PeerJ Computer Science

Container defect detection using enhanced YOLO with uncertainty-aware dynamic modulated convolution and parallelized patch-aware attention

Benshuo Zhang[1], Dapeng Cheng[1], Zhipeng Liang[2], Xinhao Li[3],
Feng Zhao[1,4,5] and Zhiyong An[1]

[1] School of Computer Science and Technology, Shandong Technology and Business University, Yantai, China
[2] Yantai Huadong Electronic Technology Co., Ltd, Yantai, China
[3] Department of Computer Science, Beijing Normal University–Hong Kong Baptist University United International College, Zhuhai, Guangdong, China
[4] College of Information Engineering, Xinjiang Institute of Engineering, Urumqi, Xinjiang, China
[5] Key Laboratory of Xinjiang Coal Resources Green Mining, Ministry of Education, Urumqi, Xinjiang, China

Submitted 11 July 2025
Accepted 2 January 2026
Published 13 February 2026

Corresponding author
Zhiyong An,
anzhiyong@sdtbu.edu.cn

Academic editor
Ankit Vishnoi

Additional Information and Declarations can be found on page 23

DOI 10.7717/peerj-cs.3627

ABSTRACT

Container defect detection is a critical task in industrial settings to ensure logistics safety and cargo integrity. However, conventional detection methods often fall short due to the diverse morphologies, varying scales of defects, and complex background interference. To address these challenges, we propose an enhanced single-stage object detection model, termed UPAD-YOLO. Specifically, we introduce a novel Uncertainty-Aware Dynamic Modulated Convolution (UDMConv), which dynamically adjusts the receptive field of convolutional kernels to efficiently extract multi-scale features, thereby accommodating the morphological diversity of defects while enhancing computational efficiency. We further integrate a Parallelized Patch-Aware Attention (PPA) module that leverages multi-branch architecture, patch-aware attention mechanisms, and both channel and spatial attention to substantially improve detection accuracy and robustness. In addition, the detection head of the original model is combined with an Adaptively Spatial Feature Fusion (ASFF) module to effectively integrate defect features across different resolution levels, thereby enhancing the model's capability to recognize multi-scale defect patterns. Experimental results demonstrate that UPAD-YOLO outperforms the baseline model YOLOv11 in container defect detection, achieving a 4% improvement in mean Average Precision (mAP) and a 2.9% increase in F1-score, highlighting its robustness and practical applicability. This study not only offers an efficient and reliable solution for container defect detection but also provides valuable insights for broader applications in industrial visual inspection tasks.

Subjects Artificial Intelligence, Computer Education, Computer Vision, Data Mining and Machine Learning, Neural Networks
Keywords Container defect detection, Multi-scale feature extraction, UPAD-YOLO, Uncertainty-aware dynamic modulated convolution, Parallelized patch-aware attention, Adaptively spatial feature fusion

How to cite this article Zhang B, Cheng D, Liang Z, Li X, Zhao F, An Z. 2026. Container defect detection using enhanced YOLO with uncertainty-aware dynamic modulated convolution and parallelized patch-aware attention. **PeerJ Comput. Sci. 12:e3627 DOI 10.7717/peerj-cs.3627**

INTRODUCTION

Containers play an increasingly vital role in the global logistics and transportation system, rendering them indispensable to modern international trade (*Bahrami et al., 2022*; *Bandong, Nazaruddin & Joelianto, 2021*; *Haralambides, 2019*; *Fouseki, 2023*; *Odiegwu, 2022*; *Skender, Host & Nuhanović, 2016*; *Rodrigue & Notteboom, 2015*). However, inevitable wear and damage accumulate during long-term use of containers, which not only compromises cargo integrity and safety, but may also result in transportation delays, increased maintenance costs, and even reduced operational efficiency across the entire supply chain (*Wang et al., 2021*; *Grzelakowski, 2019*). The causes of container defects are multifactorial, encompassing mechanical damage, environmental exposure, material aging, external impacts, and cargo-induced stress. As containers age over prolonged service periods, degradation-related issues become increasingly severe, significantly compromising their structural safety, functional applicability, and overall durability.

Early manual visual inspections heavily relied on the experience and judgment of inspectors, making them susceptible to errors and omissions due to fatigue, oversight, or subjective bias. Moreover, this inspection approach is not only time-consuming and labor-intensive, but also demands a high level of expertise from inspectors, ultimately undermining overall efficiency and reliability (*Nguyen Thi Phuong, Cho & Chatterjee, 2025*; *Zhou, Chen & Tang, 2024*; *Chen et al., 2019*).

To improve detection efficiency and accuracy, automated inspection methods based on image processing and machine learning have been increasingly investigated and applied. *İmamoğlu, Tuğlular & Baştanlar (2020)* proposed a machine learning approach that performs statistical damaged/undamaged classification using container images to detect structural defects. *Nguyen, Kam & Cheng (2014)* introduced a method combining phase symmetry enhancement and cubic spline fitting to leverage the symmetry and linear characteristics of cracks, thereby enabling effective crack-edge detection. *Enikeev, Gubaydullin & Maleeva (2017)* proposed a computer vision-based technique that integrates image preprocessing, binarization, contour detection, and feature analysis for detecting and identifying corrosion damage on metal surfaces, with fractal analysis used to characterize the corrosion patterns. *Wang (2018)* presented a damage detection method based on high-density laser point clouds, integrating digital image processing with point cloud data and employing filtering, morphological operations, adaptive thresholding, and multi-scale tensor voting. Although these image-processing- and machine-learning-based methods have made notable progress in damage detection, traditional machine learning approaches that rely on handcrafted features often fail to capture the intrinsic characteristics of defects, leading to reduced detection accuracy in complex scenarios and falling short of practical requirements (*Sami et al., 2023*; *Zhao et al., 2021*; *Roy & Bhaduri, 2023*).

Deep learning methods can automatically extract features and demonstrate robustness and adaptability in complex environments (*Liu et al., 2024*; *Yeon et al., 2022*; *Sami et al., 2023*; *Khanam & Hussain, 2024*). *Wang et al. (2021)* proposed a multi-class container damage detection approach based on transfer learning and MobileNetV2 (*Haralambides,* 2019). Kattainen (2019) conducted a comparative study of Single Shot MultiBox Detector (SSD), Faster Region-based Convolutional Neural Network (Faster R-CNN), and RetinaNet, employing MobileNet and ResNet as backbone networks. A dedicated container dataset was constructed, and experiments were performed using the TensorFlow Object Detection API. Bahrami et al. (2022) developed a container corrosion defect detection method based on HRTC R-CNN and optical flow image stitching, achieving high accuracy in corrosion detection tasks. Lin, Changming & Rigui (2021) proposed an improved YOLOv4-based container damage detection method that enhances detection performance through dataset clustering and an optimized classification loss function. Pham & Chang (2023) proposed an improved YOLO algorithm called YOLO-SO for detecting appearance defects in the metal base of TO-can packaged laser diodes. It integrates an attention mechanism Convolutional Block Attention Module (CBAM), a small object data augmentation module (RPM), and K-means++ clustering to enhance detection accuracy and performance (Pham & Chang, 2023). Liu et al. (2024) introduced a Coordinate Attention (CA) mechanism into the YOLOv8 backbone to improve container damage recognition and localization capabilities. Chen, Dong & Wan (2024) proposed an improved YOLOv5 model enhanced with a Transformer-based self-attention mechanism. Swin Transformer blocks were integrated into the backbone to strengthen global feature extraction, while the neck structure was optimized to improve container damage detection performance (Chen, Dong & Wan, 2024).

Despite the remarkable progress achieved by existing deep learning approaches in container defect detection, these methods still encounter significant challenges when applied to complex and dynamic real-world port environments. Ambiguous defect boundaries, illumination fluctuations, and background interference often degrade detection accuracy and model stability. Therefore, it is essential to design an adaptive and uncertainty-aware detection framework capable of effectively handling multi-scale, noisy, and irregular defect patterns while maintaining robustness and generalization. Based on this motivation, several key challenges are identified and addressed as follows:

(1) Traditional convolutional operations, constrained by a fixed receptive field, often fail to simultaneously capture both local details and global semantics when dealing with irregularly shaped or multi-scale defect regions, leading to suboptimal detection accuracy for small-scale or blurry-edge targets. (2) Existing attention mechanisms primarily focus on shallow feature selection at the channel or spatial level, making it difficult to consistently attend to critical defect regions across different scales and semantic levels, thereby limiting model robustness. (3) In the feature fusion process, detection heads typically rely on static feature concatenation and weighting strategies, lacking the ability to adaptively adjust for spatial distribution and semantic hierarchy, which results in limited accuracy in bounding box regression for multi-scale objects. (4) Publicly available container defect datasets often suffer from limited sample sizes and annotation errors, which significantly hinder the generalization capability of models in complex industrial scenarios.

