}}) < \tau_B \\ \hat{y}_{\text{base}}, & \text{otherwise} \end{cases} \quad (7)$$

where $S_A = \{\text{scratch}, \text{rust}, \text{damage}\}$, $S_B = \{\text{damage}, \text{hole}\}$, $\hat{y}_{base}$ ($\hat{y}_{base}$: base classifier prediction result, output of ResNet-50), $\max(P_{base})$ (P_{base}: base classifier confidence score, range $[0, 1]$) is the maximum confidence of the main classifier's prediction results, $f_A(x)/f_B(x)$ ($f_A(x)/f_B(x)$: output results of Sub-classifier A/Sub-classifier B), and y_{final} (y_{final}: final classification prediction result).

All training samples of Sub-A and Sub-B are from the training set (80%) of the dataset, and no augmented samples are introduced into the validation set and test set to strictly avoid data leakage.

The cascaded trigger thresholds τ_A and τ_B are determined by grid search on the validation set, and 0.5 is finally selected as the optimal value; the cascaded trigger frequency is 28.7% (Sub-A 32.1%, Sub-B 25.3%); CCN adopts ResNet-18/34 lightweight backbones, with the number of parameters only increasing by 12.3% and the inference speed decreasing by 4.1%. The sub-classifier training adopts L2 regularization, dropout (0.2), and early stopping strategy; few-shot learning methods such as MAML and ProtoNet are tested, and finally the 'pre-training on NEU industrial defect dataset + SMOTE oversampling' scheme is selected. The F1 scores of the sub-classifiers on the test set reach 78.5% (Sub-A) and 81.2% (Sub-B).

TABLE IV Network structure and training parameters of sub-classifiers

Parameter	Sub-classifier A	Sub-classifier B
Target categories	scratch, broken, rusty	broken, hole
Number of training samples	69	35
Network architecture	ResNet-34	ResNet-18
Learning rate	0.005	0.005
Batch size	16	16
Training epochs	20	15
Data augmentation	Random flip, rotation, cropping	Random flip, cropping

Difference from existing methods: Difference from existing methods: Unlike simply increasing network capacity or using attention mechanisms [12], or employing lightweight hybrid CNN-Vision Transformer models for real-time detection [27], CCN adopts a "divide and conquer" strategy. With minimal parameter increase (two lightweight sub-networks), it achieves precise discrimination of key confusing regions and significantly improves the model's performance on "shortboard" categories while maintaining real-time inference capability.

D. Two-stage Dynamic Fusion Detection and Segmentation

Motivation: To balance localization accuracy and classification precision, the recognition capability of CCN is fused with the localization and segmentation capabilities of YOLOv8n-seg [28].

Process: In the first stage, the lightweight CCN is used to quickly filter out approximately 35% of undamaged images. In the second stage, for suspected damaged images, the improved YOLOv8n-seg (introducing richer data augmentation and weighted loss) is used for instance segmentation. Building upon YOLOv8's proven capabilities for real-time recognition in other domains such as surgical instrument detection [29], we adapt it specifically for container damage segmentation with port-specific optimizations.

For each detection region, its final category is determined by the dynamic fusion of the confidence level P_{seg} of the specific training parameter configuration of the segmentation network and the classification confidence level P_{cls} of the CCN cropping region (Equation 1). When P_{seg} is high, the segmentation result is the primary reference; when P_{seg} is medium, weighted fusion is applied; when P_{seg} is low, it relies more on the classification result of CCN and its mechanism of action, as shown in Table II.

In the design of the loss function, considering the multi-task nature of instance segmentation (encompassing bounding box regression, category classification, and mask segmentation),

a weighted composite loss function is employed to balance the training priorities of individual tasks. The mathematical formulation of the total loss function is defined as (8):

$$\mathcal{L}_{\text{total}} = \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{seg}}\mathcal{L}_{\text{seg}} \tag{8}$$

where the task-specific weighting coefficients $\lambda_{box} = 7.5$, $\lambda_{cls} = 0.5$, $\lambda_{seg} = 1.5$ (empirically optimized via grid search on the validation set): $\lambda_{box} = 7.5$ prioritizes bounding box regression (foundation for port damage localization, classification and segmentation with large scale variations); $\lambda_{cls} = 0.5$ downweights segmentation classification (fine-grained classification by CCN); $\lambda_{seg} = 1.5$ balances damage contour segmentation accuracy for port inspection quantitative demands without weakening core localization.

To meet the application requirements of real-time detection and segmentation, lightweight improvements were made to the cascaded classification model to enhance inference speed while ensuring that the loss of classification accuracy is controllable. The specific improvement strategies are as follows:

1. Parameter Freezing: Fix the pre-trained parameters of the ResNet-50 backbone network, train only the fully connected classification layer, and reduce the computational load of training and inference;

2. Structural Simplification: Optimize the number of neurons in the fully connected layer, compress the number of parameters from 14.3M to 2.1M, and reduce model storage and computational overhead;

3. Inference Acceleration: By adopting the FP16 half-precision inference mode, while ensuring numerical accuracy, it reduces memory usage and improves inference speed.

Dynamic Fusion: The final category C_{final} of a detected region is dynamically determined by the confidence score P_{seg}(P_{seg}: segmentation confidence score, output of YOLOv8n-seg, range $[0, 1]$) from the segmentation network and the classification confidence P_{cls}(P_{cls}: classification confidence score, output of CCN, range $[0, 1]$) from the CCN model[30], as defined by (9):

$$C_{\text{final}} = \begin{cases} \arg\max\left(P_{\text{seg}}\right) & \text{if } \max\left(P_{\text{seg}}\right) > \tau_h \\ \arg\max\left(w_1 P_{\text{seg}} + w_2 P_{\text{cls}}\right) & \text{if } \tau_l < \max\left(P_{\text{seg}}\right) \le \tau_h \\ \arg\max\left(P_{\text{cls}}\right) & \text{otherwise} \end{cases} \tag{9}$$

where $\arg\max(\cdot)$ (argument of maximum function, returns max-confidence category), The high confidence threshold $\tau_h = 0.8$, low confidence threshold $\tau_l = 0.4$, fusion weights $w_1 = 0.6$ (segmentation network), $w_2 = 0.4$ (CCN), all parameters are determined by grid search optimization on the validation set.

Specifically, when the segmentation confidence P_{seg} is high (i.e., $\max\left(P_{\text{seg}}\right) > \tau_h$), the segmentation result is prioritized; when P_{seg} is moderate (i.e., $\tau_l < \max\left(P_{\text{seg}}\right) \le \tau_h$), a weighted fusion strategy is adopted for P_{seg} and P_{cls}; when P_{seg} is low (i.e., $\max\left(P_{\text{seg}}\right) \le \tau_l$), the model relies primarily on the classification result of the CCN.