To address the aforementioned challenges, we propose UPAD-YOLO, a high-precision object detection model based on improvements to the single-stage YOLOv11 framework

(*Khanam & Hussain, 2024*). The model integrates multi-dimensional innovations to balance detection accuracy and efficiency, while enhancing robustness in complex environments. The main contributions of this article are summarized as follows:

(1) An Uncertainty-aware Dynamic Modulated Convolution (UDMConv) is designed to dynamically adjust the receptive field of convolutional kernels, enabling adaptive capture of both local and global features of damage targets at different scales. This design effectively reduces the interference of complex backgrounds (*e.g.*, rust, text) on small defect regions.

(2) The Parallelized Patch-Aware Attention (PPA) module is introduced, utilizing a multi-branch feature extraction strategy to capture discriminative information across multiple scales and semantic levels. This enhances the model's focus on defective regions while suppressing irrelevant background noise.

(3) The detection head is combined with the Adaptively Spatial Feature Fusion (ASFF) module to improve the efficiency of feature pyramid fusion through multi-scale dynamic feature allocation and distributed bounding box regression. This integration enhances localization accuracy for multi-scale targets while maintaining real-time performance.

(4) A container defect dataset containing 680 images is constructed, encompassing three representative defect categories: Dent, Hole, and Deframe (Deframe refers to structural frame deformation). The dataset is utilized for training and evaluating the model, serving as a fundamental resource for related research.

The novelty of this work lies in the design of an uncertainty-driven detection framework that enhances adaptability and robustness for container defect detection. The proposed UDMConv module introduces an uncertainty-aware dynamic modulation mechanism to adaptively re-weight multi-scale features according to spatial activation variance. The PPA module applies a parallel patch-wise attention mechanism to capture fine-grained structural cues on small, localized defects under complex port environments. Additionally, the integration of ASFF ensures effective cross-scale feature fusion and consistency across the detection head. These innovations collectively distinguish the proposed UPAD-YOLO from conventional YOLO-based and transformer-based approaches, establishing a specialized yet generalizable framework for industrial defect detection.

METHOD

In this section, we provide a detailed discussion of UPAD-YOLO. As shown in Fig. 1, UPAD-YOLO is an enhanced version of the YOLOv11 architecture, augmented with three key modules: UDMConv, PPA (*Xu et al., 2024*), and ASFF (*Liu, Huang & Wang, 2019*). The integration of these modules makes the network more suitable for container damage detection and effectively addresses the challenges of multi-scale damage and complex backgrounds. Next, the overall structure of UPAD-YOLO is outlined, after which the UDMConv, PPA, and ASFF modules are described in detail in the following subsections.

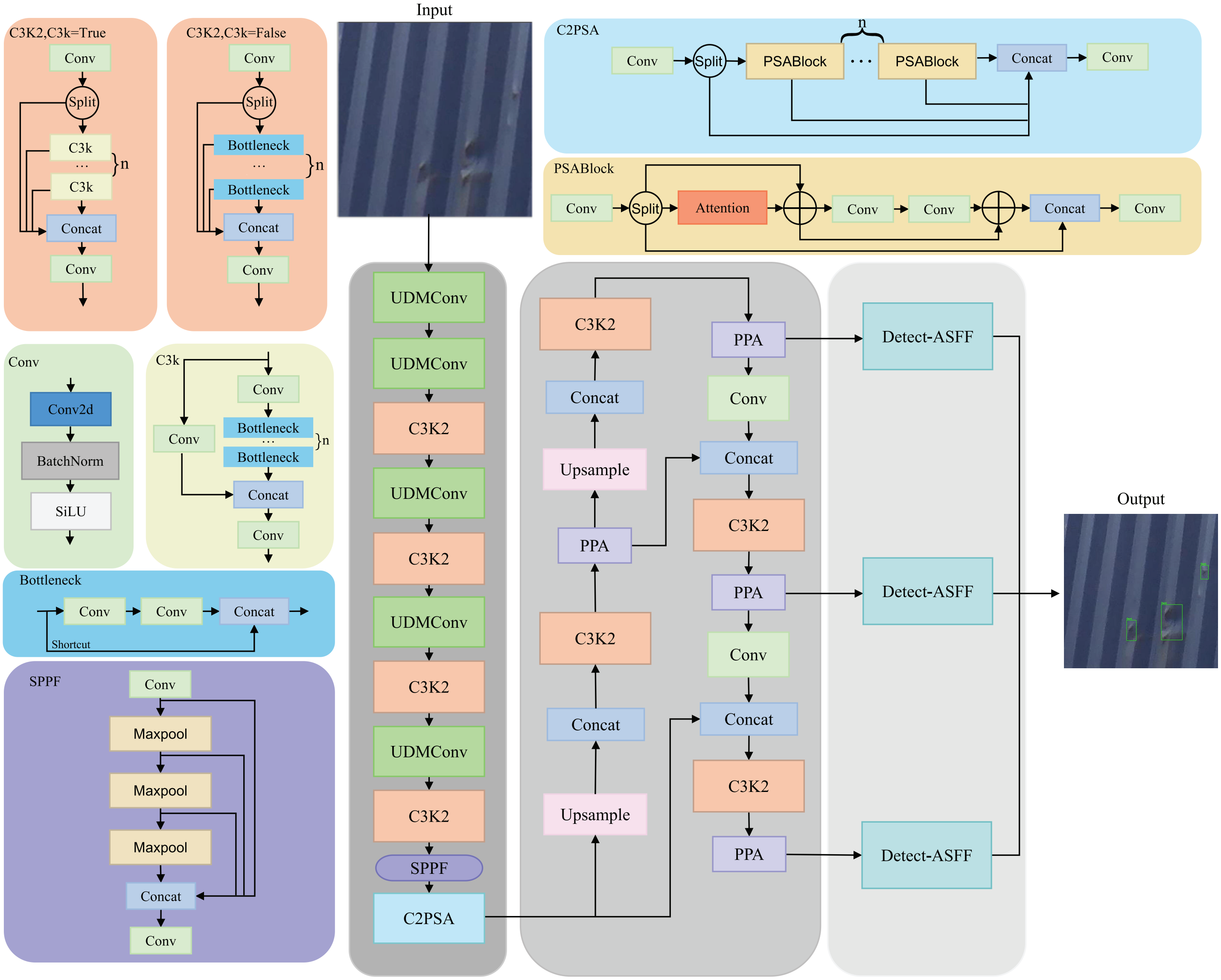

Figure 1 **Model structure diagram of UPAD-YOLO.** The network is composed of three main components: the backbone, the attention-enhanced neck, and the detection head. The backbone incorporates UDMConv blocks and multiple C3K2 modules to extract deep semantic features, followed by SPPF and C2PSA modules to enhance receptive fields and contextual understanding. The neck employs upsampling and concatenation operations integrated with PPA (Parallelized Patch-Aware Attention) modules to refine multi-scale features. Finally, Detect-ASFF modules are used in the detection head to adaptively fuse multi-level features, improving accuracy for small and dense object detection. Key blocks such as PSA, C2PSA, and SPPF are shown to demonstrate their internal structure and functionality. Full-size DOI: 10.7717/peerj-cs.3627/fig-1

Overall framework of UPAD-YOLO

As illustrated in Fig. 1, the proposed UPAD-YOLO framework is developed based on the YOLOv11 architecture, with structural enhancements designed to improve adaptability in industrial defect detection scenarios characterized by small object sizes, dense defect distributions, and complex backgrounds. The backbone adopts a Conv–BN–SiLU stem layer followed by a hierarchical downsampling structure that generates three main feature maps at different scales. For an input image of size 640 × 640, the output feature maps are

$P_3 \in \mathbb{R}^{C_3 \times 80 \times 80}$, $P_4 \in \mathbb{R}^{C_4 \times 40 \times 40}$, and $P_5 \in \mathbb{R}^{C_5 \times 20 \times 20}$. Each stage employs a C3K2 residual block—composed of two consecutive 3 × 3 Conv−BN−SiLU layers with split–concatenation connections—as the fundamental computational unit. After every downsampling operation and before entering the corresponding C3K2 stack, a UDMConv (Uncertainty-Aware Dynamic Modulated Convolution) module is inserted to dynamically adjust the receptive field and adaptively extract multi-scale contextual features. At the top of the backbone, SPPF (Spatial Pyramid Pooling-Fast) and C2PSA (Cross-Stage Partial Parallel Split Attention) modules are sequentially employed to expand the effective receptive field and model both channel-wise and spatial dependencies, thereby providing semantically rich and context-aware high-level features for subsequent processing. The neck adopts a bidirectional feature fusion topology combining Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) paths, which enables efficient top-down and bottom-up multi-scale feature interaction. From (P_5, P_4, P_3), the neck performs a sequence of upsampling, concatenation, and downsampling operations to construct cross-scale aggregated features. After each concatenation operation, a Parallelized Patch-Aware Attention (PPA) module is integrated. The PPA enhances discriminative regions and suppresses background noise by combining patch-level channel and spatial attention in a parallel manner. After convolutional refinement, the neck outputs three feature maps $(\widetilde{P}_3, \widetilde{P}_4, \widetilde{P}_5)$ corresponding to strides of 8, 16, and 32. The detection head consists of three Detect-Adaptively Spatial Feature Fusion (ASFF) heads responsible for adaptive multi-scale fusion and object prediction. Each head first aligns the multi-scale features to a uniform spatial size and channel dimension using bilinear interpolation and 1 × 1 convolution. Then, adaptive fusion coefficients are generated through a soft attention mechanism to dynamically balance the contributions from different scales and unify contextual information across layers. The fused feature is then passed through the classification, objectness, and bounding-box regression branches to generate the final detection outputs. Through this structured integration, UPAD-YOLO preserves the efficiency of YOLOv11 while significantly improving flexibility in feature extraction, adaptability in multi-scale fusion, and robustness in small-object detection. The framework explicitly defines the input–output dimensions, module insertion positions, and data-flow sequence across the backbone, neck, and head, providing a clear and reproducible architectural foundation for subsequent experimentation and deployment.