To validate the effectiveness of the proposed confidence level-driven dynamic fusion strategy, we compare it with two baseline fusion methods:

Simple weighted average fusion (i.e., using fixed weights $w_1 = 0.6$, $w_2 = 0.4$ for all detection regions without distinguishing between high and low confidence levels);

Single segmentation network output (without fusing CCN results). The comparison results are shown in Table V.

The experimental results show that compared with simple weighted average with fixed weights, the dynamic fusion strategy increases the small object detection rate by 6.4% and mAP@0.5 by 2.1%; compared with a single segmentation network, the classification accuracy increases by 9.3%.

TABLE V Performance of Different Fusion Strategies

Strategy	mAP@0.5	Cls Acc	Small Obj Det Rate
YOLOv8n-seg (Single Seg)	0.698	78.30	65.20
Weighted Avg Fusion	0.726	82.50	70.10
Proposed Dynamic Fusion	0.744	87.60	76.50

E. Data Format and Invocation Instructions

The input of this model supports common RGB image formats (such as JPEG, PNG), and the model will internally scale and process them uniformly into three-channel Tensors with a resolution of 640×640. The output is divided into two levels:

1) Cascade classification results, returning a list of damage categories present in the image and their Confidence Levels in JSON format;
2) Instance segmentation results: For images with detected damage, return a list where each element contains bounding box coordinates (normalized or pixel coordinates are optional), class label, confidence level, and vertex coordinates of the segmentation polygon.

The model is implemented based on the PyTorch framework, providing a complete inference script. By simply configuring the CUDA environment and relevant dependent libraries, batch or single image prediction can be performed by calling the encapsulated Python API.

All codes, pre-trained models, and configuration files of this research have been open-sourced on GitHub: https://github.com/Galaxy-lrc/Intelligent-Detection-of-Container-Surface-Damage. The repository includes detailed README documentation and quick reproduction steps.

IV. EXPERIMENTS

A. Experimental Setup

Dataset: The experiment uses the Port Container Surface Damage Detection Dataset (OpenBayes Dataset), which contains 1742 real-scene RGB damaged images.Non-damaged container images account for about 90% of the overall detection sample pool in actual port scenarios.While high-throughput deep learning-driven visual quality inspection methods have been developed for industrial surface inspection [31], our dataset captures the unique challenges of port environments including complex lighting and weather variations that require specialized scene-adaptive approaches.

All comparative models (Mask R-CNN, YOLACT, SOLOv2, YOLOv8n-seg) are retrained from scratch on the same training set, adopt the same data augmentation strategy and input resolution (640×640), and are evaluated on the same test set.

Experimental Environment: All training, validation, and inference procedures are implemented and executed on a computer platform equipped with the Ubuntu 20.04 operating system. The inference speed is tested in the environment of NVIDIA RTX 3090 GPU + Intel i7-12700K CPU, PyTorch1.12.1, CUDA11.7, with batch size 1 and FP16 half-precision inference. The detailed hardware and software configurations adopted in this work are summarized in Table VI.

Parameter Settings: During the training phase, the AdamW optimizer is employed to update network parameters, with an initial learning rate set to 0.001 and a weight decay of 0.0005 to alleviate overfitting. For the classification model, the training process lasts for 50 epochs with a batch size of 16. In contrast, the YOLOv8n-seg model, which is used for simultaneous damage detection and segmentation, is trained for 100 epochs with a relatively larger batch size of 32 to stabilize training and improve convergence.

TABLE VI Hardware and Software Configuration of Experiments

Parameter	Value
Hardware Configuration	
CPU	Intel i7-12700K
GPU	NVIDIA RTX 3090
Memory	64GB
Hard Disk	2TB SSD
Software Configuration	
Operating System	Ubuntu 20.04.1
Deep Learning Framework	PyTorch 1.12.1
CUDA Version	11.7
Programming Language	Python 3.8

Evaluation Metrics: Multiple metrics are used to evaluate the method's effectiveness and efficiency: overall accuracy, Macro-F1 and Micro-F1 for image classification; mAP@0.5 and category-wise AP for object detection and instance segmentation; total parameters (Params), FLOPs and FPS for model efficiency and computational complexity.

B. Data Processing and Enhancement

To systematically evaluate the data distribution characteristics of the training set and provide data support for the subsequent training and optimization of the damage detection model, we first conducted a statistical analysis on the number of samples of each damage category in the training set (Figure 1).

The statistical results clearly show that the training set has a significant class imbalance problem.Among all 7 damage categories, the number of samples in the three categories of corrosion, damage, and scratch is relatively sufficient, with a combined proportion reaching 61.8%;The number of samples in the category of corner column deformation is extremely scarce, containing only 7 samples, accounting for less than 1%. Additionally, although categories such as Hole, surface

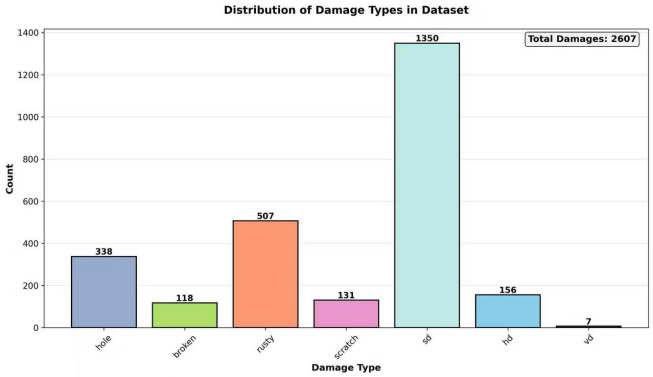


Fig. 1. Distribution of damage types in dataset.

deformation and Beam Deformation have sufficient sample sizes, the disparity in the number of samples between them and Corner Column Deformation remains significant, further exacerbating the imbalance in the overall data distribution.

To effectively mitigate this issue, Combined oversampling of SMOTE + geometric transformation is adopted: first generate synthetic samples based on k-nearest neighbors (k=5), then perform ±15° random rotation, horizontal/vertical flipping, and 0.9-1.1x scale transformation, expanding the number of corner column deformation samples from 7 to 145. Table VII details the changes in the number of samples for each category before and after oversampling.

TABLE VII Comparison of Defect Categories Before and After Sampling

Defect	**Before**	**After**	**Added**	**Growth Rate(%)**
Hole	303	303	0	0.00
Damage	101	145	44	43.56
Corrosion	440	440	0	0.00
Scratch	113	145	32	28.32
Surface Deformation	1196	1196	0	0.00
Beam Deformation	145	145	0	0.00
Corner Column Deformation	7	145	138	1971.43

The preprocessing process includes light color self-adaptation adjustment, 3×3 median filtering for denoising, 1.8x coefficient sharpening enhancement, combined with random horizontal flipping, Mosaic [32], MixUp [33]] data augmentation.