Uncertainty-aware dynamic modulated convolution

In container damage detection tasks, to address the limitations of traditional convolution operations, such as a narrow receptive field, insufficient flexibility in multi-scale feature fusion strategies, and poor computational efficiency, an Uncertainty-aware Dynamic Modulated Convolution (UDMConv) model is proposed. As shown in Fig. 2,Through the collaborative design of multi-branch parallel depthwise separable convolution processing and a dynamic feature weighting mechanism, adaptive multi-scale feature fusion is achieved. Unlike conventional adaptive convolution models that passively learn spatial offsets or channel weights, UDMConv incorporates an uncertainty-aware dynamic modulation strategy to re-weight multi-scale features according to confidence levels. This enables the model to focus on ambiguous regions caused by dent, hole, or deframe defects, thereby providing a more robust and reliable solution for industrial container-defect detection.

For the input feature map $X \in \mathbb{R}^{S\times C\times H\times W}$, three parallel Depthwise Separable Convolution are applied, each with different dilation rates d_i, to capture both local and long-range feature information.

$$F_i = \mathrm{Conv}_{d_i}(X), \quad i \in \{1,2,3\}. \tag{1}$$

Here, $\mathrm{Conv}_{d_i}(\cdot)$ denotes a depthwise separable convolution with dilation rate d_i. This operation produces feature maps F_1, F_2, F_3, each having the same number of channels as the input X, but with different receptive fields. To adaptively fuse features with varying receptive fields, a learnable channel-wise modulation coefficient $\alpha_i \in \mathbb{R}^{1\times C\times 1\times 1}$ is introduced. Each α_i is a trainable tensor initialized using Xavier initialization, allowing for per-channel independent scaling. This dynamic modulation mechanism effectively reduces uncertainty, emphasizes task-relevant features, and suppresses redundant information. The process is formulated as:

$$F'_i = \alpha_i \cdot F_i. \tag{2}$$

Due to the varying receptive fields resulting from different dilation rates, bilinear interpolation is employed to spatially align the feature maps and ensure consistency across spatial dimensions.

$$\tilde{F}_i = \mathrm{Interpolate}(F'_i, \mathrm{size} = F_1). \tag{3}$$

The aligned feature maps are subsequently concatenated along the channel dimension to integrate multi-scale information, resulting in a feature map with dimensions $S \times (3C) \times H \times W$.

$$F_{\mathrm{cat}} = \mathrm{Concat}(\tilde{F}_1, \tilde{F}_2, \tilde{F}_3). \tag{4}$$

To reduce computational complexity and further fuse multi-scale features, a pointwise convolution is applied to compress the channel dimension. This is followed by batch normalization for feature normalization, and a ReLU activation function is applied to enhance training stability and nonlinear representational capacity.

$$Y_{\mathrm{final}} = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}_{1\times 1}(F_{\mathrm{cat}}))). \tag{5}$$

Parallelized patch-aware attention

The PPA consists of two core components: multi-branch feature extraction and feature fusion with an attention mechanism. PPA extracts local, global, and multi-level semantic features through multiple branches, thereby enhancing the detailed representation of damaged container regions. By incorporating both channel and spatial attention mechanisms, it suppresses background interference such as container texture and illumination, and focuses on subtle features in damaged areas.

As illustrated in Fig. 3, a parallel multi-branch architecture is adopted, where each branch is designed to extract features at different scales and semantic levels. This

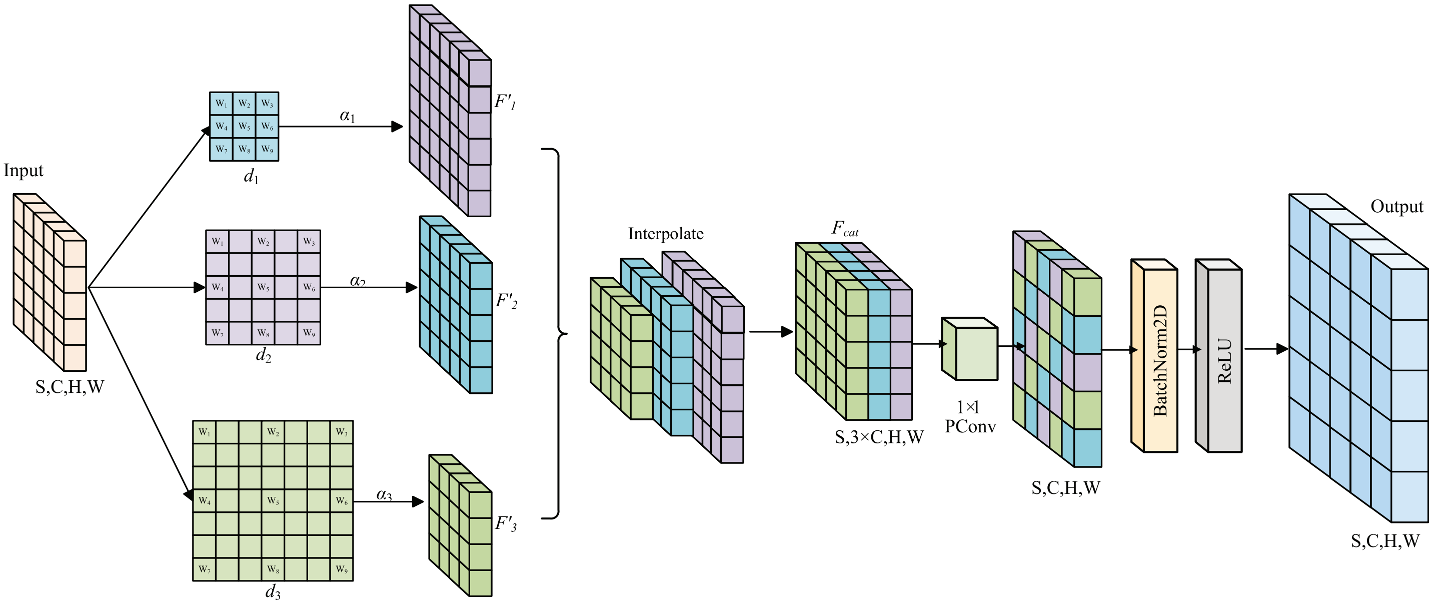

Figure 2 Overall architecture of the Uncertainty-Aware Dynamic Modulated Convolution (UDMConv) module. The input feature map is processed through three parallel dilated depthwise separable convolutions with different dilation rates (d_1, d_2, d_3) to generate multi-scale features (F'_1, F'_2, F'_3). These feature maps are interpolated and fused to form F_{cat}, which is then refined using pointwise convolution, batch normalization, and activation to produce the final output. Full-size DOI: 10.7717/peerj-cs.3627/fig-2

multi-branch strategy facilitates the capture of multi-scale damage features in containers, thereby improving detection accuracy. Specifically, the input feature map $\mathbf{F} \in \mathbb{R}^{H'\times W'\times C}$ is first transformed *via* attention-based convolution into a resized representation $F' \in \mathbb{R}^{H'\times W'\times C'}$, which is then processed through three parallel branches to obtain $\mathbf{F}_{local} \in \mathbb{R}^{H'\times W'\times C}$, $\mathbf{F}_{global} \in \mathbb{R}^{H'\times W'\times C'}$, $\mathbf{F}_{conv} \in \mathbb{R}^{H'\times W'\times C'}$. Finally, these three results are summed to produce the final output $\tilde{\mathbf{F}} \in \mathbb{R}^{H'\times W'\times C'}$.

As illustrated in Fig. 4, the local and global branches are differentiated by modulating the block size parameter p, which is achieved through aggregation and displacement of non-overlapping blocks along spatial dimensions. Furthermore, we compute attention matrices between non-overlapping patches to enable simultaneous extraction and interaction of local and global features. In both $\mathbf{F}_{local}$ and $\mathbf{F}_{global}$ branches, the feature map $\mathbf{F}'$ is partitioned into $N = \frac{H'}{p} \times \frac{W'}{p}$ spatially contiguous local patches of size $p \times p$ *via* Unfold and Reshape operations, formally expressed as $\mathscr{P} = \text{Unfold}(\mathbf{F}') \in \mathbb{R}^{N\times(p^2\cdot C')}$. This operation results in each local patch containing C channels. We therefore compute channel-wise averaging across all C channels for each patch, generating a novel feature representation: $\overline{\mathscr{P}}_i = \frac{1}{C'}\sum_{c=1}^{C'} \mathscr{P}_{i,c} \in \mathbb{R}^{p^2}$. The averaged features then undergo nonlinear transformation through a two-layer Multilayer Perceptron (MLP) to enhance representational capacity: $Z_i = W_2 \cdot \text{GELU}(W_1 \cdot \overline{\mathscr{P}}_i) \in \mathbb{R}^{C'}$, where $W_1 \in \mathbb{R}^{p^2\to C'/2}$, $W_2 \in \mathbb{R}^{C'/2\to C'}$. To facilitate interaction between local and global features, a feature selection mechanism is introduced. Specifically, cosine similarity functions are employed to measure affinity between each token and task embeddings, followed by similarity-based

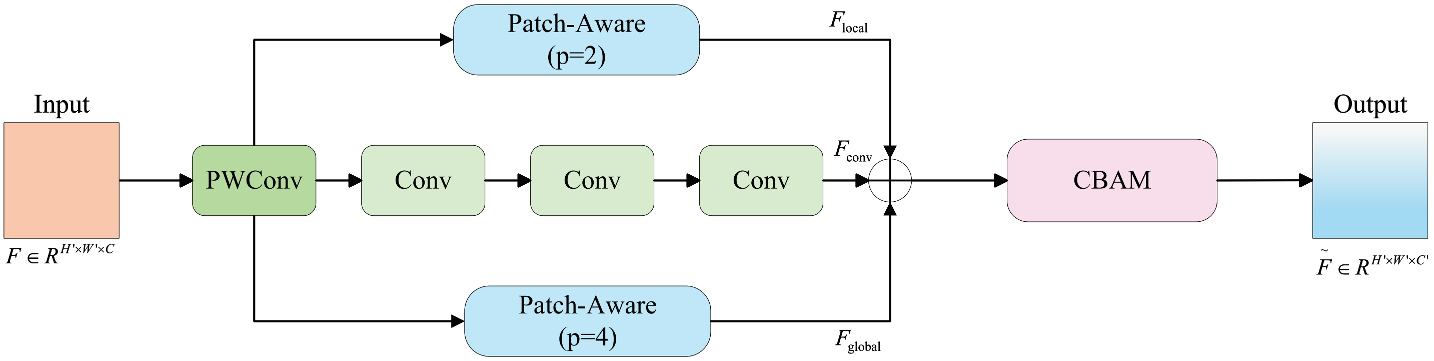

Figure 3 Structure of the parallelized patch-aware attention module. Structure of the parallelized patch-aware attention module. The input feature map is first processed by point-wise and standard convolutional layers to extract the central features F_{conv}. Simultaneously, two parallel patch-aware branches with patch sizes $p = 2$ and $p = 4$ extract local and global contextual features F_{local} and F_{glocal}, respectively. These are aggregated and passed to a CBAM (Convolutional Block Attention Module) to generate the attention-enhanced output.

Full-size DOI: 10.7717/peerj-cs.3627/fig-3

weighting of individual tokens. Finally, these weighted features are projected to the final convolutional output through a convolutional layer.

Following multi-branch feature extraction, a Convolutional Block Attention Module (CBAM) mechanism is employed for feature enhancement, as shown in Fig. 5. This mechanism comprises two components: efficient channel attention and spatial attention modules. In this context, the tensor $\tilde{\mathbf{F}} \in \mathbb{R}^{H' \times W' \times C'}$ undergoes MaxPool and AvgPool operations along spatial dimensions, yielding two $1 \times 1 \times C'$ feature maps. This spatial dimension compression facilitates channel-wise feature learning by reducing feature map resolution. The pooled features from both MaxPool and AvgPool are processed through a shared MLP, producing two refined $1 \times 1 \times C'$ feature maps. These MLP outputs are element-wise summed and subsequently processed through a sigmoid activation function, generating the channel attention weight matrix $M_c \in \mathbb{R}^{1 \times 1 \times C'}$. Global MaxPool and Global AvgPool operations are applied to $\tilde{\mathbf{F}} \in \mathbb{R}^{H' \times W' \times C'}$, producing two $H' \times W' \times 1$ feature maps. These global-pooled features are channel-wise concatenated, resulting in a combined feature map of dimensions $H' \times W' \times 2$. A convolutional layer transforms the concatenated features to $H' \times W' \times 1$ resolution, followed by sigmoid activation to produce the spatial attention weight matrix $M_s \in \mathbb{R}^{H' \times W' \times 1}$. The enhanced feature representation $\tilde{\mathbf{F}}' \in \mathbb{R}^{H' \times W' \times C'}$ is ultimately obtained through integration of both attention matrices.

Adaptively spatial feature fusion

Containers exhibit substantial variations in damage scales and types. Small-scale dents typically require high-resolution features to capture fine-grained details, medium-sized perforations rely on macroscopic features for rapid localization and identification, while large-scale structural damages necessitate multi-range feature fusion to precisely delineate morphology and extent. Conventional feature fusion approaches struggle to adequately integrate these multi-scale variations, whereas the ASFF mechanism effectively addresses such limitations.

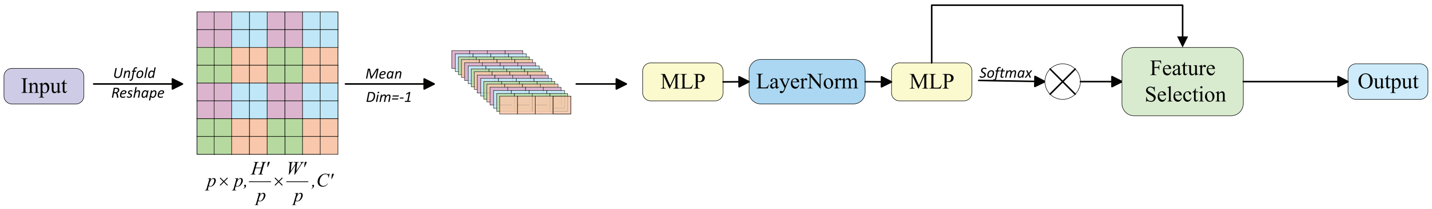

Figure 4 Patch-Aware structure diagram. The input feature is reshaped into patches, followed by mean pooling to generate patch-level descriptors. These descriptors are processed by a multilayer perceptron (MLP), layer normalization, and another MLP to generate attention weights *via* softmax. The weights are applied to perform feature selection and produce the output representation. Full-size DOI: 10.7717/peerj-cs.3627/fig-4

The ASFF module primarily adopts target detection resolution as its reference scale, enabling flexible adaptation to multi-level features. As depicted in Fig. 6, the hierarchical architecture demonstrates progressively increasing resolution and contracting receptive fields from Level 1 to Level 3. To optimize detection performance, the module enhances contextual representation through fusion of Level 1 and Level 2 feature maps, achieving synergistic multi-level feature enhancement.

Specifically, spatial importance weights are generated for feature maps at each scale. Taking Level 3 feature fusion as an example (see Fig. 6), let $x_{pq}^{1\to 3}$, $x_{pq}^{2\to 3}$ and $x_{pq}^{3\to 3}$ denote the feature values at position (p, q) from Level 1 to Level 3 respectively. The fused feature at Level 3 is formulated as:

$$y_{pq}^{3} = \lambda_{pq}^{3}\cdot x_{pq}^{1\to 3} + \mu_{pq}^{3}\cdot x_{pq}^{2\to 3} + \nu_{pq}^{3}\cdot x_{pq}^{3\to 3} \tag{6}$$

where $\lambda_{pq}^{3}\mu_{pq}^{3}$ and ν_{pq}^{3} represent learnable spatial weighting parameters subject to the constraints: $\lambda_{pq}^{3} + \mu_{pq}^{3} + \nu_{pq}^{3} = 1$ with $\lambda_{pq}^{3}, \mu_{pq}^{3}, \nu_{pq}^{3} \in [0, 1]$. This adaptive mechanism enables dynamic weight allocation based on the relative contributions of multi-scale features to container damage detection. The principal advantage of ASFF integration into the YOLOv11 detection head resides in its capability for dynamic multi-scale feature weighting. Specifically engineered to address inherent multi-scale challenges in container damage inspection, this design significantly enhances detection accuracy through adaptive optimization of cross-scale feature contributions. Concurrently, the method effectively suppresses false positives and missed detections induced by scale variations, thereby bolstering model robustness in complex scenarios.

EXPERIMENTS

Experimental setup

The experiments were conducted using the PyTorch deep learning framework within an Anaconda environment. The hardware configuration details are summarized in Table 1, and the hyperparameter settings are documented in Table 2.