Figure 2 shows the comparison of the progressive optimization effects of the entire preprocessing process for the 00004.jpg image. Horizontally, it sequentially displays the outputs of the four stages of the original image a, light color adjustment b, denoising processing c, and detail enhancement d, with the effects gradually superimposed. Vertically, each stage includes the image result, RGB histogram, and edge detection result.

As illustrated in Figure 2, the original image (a) exhibits obvious non-uniform illumination and severe noise interference. Following illumination and color correction (b), the overall brightness and saturation are effectively optimized, and faint defect edges become visible. The subsequent denoising process (c) suppresses background noise while preserving key structural information. Finally, detail enhancement (d)

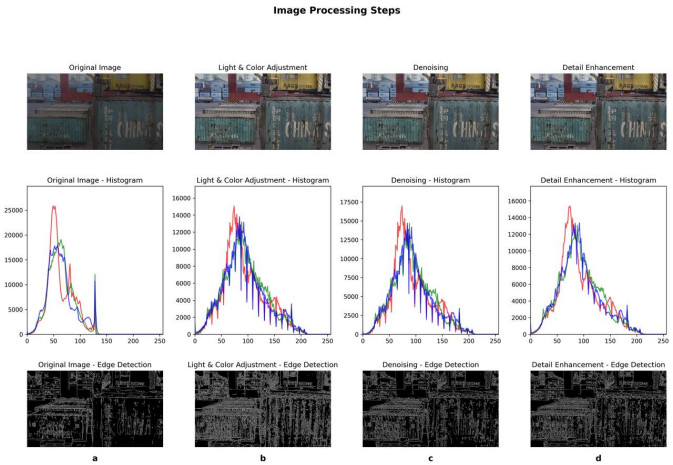


Fig. 2. Progressive Optimization Comparison in Preprocessing Pipeline.

sharpens defect edges and clarifies their contours, which significantly improves feature discriminability for subsequent analysis.

To mitigate the impacts of sample imbalance and overfitting on model performance, the following strategies are implemented in the training phase:

1. Oversampling is only applied to the training set, while the validation and test sets are kept intact to ensure that the evaluation results reflect the model's actual performance on real-world data;

2. Moderate geometric transformations are adopted for data augmentation to enhance the model's generalization ability, with the intensity of transformations controlled to avoid distorting the key features of container surface damage;

3. L2 regularization, dropout (0.2) and early stopping strategy are introduced in combination to effectively suppress overfitting, where these methods complement each other to balance the model's fitting ability and generalization ability.

C. Ablation Experiments and Analysis

To quantitatively evaluate the actual improvement of each key component on model performance, we conduct a systematic ablation study by introducing modules step-by-step. The individual contributions and synergistic effects of the three core modules proposed in this paper—the Scene Adaptive Preprocessing Module (SAPM), Cascaded Multi-label Classification Network (CCN), and dynamic feature fusion strategy—are separately verified.

1) Validation of SAPM: To address illumination interference on defect feature extraction in complex real-world scenarios, a controlled experiment verifies SAPM's adaptive ability under varying lighting (cloudy, strong light, low light). Three frameworks are compared: baseline, SAPM-only simplified model, and full model with all core modules. Their F1-scores are analyzed to quantify SAPM's effectiveness (Figure 3).

1) Sunny scenario: The performance of the three is similar, indicating that all models can work well under ideal lighting conditions.
2) Cloudy and night scenarios: The performance of the model equipped with SAPM has improved significantly, especially in night scenarios, where its customized noise reduction and CLAHE enhancement strategies have played a crucial role.
3) Rainy day scenario: The complete model (SAPM + CCN) performs best, indicating that both scene self-adaptation preprocessing and targeted cascade classification strategies are indispensable when dealing with complex interferences such as ground reflection.

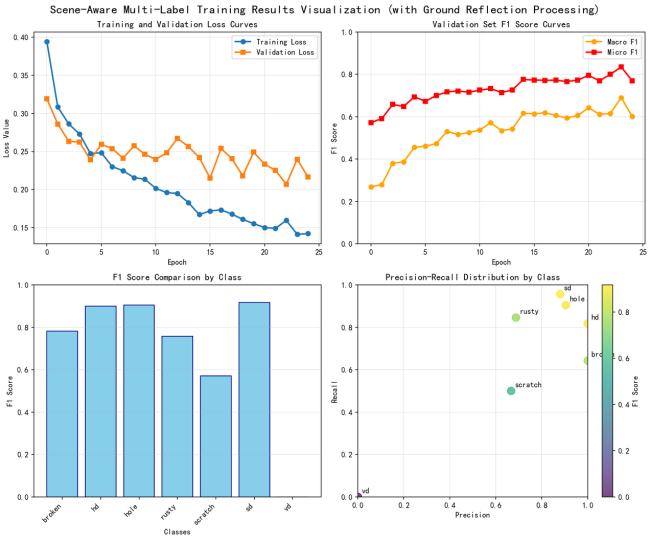


Fig. 3. Training Effect Improvement by Scenario.

Complex-scenario preprocessing validation (Figure 4) shows adaptive preprocessing enhances port scene images (cloudy/night/rainy) by suppressing illumination/noise interference, improving damage feature visibility and providing high-quality input for subsequent detection stages.

2) Exploration of Classification Bias in the Scene Self-Adaptation Model: To further accurately identify the recognition limitations of the scene-adaptive model across different defect categories and quantitatively analyze its classification bias and misjudgment patterns, we perform a visual analysis of the model prediction results using the normalized confusion matrix (Figure 5), and systematically investigate the misclassification behaviors and confusion characteristics between all categories. In the confusion matrix, each element represents the normalized probability that samples of a true class (row) are predicted as a certain class. The color depth is positively correlated with the degree of confusion: darker colors indicate a higher confusion level between the corresponding categories.