Dataset construction

The dispersed nature of container damage locations during transportation, coupled with the requirement for specialized equipment and personnel in data acquisition, results in exorbitant costs for dataset collection. Consequently, no publicly available high-quality

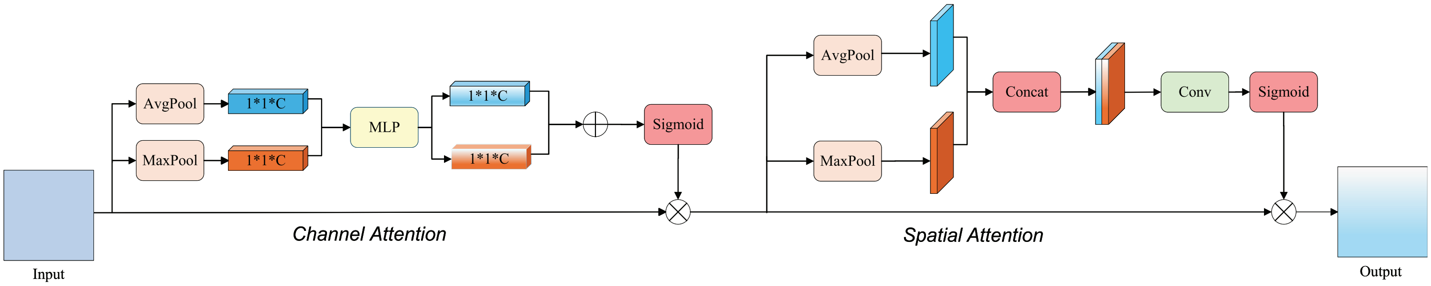

Figure 5 **Convolutional block attention module structure diagram.** Detailing the progression of input data through channel attention followed by spatial attention, culminating in the enhanced output. Full-size DOI: 10.7717/peerj-cs.3627/fig-5

container defect datasets currently exist. To address this limitation, we constructed a proprietary dataset for experimental validation. Existing container-damage detection datasets contain relatively limited samples (*e.g*., Defect Container Detection: 385 images (*Defect, 2024*); Containers Damage Detect V2: 549 images (*Fianna, 2024*)), whereas the dataset in this study comprises 680 high-resolution RGB images captured under real port conditions, providing a larger and more representative sample base for model training and evaluation. The 680-image dataset contains three annotated defect categories: Dent, Hole, and Deframe (Deframe refers to structural frame deformation), with manual annotations illustrated in Fig. 7. The dataset was partitioned into training (544 images), validation (68 images), and test sets (68 images).

Dataset description

To further improve the transparency and reproducibility of this study, this subsection provides additional details supplementing the Dataset Construction section. The image data were collected by port personnel during regular operations, capturing realistic variations in lighting, background, and surface textures in authentic working environments. Container inspection staff assisted in and verified the manual annotation process to ensure domain consistency and labeling precision. The dataset consists of 680 high-resolution (1,920 × 1,080 pixels) RGB images, acquired using a HIKROBOT MV-CU020-19GC V2 industrial camera at the port container yard. Each image is annotated into one of three defect categories: Dent, Hole, or Deframe (structural frame deformation). To ensure training consistency, all images were resized to 640 × 640 pixels before being used in model training. To promote academic openness and facilitate subsequent research, the full dataset has been publicly released on the Kaggle platform (DOI: 10.34740/kaggle/dsv/12436314).

Evaluation metrics

To comprehensively evaluate model performance in container defect detection, this study adopts widely-used object detection metrics: Precision (P), Recall (R), mAP@50, mAP@50:95, and F1-Confidence. Precision quantifies detection accuracy by measuring the proportion of true defects among all predicted defects. A higher precision indicates

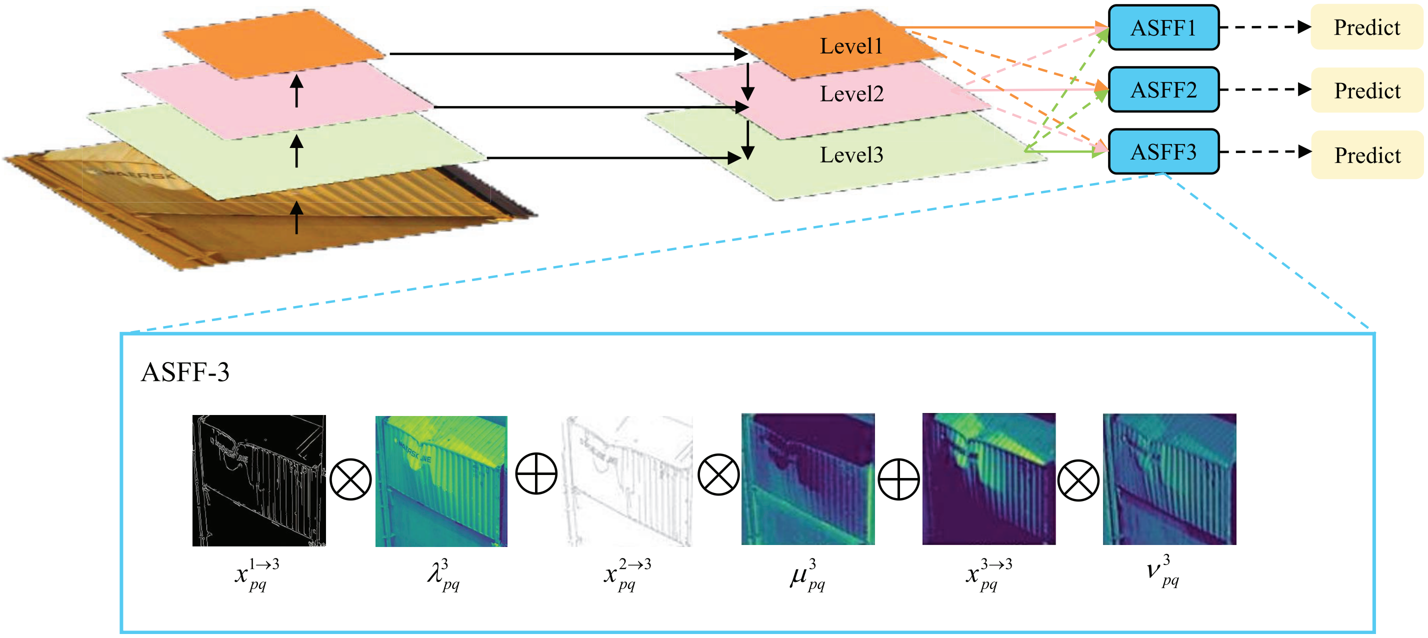

Figure 6 Schematic diagram of the Adaptively Spatial Feature Fusion (ASFF) module, illustrating the multi-level feature fusion process and the subsequent prediction steps. The inset details the ASFF-3 process, showing the iterative feature fusion and enhancement mechanism.

Full-size DOI: 10.7717/peerj-cs.3627/fig-6

Table 1 Experimental environment configuration.

Configuration item	Specification
Operating system	Windows 11 Professional 64-bit
CPU	Intel(R) Core(TM) i9-10900K @ 3.70 GHz, 10 cores
GPU	NVIDIA GeForce RTX 3070 (8 GB VRAM)
RAM	32 GB DDR4
CUDA version	CUDA Toolkit 12.8
cuDNN version	cuDNN 8.9
Development environment	PyCharm 2024.1
Programming language	Python 3.9.21
Deep learning framework	PyTorch 2.8.0
TorchVision version	torchvision 0.19.0
Package management	Anaconda (conda 24.3.0)

enhanced capability in distinguishing true defects from intact regions, thereby reducing false positives. The calculation follows:

$$P = \frac{TP}{TP + FP} \tag{7}$$

where TP (True Positive) denotes correctly identified defect regions, and FP (False Positive) represents misclassified intact areas. Recall measures the detection completeness as the ratio of true defects successfully identified by the model. This metric reflects the

Table 2 Hyperparameter settings used in training.

Hyperparameter	Value
Number of epochs	300
Batch size	8
Input image size	640 × 640
Optimizer	SGD
Initial learning rate	0.001
Learning rate scheduler	Cosine annealing
Cosine scheduler enabled	True
Weight decay	0.0005
Momentum	0.9
Number of workers	4
Cache mode	Disk
Automatic Mixed Precision (AMP)	True
Random seed	42

model's ability to minimize false negatives during defect screening. High recall values ensure maximal defect capture for subsequent processing (*e.g.*, maintenance and classification), critical for operational efficiency in port logistics. Computed as:

$$R = \frac{TP}{TP + FN}. \tag{8}$$

Here, FN (False Negative) denotes the number of actual damaged regions that are mistakenly classified as normal by the model.

The mean Average Precision (mAP) is a comprehensive metric that combines both precision and recall. The AP value ranges from 0 to 1, with higher values indicating better model performance. The mAP is calculated as the mean of AP values across all categories. In container damage detection tasks, we typically focus on mAP@50, which refers to the mAP when the IoU threshold is set to 0.5. Some related detection tasks also emphasize the accuracy of bounding box matching. Additionally, mAP@50–95, which averages the mAP scores over IoU thresholds from 0.5 to 0.95 in steps of 0.05, is used to provide a more comprehensive assessment of model performance across varying detection stringencies. The corresponding calculation formula is as follows:

$$AP = \int_0^1 p(r)\,dr \tag{9}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i. \tag{10}$$

As an overall performance indicator, mAP enables a more holistic evaluation of the model's effectiveness in diverse damage detection tasks. To further assess the model's overall performance in container damage detection tasks, the F1-Confidence metric is introduced as a supplementary indicator. The F1-score considers the harmonic mean of

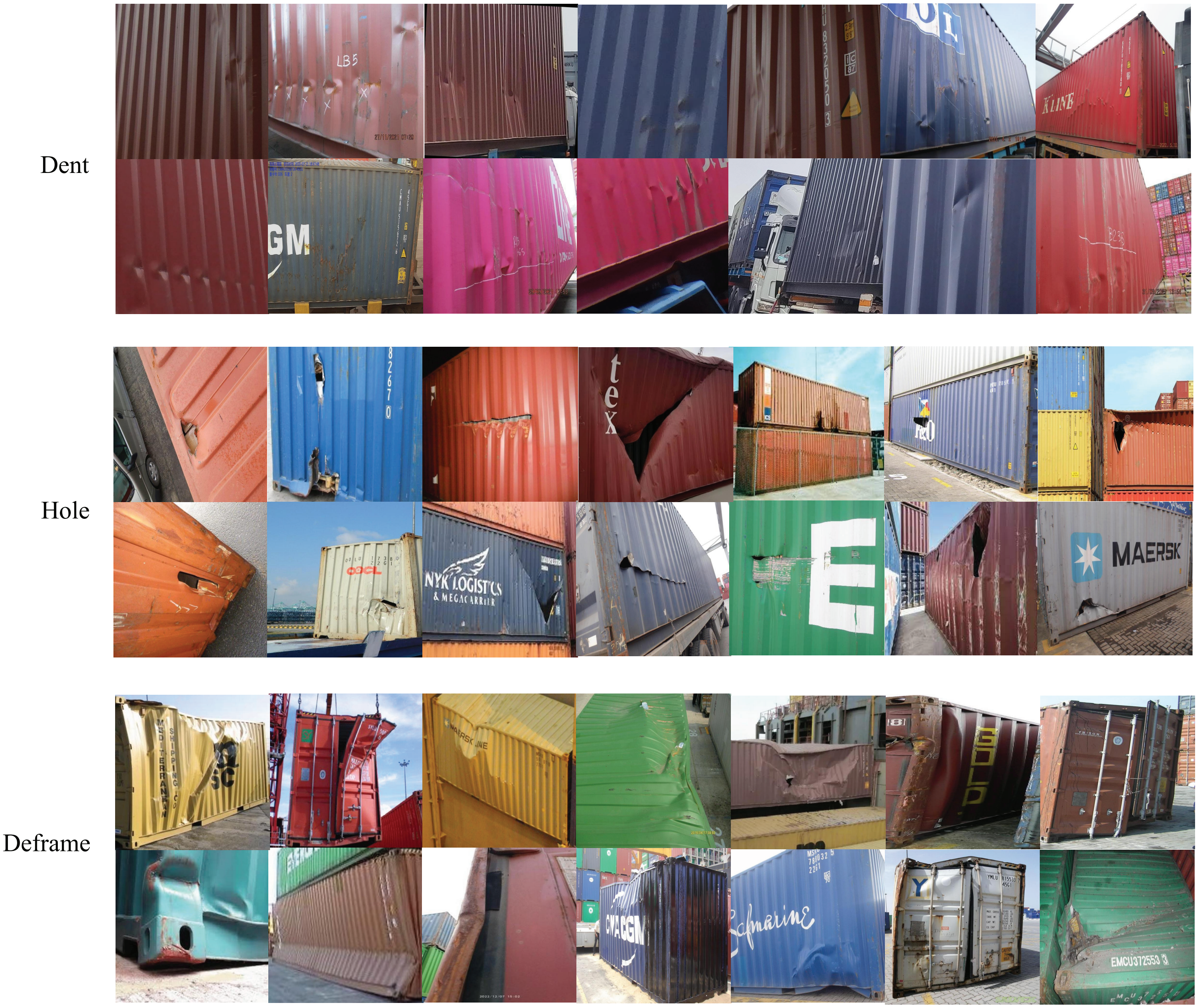

Figure 7 Sample images of three defect categories in the dataset. Representative sample images of the three defect categories in the dataset: Dent, Hole, and Deframed container. Full-size DOI: 10.7717/peerj-cs.3627/fig-7

precision and recall, making it suitable for evaluating model performance under different thresholds. The formula is given as follows:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{11}$$

Experimental results of UPAD-YOLO model

Figure 8 illustrates the Precision-Recall (P-R) curve of the proposed model, clearly demonstrating the trade-off between precision and recall across varying confidence thresholds, thereby accurately reflecting the model's performance under different confidence levels. The overall trend of the curve indicates that the model exhibits a relatively smooth P-R trajectory, maintaining a high recall while preserving a satisfactory level of precision. This suggests that the model can effectively suppress the false positive rate while ensuring comprehensive detection of container defects, thereby achieving satisfactory recognition performance.

During the training and validation phases, a series of key metrics were recorded and analyzed, as illustrated in Fig. 9. In terms of time consumption, as the number of iterations increased from 0 to 300, the training time exhibited a stable linear growth pattern, indicating that the model training progressed in an orderly manner according to the pre-defined schedule, with the time consumed per iteration remaining essentially consistent, demonstrating good controllability of the training process in terms of time. Regarding the loss-related indicators, the values of box_loss, cls_loss, and dfl_loss on the training set showed a clear downward trend as the number of iterations increased. At the early stages, these loss values were relatively high, but they decreased rapidly as the model continued to learn and optimize, stabilizing around the 200th iteration. This indicates that the model gradually learned to better fit the training data, with the decrease in box_loss reflecting improved accuracy in object localization, the decline in cls_loss indicating enhanced classification performance, and the reduction in dfl_loss suggesting better distribution fitting. On the validation set, both val/box_losss and val/cls_loss also exhibited an overall downward trend, reflecting the model's improving generalization ability throughout training, as the learned features and patterns were not only effective for the training data but also generalized well to unseen validation data, ensuring the practical applicability of the model. In terms of performance metrics, metrics/precision(B) increased steadily, indicating that the model became increasingly precise in identifying true positives while reducing false positives; metrics/recall(B) also showed a rising trend, suggesting that the model was able to detect a higher proportion of actual targets, thereby reducing missed detections; the composite metrics, metrics/$\mathrm{mAP50} - 95(\mathrm{B})$ and metrics/mAP50(B), also continued to increase with the number of iterations, clearly demonstrating continuous improvement in the model's overall performance, with steadily improving average precision across different IoU thresholds (50–95 and 50), highlighting the effectiveness of the model's ongoing refinement in object detection tasks.

To comprehensively evaluate the model's detection capability for container defects under varying confidence thresholds, Fig. 10 presents the F1-score curves for each category. As a comprehensive metric, the F1-score balances precision and recall, effectively reflecting the overall performance of the model.

Ablation experiment

The experiments presented in 'Dataset Construction' to 'Evaluation Metrics' demonstrate the superior performance of the UPAD-YOLO model. To validate the effectiveness of each improved module in the UPAD-YOLO model, a series of ablation studies was conducted on the dataset. In the experiments, the original YOLOv11 model was used as the baseline, and the UDMConv, PPA, and ASFF modules were incrementally added to compare the

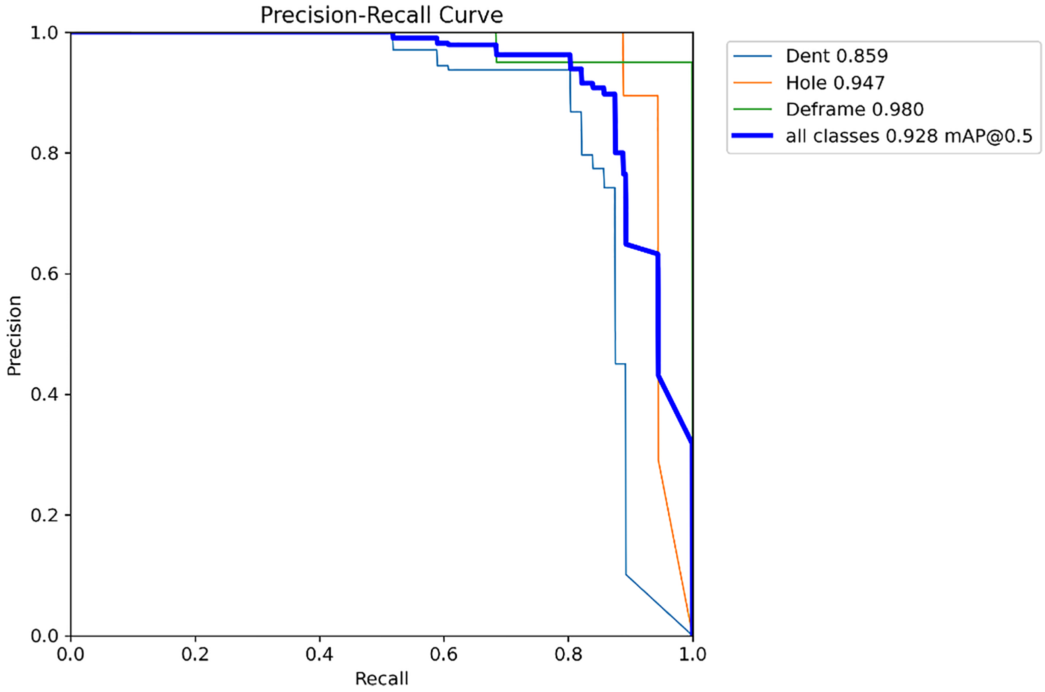

Figure 8 Precision–recall curve of UPAD-YOLO model.
Full-size DOI: 10.7717/peerj-cs.3627/fig-8

performance metrics of different model variants. The ablation experiments aim to investigate the impact of the proposed modules on container damage detection. Table 3 presents a comparison between the original model and its improved variants with the added modules.