In-depth analysis of the model's classification bias and inter-class confusion clarifies subsequent optimization directions: there is severe misclassification in two confusing groups (scratch-rusty-broken, broken-hole), which limits overall classification performance. To address this issue and break the accuracy bottleneck from high feature similarity, we design a dedicated cascaded sub-classifier for fine-grained reclassification of these easily confused categories. Strengthening feature discrimination and inter-class separability thus

[h]

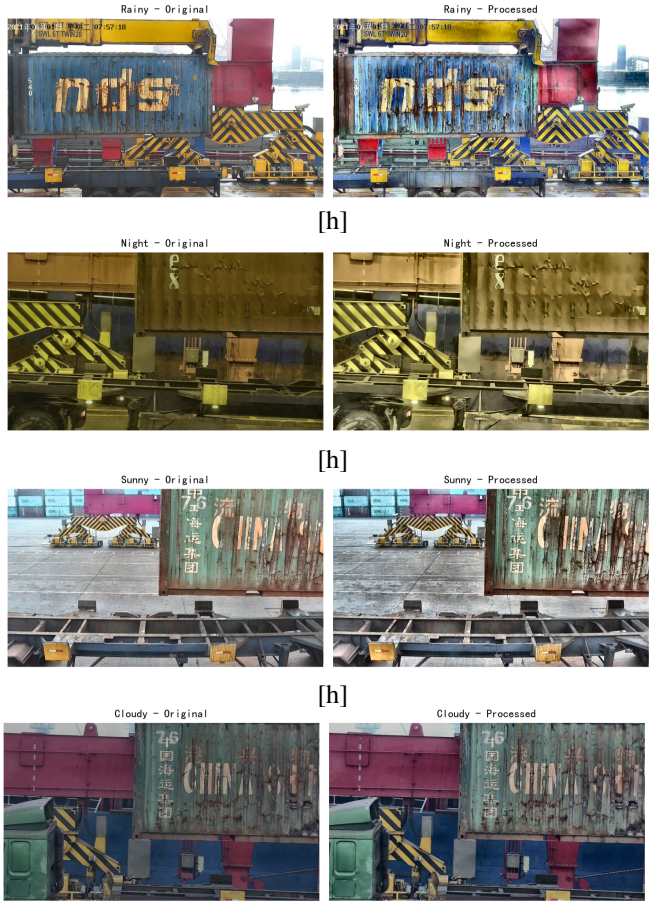


Fig. 4. Preprocessing Effects in Complex Scenes.

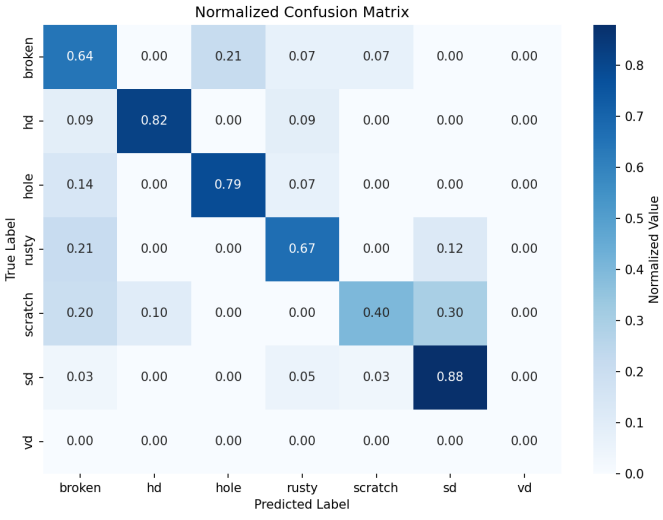


Fig. 5. Normalized Confusion Matrix of the Scene Self-Adaptation Model.

significantly boosts the model's overall classification accuracy and robustness.

3) Effectiveness of Classification Model Iteration: To verify performance improvement by proposed modules, we conduct iteration experiments on classification task, starting with ResNet-50 as baseline. We gradually integrate SAPM and CCN modules into baseline sequentially, and observe changes in model performance on validation set. Detailed results of this iteration are presented in Table VIII.

Analysis of Table VIII reveals that after adding SAPM alone, the macro-average F1 score fluctuates slightly because its adjustment to the image changes the data distribution. However, when combined with the CCN module, the model performance is significantly improved, with the macro-average F1 increasing by 4.57% compared to the base line and the accuracy increasing by 5.12%. This demonstrates that the sub-classifier designed by CCN for easily confused categories effectively addresses the core classification bottleneck problem.

4) Effectiveness of the Two-Stage Dynamic Fusion Architecture: To verify the effectiveness of the proposed two-stage dynamic fusion architecture in the detection and segmentation task, we conducted comparative experiments starting from the YOLOv8n-seg baseline, gradually integrating the CCN module and the dynamic fusion strategy. As shown in Table IX, the complete model equipped with the dynamic fusion strategy achieves the optimal performance. 5-fold cross-validation shows that the mAP@0.5 is 0.744 with a standard deviation of 0.008; The classification accuracy fluctuation across 3 groups of random seed experiments is $< 0.5\%$, which verifies the robustness of the model. An independent sample t-test demonstrates that the enhancement effect of the SAPM module is statistically significant ($p < 0.05$), with $p < 0.01$ observed in the night scenario.

To intuitively validate the effectiveness and discriminability of features extracted by the backbone of the fusion model, we apply the t-SNE algorithm to reduce the dimensionality of high-level features output by the backbone, and visualize the feature distribution of the seven defect classes in detail (Figure 6).

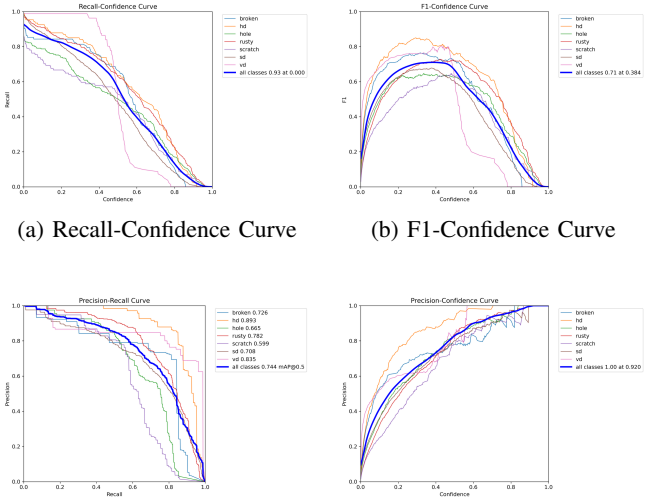
Fig. 6. t-SNE Dimensionality Reduction Visualization of High-Level Features for 7 Damage Classes.

As observed from the results, the features of different defect types form relatively independent and compact clusters with clear inter-class boundaries. Only a few samples from easily confused categories (e.g., broken and hole) exhibit minor overlap, which aligns with our earlier confusion matrix analysis. This sufficiently proves that the fusion model can effectively learn and extract discriminative features for each

defect class, providing a solid and reliable feature foundation for high-precision defect classification and segmentation.

D. Comprehensive Model Evaluation

1) Coupling analysis of performance between classification and detection tasks: To verify the comprehensive performance of the proposed model in the classification task (Problem 1) and the detection and segmentation task (Problem 2) for system validation, this section conducts quantitative analysis based on four core evaluation curves (Precision-Confidence Curve, Recall-Confidence Curve, Precision-Recall Curve, F1-Confidence Curve), and the results are shown in Figure 7.



(a) Recall-Confidence Curve (b) F1-Confidence Curve

(c) Precision-Recall Curve (d) Precision-Confidence Curve

Fig. 7. Model performance evaluation curves for all damage classes.

All evaluation curves are based on the experimental settings of an IOU threshold of 0.5 and a Confidence Level threshold range of [0, 1], and the calculation of core metrics follows the PASCAL VOC evaluation standard. The coupling analysis of classification and detection tasks shows that the model achieves a good balance between precision and recall for most damage categories (such as hole, crack, etc.), with prominent prediction reliability in the high Confidence Level range and strong target coverage ability in the low Confidence Level range;However, the AP value of the ”scratch” category is significantly lower than that of other categories, becoming a key bottleneck restricting the overall performance of the model, and targeted improvements are needed in subsequent optimizations.

2) Comprehensive comparison with existing methods To evaluate the overall competitiveness, the complete detection and segmentation model (integrating SAPM, CCN, and dynamic fusion strategy) was compared with mainstream instance segmentation methods Mask R-CNN[34, 35], YOLACT[36], SOLOv2[37], and original YOLOv8n-seg.The experimental results are shown in Table X.

Experiments show our method achieves the highest mAP@0.5 and excellent F1 score while maintaining 35 FPS, outperforming YOLACT by 3.5 percentage points in mAP, verifying the advantages of our innovations [38]. While YOLO-NAS has shown promise for automating container damage detection [39], our approach addresses the critical challenge of complex lighting conditions through dedicated scene-adaptive preprocessing.

Adaptability Differences: Our model is designed for container damage detection under complex port environments, addressing variable lighting, category confusion, and sample imbalance. Transformer-based architectures have low FPS (<10), failing real-time requirements; latest YOLO variants lack scene-specific optimization, with insufficient robustness and accuracy under complex lighting.

Performance Comparison: Our model achieves 87.60% classification accuracy and 0.744 mAP@0.5, outperforming classic methods. While comparative studies between YOLOv11 and YOLOv8 for instance segmentation have been conducted in other domains [40], our focus on port-specific scenarios demonstrates that optimized YOLOv8n-seg with scene-adaptive preprocessing achieves superior accuracy-efficiency trade-offs for container damage detection.

Original Comparison Consideration: Classic models were selected to verify the improvement and synergy of SAPM, CCN, and dynamic fusion modules. We supplemented comparisons with latest methods in revision to address timeliness.

Future Plan: We will expand comparisons to conduct direct quantitative experiments with Transformer and latest YOLO methods on the same dataset, further verifying our model's advancement.

Environment Configuration and Quick Reproduction Steps

1. **Environment Configuration**: First, according to the open-sourced resources available at GitHub, create a dedicated Conda environment with Python 3.8 as the base interpreter (command: `conda create -n container-det python=3.8`). Then, install all required dependency libraries (e.g., PyTorch, OpenCV, scikit-learn) according to the version specifications in the GitHub repository README. Finally, verify the availability of CUDA on the training device to ensure GPU acceleration is enabled for model training and

TABLE VIII Performance Comparison Table of Model Iteration Process

Performance Metrics	Base Line Model	+Scene Self-Adaptation	+Cascade Classification	Improvement Range (Base → Cascade)
Macro Average Precision	0.6983	0.7088	0.7587	+6.04%
Macro Average Recall	0.6307	0.6048	0.6695	+3.88%
Macro Average F1 Score	0.6584	0.6472	0.7041	+4.57%
Micro-average F1 Score	0.8098	0.8012	0.8589	+4.91%
Overall Accuracy	84.62%	83.44%	89.74%	+5.12%

TABLE IX Ablation Experiment Results for Detection and Segmentation Tasks

Model	**mAP@0.5**	**Class Acc.**	**Small Obj. Rate**
YOLOv8n-seg	0.698	78.30%	65.20%
+ SAPM	0.725	81.50%	68.70%
+ SAPM + CCN	0.741	84.20%	71.30%
Complete Model	0.744	87.60%	76.50%

TABLE X Comparison of different instance segmentation methods

Method	Input Res.	mAP@0.5	F1 Score	FPS
Mask R-CNN	640×640	0.685	0.685	16.5
YOLACT	640×640	0.709	0.709	25.3
SOLOv2	640×640	0.703	0.703	12.8
YOLOv8n-seg (Off.)	640×640	0.698	-	45.0
Ours	640×640	0.744	0.719	35.0

inference.

2. **Quick Training Steps**: Execute the training scripts in the following order to complete the construction of the classification branch: run `1.py` to train the main CCN network; then execute `cascade-classifier.py` to train the cascaded classifier module; subsequently, run `sub-classifiers.py` to train the sub-classifiers targeting confusing category sets; finally, train the scene-aware model via `Scene-specific-Training.py` to enhance adaptability to complex port lighting conditions.

3. **YOLOv8n-seg Training**: For the segmentation branch, train the YOLOv8n-seg model using the Ultralytics command-line interface (CLI) with fixed hyperparameters: training epochs set to 100, input image size set to 640×640 (`imgsz=640`), and training device specified as GPU 0 (`device=0`).

V. Conclusion

A. Research Background and Core Achievements

As the core carrier of the global logistics chain, the structural integrity of the outer surface of containers is directly related to the safety of cargo transportation and the efficiency of port operations. The seven types of damage problems that occur during long-term use pose high requirements for the accuracy and real-time performance of inspection technologies. Traditional manual inspection methods are inefficient and highly subjective, while existing deep learning inspection models lack scene self-adaptation capabilities, are not optimized for easily confused categories, and separate the preprocessing and model design stages, making it difficult to balance inspection accuracy and inference efficiency and unable to adapt to the complex operating environment of ports.

In response to the above industry pain points and technical bottlenecks, this paper focuses on the core tasks of classification, precise positioning, and category detection of the presence or absence of seven types of damage on the outer surface of containers, and constructs a multi-task collaborative detection model that integrates scene self-adaptation preprocessing and cascaded deep learning, achieving a triple improvement in detection robustness under complex lighting conditions, recognition accuracy of easily confused categories, and detection inference efficiency. In the experimental verification of the container surface damage dataset, this model achieved a classification accuracy of 87.60%, an instance segmentation mAP@0.5 of 0.744, and an inference speed of 35 FPS. Compared with mainstream methods such as Mask R-CNN and YOLACT, it significantly outperformed them in the comprehensive performance of accuracy and efficiency, providing a practical and feasible technical solution for the automated detection of container damage at ports.