As shown in Table 3, the YOLOv11 baseline model, without incorporating any enhancement modules, achieved a Precision (P) of 0.899, Recall (R) of 0.853, mAP@50 of 0.914, and an F1-score of 0.875, serving as the performance benchmark. Replacing the original convolution modules in the backbone with the UDMConv module led to improvements across all evaluation metrics, particularly in Recall (0.877) and F1-score (0.893), indicating enhanced feature extraction capability and improved recall performance. With the addition of the PPA module, Precision significantly increased to 0.934 and mAP@50 to 0.932. Although Recall slightly decreased to 0.838, the overall F1-score still surpassed that of the baseline, demonstrating the module's advantage in boosting precision. Integrating the ASFF module into the detection head further improved Recall to 0.882, with mAP@50 at 0.923 and F1-score rising to 0.891, suggesting enhanced feature fusion and multi-scale perception capabilities. The UPAD-YOLO model achieved the highest overall performance, with a Precision of 0.928, Recall of 0.912, mAP@50 of 0.954, and F1-score of 0.920. Overall, the synergistic integration of the proposed modules substantially improved detection performance, confirming the effectiveness and feasibility of the enhancement strategies. The experimental results indicate that the PPA module enhances the model's spatial perception capability for multi-scale targets, while the ASFF

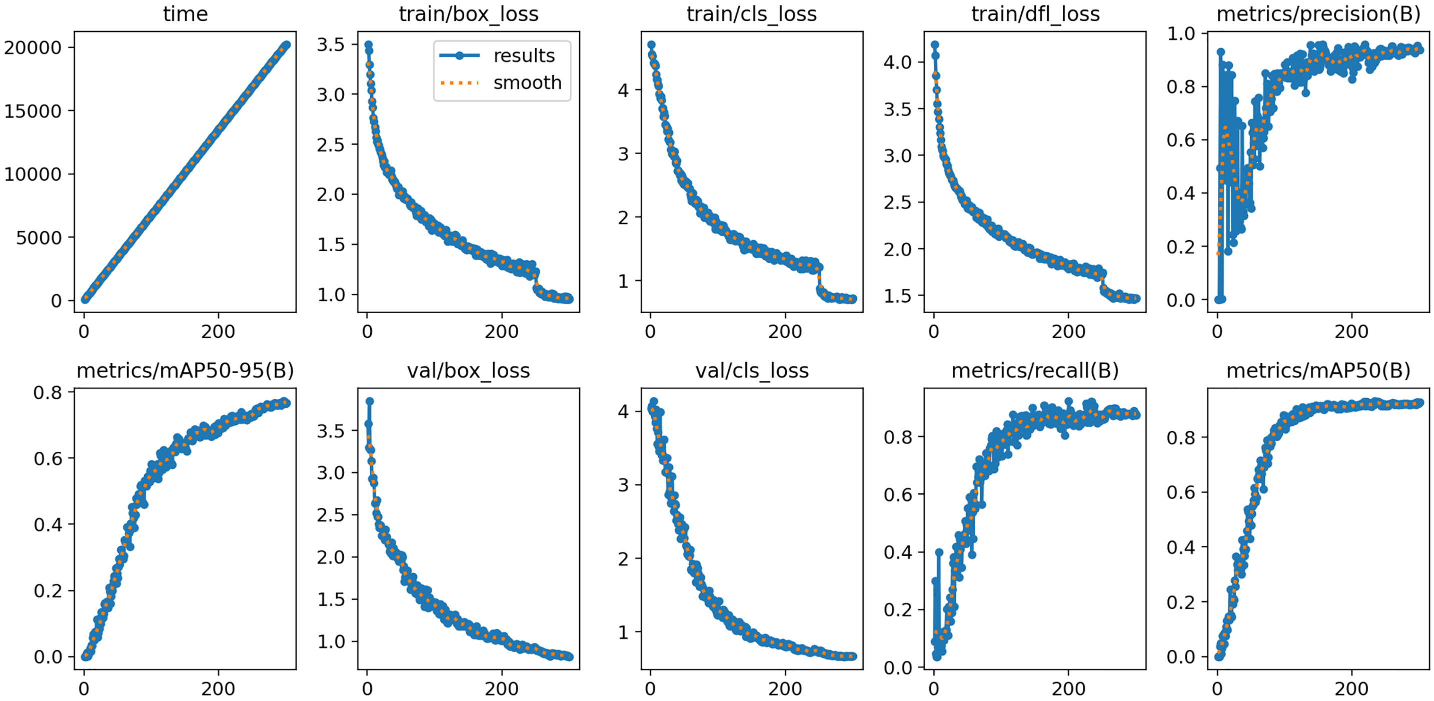

Figure 9 The figure illustrates the variation of key metrics during the training process. The top row, from left to right, shows training time, training box loss (train/box_loss), classification loss (train/cls_loss), distribution focal loss (train/dfl_loss), and precision (metrics/precision(B)). The bottom row, from left to right, shows mean average precision (metrics/mAP$_{50-95}$(B)), validation box loss (val/box_loss), validation classification loss (val/cls_loss), recall (metrics/recall(B)), and average precision (metrics/mAP$_{50}$(B)). All loss values gradually decrease as training progresses, while precision, recall, and average precision steadily improve, indicating continuous optimization of model performance.

Full-size DOI: 10.7717/peerj-cs.3627/fig-9

module optimizes the feature fusion strategy, effectively strengthening the detection heads' perception ability in critical target regions. Simultaneously, UDMConv provides a more efficient semantic representation basis during the feature extraction stage, thereby improving overall detection performance. The synergistic integration of these three modules significantly enhances container defect detection performance, with all evaluation metrics surpassing those of the baseline model, further validating the applicability and effectiveness of this improvement strategy in complex detection tasks.

To evaluate the effectiveness of the patch-size configuration in the proposed Parallelized Patch-Aware Attention (PPA) module, an ablation study was conducted by varying the patch size pwhile maintaining all other training settings identical. The tested configurations included p = 2, 4, 8 and a dual-branch configuration with p = 2,4. As summarized in Table 4, employing a small patch size (p = 2) improves fine-grained local feature extraction but limits global contextual representation, leading to fragmented responses in large deformation regions. In contrast, using a large patch size (p = 8) enhances global context perception but blurs fine defect boundaries, thereby reducing precision. The dual-branch configuration with p = 2, 4 achieves the optimal balance between local detail preservation and global semantic aggregation, yielding the highest mAP@0.5 of 0.954 and F1-score of 0.920. These findings demonstrate that the multi-scale

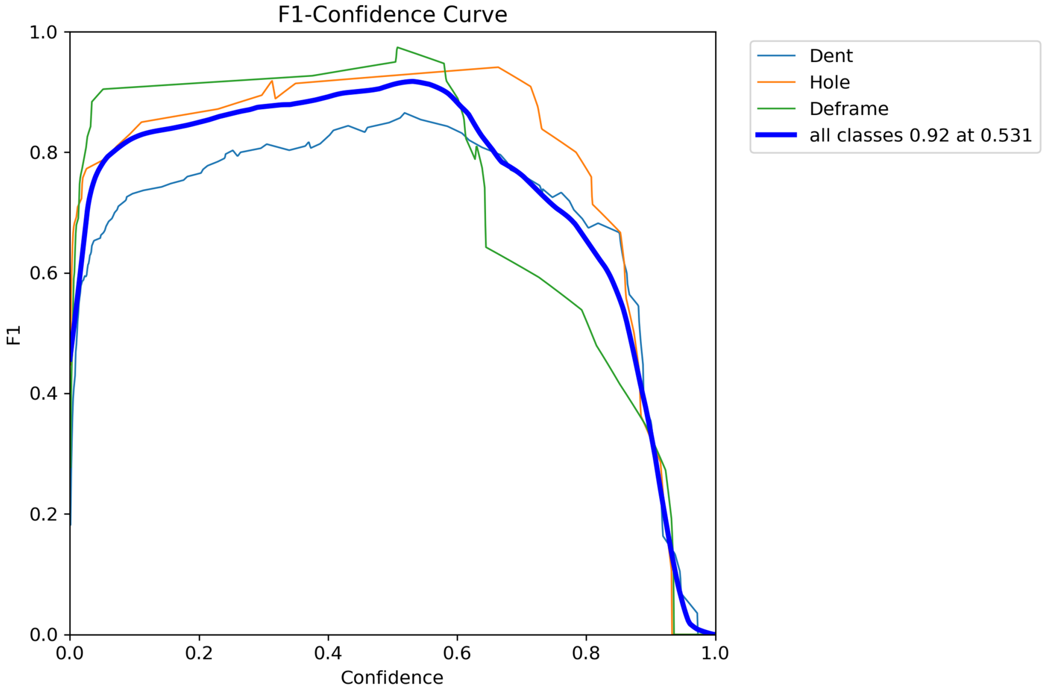

Figure 10 F1-Confidence curve showing the performance of different classes (Dent, Hole, Deframe) and the overall performance (all classes) across varying confidence thresholds. The thick blue line represents the F1-score for all classes, peaking at 0.92 when the confidence is 0.531.