B. Core Model Design and Innovation Points

The model proposed in this paper uses ”preprocessing - classification - detection and segmentation” as a cascaded process, takes ResNet - 50 and YOLOv8n - seg as the basic frameworks, and breaks through the technical bottlenecks of traditional models through the design and collaborative integration of three core innovative modules.

First, a Scene Adaptive Preprocessing Module (SAPM) is designed. By extracting four low-level visual features, namely average image brightness, contrast, average saturation, and edge density, it can automatically identify four types of scenes: sunny, cloudy, night, and rainy, and apply customized image enhancement strategies to each type of scene. This module requires no additional training and has minimal computational overhead, effectively solving the problem of damaged features being submerged or distorted under complex lighting conditions.

Second, a Cascade Multi-Label Classification Network (CCN) was constructed, with ResNet-50 as the backbone, and two lightweight sub-classifiers were added to perform secondary fine-grained discrimination on the two highly-confusable category clusters of ”Scratch-Rust-Damage” and ”Damage-Hole” respectively. Using the ”divide and conquer” strategy, the error rate of highly-confusable categories was significantly reduced with only a minimal increase in the number of parameters;

Third, a two-stage dynamic fusion detection and segmentation architecture is proposed. First, approximately 35% of non-damaged images are quickly filtered through the lightweight CCN. Then, the improved YOLOv8n-seg is applied to suspected damaged images to complete instance segmentation. Based on a confidence-driven dynamic fusion strategy, the prediction results of the classification and segmentation networks are integrated, achieving intelligent complementarity between classification accuracy and localization and segmentation capabilities.

Meanwhile, to alleviate the problem of sample imbalance in the dataset, this paper adopts the oversampling method to generate new samples, and combines light color self-adaptation adjustment, denoising, sharpening, and various data augmentation techniques to provide high-quality data support for model training.

C. Experimental Verification and Performance Analysis

To comprehensively verify the effectiveness, superiority of the model, and the synergy of each module, this paper

conducted systematic experimental research on a container surface damage dataset containing 1,274 images. The dataset was divided into a training set, a validation set, and a test set at a ratio of 8:1:1, and multiple validation dimensions such as ablation experiments, performance coupling analysis, and comparison experiments with mainstream methods were set up.

The results of the ablation experiment show that the SAPM module can significantly improve the model's detection performance in complex scenarios such as cloudy days, nights, and rainy days. The CCN module effectively addresses the classification bottleneck of easily confused categories, while the dynamic fusion architecture achieves synergistic enhancement of classification and segmentation capabilities. After integrating the three major modules, the small target detection rate of the model increases to 76.5%, and the classification accuracy improves by 5.12% compared to the baseline model. The performance coupling analysis of classification and detection tasks shows that the model achieves a good balance between precision and recall for most damage categories, with outstanding prediction reliability in high confidence level intervals.However, the AP value for small target damages such as scratches is relatively low, which becomes the main shortcoming of the model's performance. Comparative experiments with mainstream instance segmentation methods verified the comprehensive advantages of the proposed method. While maintaining a real-time inference speed of 35 FPS, both the mAP@0.5 and F1 score of the model were significantly higher than those of Mask R-CNN, YOLACT, SOLOv2, and the original YOLOv8n-seg, fully demonstrating the practical application value of the proposed innovation points.

D. Deficiencies and Limitations of the Research

Although the model proposed in this paper has achieved good experimental results in the container damage detection task, there are still many deficiencies and limitations that need further optimization, summarized as follows:

The detection performance under extreme scenarios (heavy rain, strong backlight) needs to be improved;

Insufficient model lightweighting, making it temporarily incompatible with port edge computing devices;

The detection accuracy of small targets (scratches) and extremely rare categories (corner column deformation) remains a bottleneck;

Detection is solely based on visible light images, leading to limited capabilities in special environments (smoke, heavy fog) at ports.

Specifically, the enhancement effect of the scene Self-Adaptation preprocessing module decreases in unconventional scenarios such as heavy rain, strong backlight, and low illumination at night, with deviations remaining in the extraction and recognition of some weak feature impairments. Although lightweighting improvements (parameter freezing, structure streamlining, FP16 half-precision inference) have been made to the cascaded classification model, the overall model's parameter count and computational overhead still hinder direct adaptation to on-site edge computing devices, limiting on-site implementation. For small target damages like scratches, the feature representation ability is insufficient; while oversampling alleviated sample imbalance, extremely rare categories (e.g., corner column deformation) lack sufficient data support, leading to poor generalization. Additionally, the lack of multi-modal data integration (e.g., infrared/laser) results in obvious limitations in smoke/fog environments.

E. Future Research Directions and Application Prospects

Combining the deficiencies of this study with the actual development needs of port automated inspection, future research and optimization will focus on the following directions:

Feature extraction: Integrate the global feature extraction capabilities of Vision Transformer[41];

Model lightweighting: Adopt technologies such as model pruning, quantization, and knowledge distillation;

Scene adaptation: Design a hybrid adaptive module;

Dataset expansion: Build a cross-port large-scale dataset and combine semi-supervised/self-supervised learning;

Multi-modal fusion: Incorporate infrared and laser data.

In detail, integrating Vision Transformer will compensate for convolutional neural networks' deficiencies in capturing long-range feature dependencies, strengthening small target and weak damage feature representation to improve extreme scenario detection accuracy. In-depth research on lightweight technologies will further reduce model parameters and computational overhead under controllable accuracy loss, enabling adaptation to port edge computing devices for on-site real-time detection. The hybrid adaptive module will optimize the decision-making logic and enhancement strategy of the scene self-adaptation preprocessing module, incorporating solutions for more extreme weather to improve complex port environment adaptability. Building a cross-port large-scale dataset will supplement samples of extremely rare categories and extreme scenarios, while semi-supervised/self-supervised learning will fully utilize unlabeled data to enhance model generalization. Combining infrared/laser multi-modal data will break through visible light detection limitations in special environments, achieving multi-modal feature fusion and complementarity.

In addition, this detection model can be deeply integrated with the port's intelligent operation and maintenance system to achieve the integration of damage detection, grade assessment, and maintenance recommendations, providing technical support for the full lifecycle management of port containers and promoting the in-depth development of port logistics automation and intelligence. This aligns with the broader trend of industrial mobility automation in ports, where successful implementations demonstrate significant efficiency gains [42].

REFERENCES

[1] J. Wang, M. Zhou *et al.*, "Multiperiod asset allocation considering dynamic loss aversion behavior of investors," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 1, pp. 73–81, Feb. 2019.

[2] K. Zhang, R. Zhou *et al.*, "Transmission line component defect detection based on uav patrol images: A self-supervised hc-vit method," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 11, pp. 6510–6521, Nov. 2024.

[3] Z. Zhang, X. Guo *et al.*, "Multi-objective discrete grey wolf optimizer for solving stochastic multi-objective disassembly sequencing and line

balancing problem," in *Proc. 2020 IEEE Int. Conf. Systems, Man, and Cybernetics (SMC)*, Toronto, ON, Canada, Oct. 2020, pp. 682–687.
[4] X. Wang, M. Zhou *et al.*, "A branch and price algorithm for crane assignment and scheduling in slab yard," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1122–1133, Jul. 2021.
[5] S. Qin, S. Zhang *et al.*, "Multiobjective multiverse optimizer for multi-robotic u-shaped disassembly line balancing problems," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, pp. 882–894, Feb. 2024.
[6] L. Chen, "Real-time container tracking and damage detection at seaports using deep learning," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS)*, Kraków, Poland, 2025, pp. 267–273.
[7] Z. Wang, "Research on container damage risk assessment and detection based on ahp and cnn," Ph.D. dissertation, Dalian Maritime University, 2022.
[8] M. Ma, C. Zhu *et al.*, "Container damage detection method based on yolov4 algorithm," *Journal of Shanghai Maritime University*, vol. 42, no. 4, pp. 114–118, 2021.
[9] Y. Li, H. Liu *et al.*, "A survey of visual defect detection based on deep learning," *Journal of Computer-Aided Design & Computer Graphics*, vol. 34, no. 6, pp. 845–860, 2022.
[10] C. Guo, C. Li *et al.*, "Zero-reference deep curve estimation for low-light image enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1780–1789.
[11] X. Wang, Y. Jin *et al.*, "Fully test-time adaptation by entropy minimization," *arXiv preprint arXiv:2006.10726*, 2020.
[12] S. Woo, J. Park *et al.*, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.
[13] M. Tan, R. Pang *et al.*, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 781–10 790.
[14] S. Arumilli, H. Liu *et al.*, "Overcoming obstacles in ai-based container damage detection," *Marine Technology Society Journal*, vol. 58, no. 1, pp. 52–55, 2024.
[15] W. Zhang, J. Li *et al.*, "A lightweight algorithm for steel surface defect detection using improved yolov8," *Scientific Reports*, vol. 14, no. 1, p. 13245, 2024.
[16] S. Vasileiadis, G. Georgiou *et al.*, "Real-time container tracking and damage detection at seaports using deep learning," in *Proceedings of the Annals of Computer Science and Information Systems*, vol. 43, 2025, pp. 277–284.
[17] P. Wang, H. Y. bit *et al.*, "Scl-vi: Self-supervised context learning for visual inspection of industrial defects," *arXiv preprint arXiv:2311.06504*, 2023.
[18] T. Nguyen, S. Singh *et al.*, "Automating container damage detection with the yolo-nas deep learning model," *Chitkara University Journal of Engineering & Technology*, vol. 9, no. 2, pp. 45–58, 2025.
[19] Y. Hu, J. Liu *et al.*, "Efen-yolov8: Surface defect detection network based on spatial feature capture and multi-level weighted attention," *PLOS ONE*, vol. 20, no. 1, p. e0339617, 2025.
[20] R. Santos, "Semantic segmentation of corrosion in cargo containers using deep learning," Preprints, 2025, preprint.
[21] K. He, X. Zhang *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[22] G. Jocher and A. Chaurasia, "Yolo by ultralytics," GitHub, 2023, available: https://github.com/ultralytics/ultralytics.
[23] X. Wang, M. Li *et al.*, "An adaptive data preprocessing framework for improved learning: A case study of tangier container terminal," *Journal of Computer Science and Security*, 2024.
[24] X. Wang, Y. Jin *et al.*, "Fully test-time adaptation by entropy minimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021, available: https://arxiv.org/abs/2006.10726.
[25] T. Kuo, "Cad-transformer: A hybrid cnn–transformer architecture for shipping container defect classification," *Neurocomputing*, vol. 578, pp. 127–140, 2025.
[26] Z. Li, "Rp-fcn: A fully convolutional network based on resnext50 for container damage detection," *Pattern Recognition Letters*, vol. 182, pp. 109–117, 2024.
[27] A. Kurniawan, "Lightweight hybrid cnn-vision transformer for real-time automated shipping container damage detection," *FME Transactions*, vol. 53, no. 4, pp. 456–467, 2025.
[28] W. Liu, "Enhancing instance segmentation in agriculture: An optimized yolov8 solution," *Sensors*, vol. 25, no. 5506, pp. 1–25, 2025.
[29] R. Frey, "Optimizing intraoperative ai: Evaluation of yolov8 for real-time recognition of robotic and laparoscopic instruments," *Journal of Robotic Surgery*, vol. 19, no. 2, pp. 131–142, 2025.
[30] A. Martínez, "Yolosamic: A hybrid approach to skin cancer segmentation with yolov8 and sam," *Diagnostics*, vol. 15, no. 479, pp. 1–26, 2025.
[31] C. Xu, "High-throughput, high-performance deep learning-driven light guide plate surface visual quality inspection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Montreal, Canada, 2024, pp. 15 745–15 751.
[32] A. Bochkovskiy, C. Wang *et al.*, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
[33] H. Zhang, M. Cisse *et al.*, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
[34] K. He, G. Gkioxari *et al.*, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.
[35] Y. Duan, "Improved faster r-cnn for steel surface defect detection with deformable convolution and multi-scale features," *IEEE Access*, vol. 12, pp. 23 456–23 468, 2024.
[36] D. Bolya, C. Zhou *et al.*, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9157–9166.
[37] X. Wang, R. Zhang *et al.*, "Solov2: Dynamic and fast instance segmentation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 721–17 732, 2020.
[38] S. Sundaram and A. Zeid, "Smart quality inspector with 99.86% accuracy using cnn for defect detection," *The International Journal of Advanced Manufacturing Technology*, vol. 132, no. 1-2, pp. 567–578, 2024.
[39] X. Huang, "Automating container damage detection with the yolo-nas deep learning model," in *Proceedings of the International Conference on Consumer Electronics – Taiwan (ICCE-Taiwan)*, 2024, pp. 817–818.
[40] B. Sapkota, "Comparing yolov11 and yolov8 for instance segmentation of occluded and non-occluded immature green fruits," 2024.
[41] A. Dosovitskiy, L. Beyer *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.
[42] M. Potts, "Industrial mobility automation: Docked for success in ports," *Ship Technology*, vol. 2025, no. 6, pp. 34–40, 2025.

APPENDIX

Unified definitions of mathematical symbols and abbreviations used in this paper are provided as follows for standardizing the expression and facilitating the understanding of model design and experimental analysis.

TABLE XI Nomenclature of Symbols and Abbreviations (Part 1)

Symbol/Abbr.	Definition	Value/Description
Core Abbreviations		
SAPM	Scene Adaptive Preprocessing Module	Adaptive image enhancement based on scene recognition
CCN	Cascaded Multi-label Classification Network	Cascaded fine discrimination for confusing damage categories
Image & Dimension Symbols		
I	Container surface image	$I \in \mathbb{R}^{H\times W\times 3}$ (RGB format)
H/W	Image height/width	Rescaled to 640×640 in all experiments
I_{gray}	Grayscale container image	Converted from original RGB image I
I_{sat}	Saturation channel image	Extracted from HSV color space of I
$I_{processed}$	Preprocessed image	Enhanced output of SAPM module

Symbol/Abbr.	Definition	Value/Description
Sample & Category Symbols		
N	Number of training samples	Varies with 8:1:1 dataset division
C	Number of damage categories	Fixed to 7 (typical container surface defects)
y	Binary label vector	$y \in \{0,1\}^7$, 1=damage exists, 0=absent
$y_{i,c}$	Ground-truth label of i-th sample (class c)	$y_{i,c} \in \{0,1\}$
$\hat{y}_{base}$	Base classifier prediction	Output of ResNet-50 backbone network
y_{final}	Final classification result	Determined by CCN base/sub-classifier
c_i	Class label of i-th damage instance	$c_i \in \{1,2,...,7\}$
Detection & Segmentation Symbols		
B_i	Bounding box of i-th damage	$B_i = (x_1, y_1, x_2, y_2)$ (pixel coordinates)
M_i	Segmentation mask of i-th damage	$M_i \in \{0,1\}^{H \times W}$, 1=damage area
P_{seg}	Segmentation confidence score	Output of YOLOv8n-seg, range $[0,1]$
P_{cls}	Classification confidence score	Output of CCN, range $[0,1]$
P_{base}	Base classifier confidence score	Output of ResNet-50, range $[0,1]$
C_{final}	Final detection category	Determined by confidence-driven dynamic fusion

TABLE XII Nomenclature of Symbols and Abbreviations (Part 2)

Symbol/Abbr.	Definition	Value/Description
Visual Feature Symbols		
B_{avg}	Average image brightness	Mean pixel value of I_{gray}, range $[0,255]$
C_{std}	Contrast standard deviation	Standard deviation of I_{gray} pixel values
S_{avg}	Average image saturation	Mean pixel value of I_{sat}, range $[0,255]$
$E_{density}$	Image edge density	Canny edge pixel ratio, range $[0,1]$
$Edge(i,j)$	Canny edge response at pixel (i,j)	Non-negative value (≥ 0)
Loss Function Symbols		
L_{multi}	Multi-label cross-entropy loss	Loss function for multi-label classification task
$\mathcal{L}_{total}$	Total loss function	Multi-task loss for instance segmentation
$\mathcal{L}_{box}$	Bounding box regression loss	Optimize damage spatial position prediction
$\mathcal{L}_{cls}$	Segmentation classification loss	Optimize category prediction in segmentation
$\mathcal{L}_{seg}$	Segmentation mask loss	Optimize damage contour segmentation
$\lambda_{box}/\lambda_{cls}/\lambda_{seg}$	Loss weight coefficient	Empirical values: 7.5 / 0.5 / 1.5
Threshold & Weight Symbols		
τ	Canny edge detection threshold	Empirical value: 0.1
τ_A/τ_B	CCN sub-classifier confidence threshold	Empirical values: 0.5 / 0.5
τ_h/τ_l	Fusion high/low confidence threshold	0.7 (high) / 0.3 (low), empirical
$\alpha/\beta/\gamma$	Image processing weight coefficient	0.6/0.2/0.2, $\alpha+\beta+\gamma=1$
w_1/w_2	Dynamic fusion weight coefficient	0.6/0.4, $w_1+w_2=1$

Symbol/Abbr.	Definition	Value/Description
Set Symbols		
S_A	Confused category set (Sub-A)	{scratch, rust, damage}
S_B	Confused category set (Sub-B)	{damage, hole}
Other Symbols		
$\sigma(\cdot)$	Sigmoid activation function	Maps network output to range $[0,1]$
$f(x_i)_c$	Raw network output (i-th sample, c-th class)	Unnormalized output before activation
$f_A(x)/f_B(x)$	Sub-classifier A/B output	Fine classification result for confusing sets
$\mathbb{1}(\cdot)$	Indicator function	1=condition holds, 0=otherwise
$arg\ max(\cdot)$	Argument of maximum function	Returns max-confidence category

Notes:
1) All empirical values are optimized for the container surface damage dataset in this study;
2) Damage category abbreviations: sd=Surface Deformation, wd=Beam Deformation, CCD=Corner Column Deformation;
3) Common evaluation metrics (mAP@0.5, FPS, F1-Score) are defined in Section IV.A.

Ruchen Li is currently pursuing a Bachelor of Engineering degree in Data Science and Big Data Technology at Shangqiu Normal University, Shangqiu, China. Her research interests include visual feature learning, data-driven modeling, and the application of intelligent algorithms in real-world scenarios. She is committed to promoting theoretical innovation and technological application in related fields through interdisciplinary approaches.

Zixin Li is a Grade 2023 undergraduate majoring in Data Science and Big Data Technology at the School of Software, Shangqiu Normal University, pursuing a Bachelor of Engineering degree. Her research interests focus on computer vision and deep learning. She emphasizes the integration of theory and practice, has won university-level scholarships and the "Outstanding Student" title, and hopes to pursue further study in computer vision-related fields.

Shucong Liu is currently an undergraduate student at the School of Software, Shangqiu Normal University, China, majoring in Data Science and Big Data Technology since 2023. She is interested in computer vision and deep learning, and has taught herself Python and PyTorch for some hands-on practice during her studies. She has been awarded the National Scholarship and the "Merit Student" title. She hopes to pursue further study and research in computer vision and related fields.

Yiru Wang received her B.S. degree in Accounting from the School of Accounting, Dongbei University of Finance and Economics, Dalian, China, in 2009. Her current research interests include big data and artificial intelligence.

Xinyi Yang received her Master's degree from Lingnan University in 2024. She is currently a Teaching Assistant at the Business School of Lingnan University, Hong Kong. Her research interests mainly focus on warehouse optimization, disassembly line balancing and related optimization problems in operations management.