Full-size DOI: 10.7717/peerj-cs.3627/fig-10

Table 3 Results of ablation experiments. YOLOv11 is the baseline model; UDMConv, PPA, and ASFF are added incrementally.

No.	YOLOv11	UDMConv	PPA	ASFF	P	R	mAP@0.5	F1
1	✓				0.899	0.853	0.914	0.875
2	✓	✓			0.911	0.877	0.912	0.893
3	✓		✓		0.915	0.881	0.920	0.883
4	✓			✓	0.901	0.882	0.923	0.891
5	✓	✓	✓	✓	**0.928**	**0.912**	**0.954**	**0.920**

Note:
The best results are highlighted in bold.

parallel patch strategy substantially enhances the adaptability of the model to diverse defect morphologies. Consequently, the dual-scale configuration ($p = 2, 4$) was adopted in the final UPAD-YOLO framework.

Comparative experiments

Three mainstream object detection frameworks were selected to validate the detection performance of the UPAD-YOLO model and to ensure representativeness across different detection paradigms—two-stage detectors (Faster R-CNN), one-stage detectors (YOLOv5, YOLOv8, YOLOv11, and its improved version SSA-YOLO), and Transformer-based detectors (RT-DETR and Hint-DETR), thereby comprehensively covering the major

Table 4 Ablation study of patch-size configurations in the PPA module.

Patch size (p)	Configuration	mAP@0.5	Precision	Recall	F1-score
2	Local-only	0.898	0.892	0.827	0.858
4	Global-only	0.925	0.903	**0.921**	0.913
8	Coarse-only	0.921	0.882	0.881	0.881
{2, 4}	Multi-scale (dual-branch)	**0.954**	**0.928**	0.912	**0.920**

Note:
The best results are highlighted in bold.

Table 5 Comparison of detection performance between different models.

Model	P	R	mAP@0.5	F1
Faster R-CNN (*Girshick, 2015*)	0.820	0.834	0.834	0.826
YOLOv5 (*Jocher et al., 2020*)	0.839	0.815	0.863	0.827
YOLOv8 (*Varghese & Sambath, 2024*)	0.846	0.836	0.890	0.841
YOLOv10 (*Wang et al., 2024*)	0.875	0.898	0.917	0.886
YOLOv11(*Khanam & Hussain, 2024*)	0.899	0.853	0.914	0.875
SSA-YOLO (*Huang, Zhu & Huo, 2024*)	0.865	0.877	0.902	0.871
RT-DETR (*Zhao et al., 2024*)	0.882	0.809	0.878	0.844
Hint-DETR (*Liu, Zhang & Li, 2025*)	0.892	0.813	0.862	0.852
UPAD-YOLO	**0.928**	**0.912**	**0.954**	**0.920**

Note:
The best results are highlighted in bold.

existing detection approaches. In addition, considering the advancement and reproducibility of the methods, all selected models were recently published and are well-maintained, with complete source codes and official pretrained weights available. Furthermore, to guarantee fairness and comparability, all models were trained under identical configurations, including dataset splits, input resolution (640 × 640), training epochs (300), and consistent data augmentation strategies. As shown in Table 5, all models were evaluated under identical training conditions using four core metrics: P, R, mAP50, and F1-score.

The experimental results demonstrate that UPAD-YOLO exhibits significant advantages in defect detection tasks: its precision (0.928) surpasses the suboptimal YOLOv11 by 3.2%, and recall (0.912) outperforms YOLOv10 by 1.6%. Regarding comprehensive performance metrics, the model achieves an mAP50 of 0.954 (4.4% improvement over the baseline YOLOv11) while maintaining a leading F1-score (0.920). Compared with the classic two-stage detector Faster R-CNN, UPAD-YOLO shows a 14.4% enhancement in mAP50. Among single-stage detectors, its F1-score exceeds YOLOv10 (0.886) and YOLOv11 (0.875) by 3.4% and 4.5%, respectively. Furthermore, to provide a more comprehensive evaluation, three recent state-of-the-art models—SSA-YOLO, RT-DETR, and Hint-DETR—were introduced for comparison. UPAD-YOLO achieves consistently superior results, with its mAP@0.5 (0.954) exceeding SSA-YOLO (0.902), RT-DETR (0.878), and Hint-DETR (0.862) by 5.2%, 7.6%, and 9.2%, respectively. These

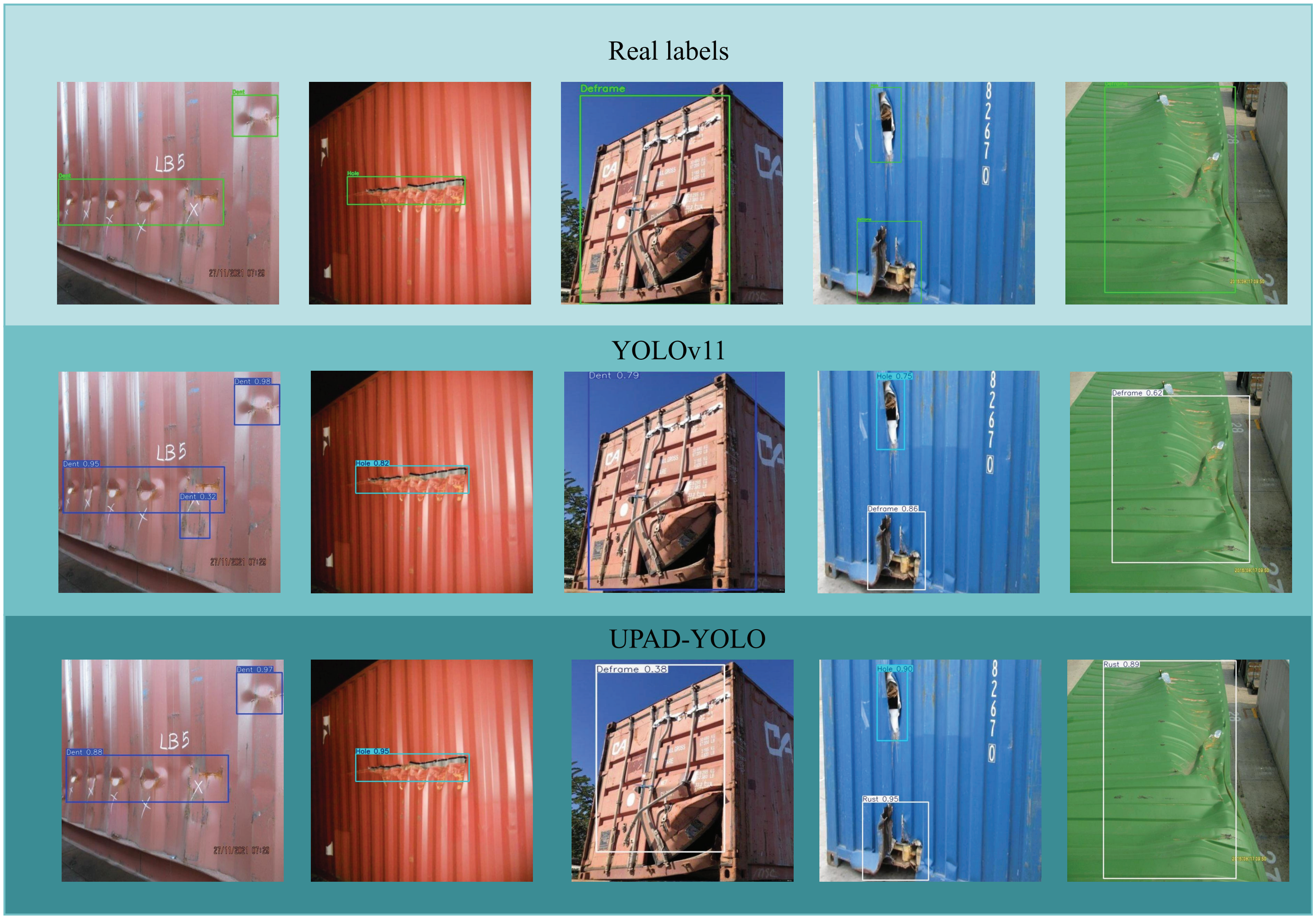

Figure 11 This figure shows a comparison of the detection results of container defects between the original model and UPAD-YOLO.
Full-size DOI: 10.7717/peerj-cs.3627/fig-11

results confirm that the proposed model not only outperforms traditional convolution-based detectors but also maintains an advantage over transformer-based architectures in both precision and overall detection performance. These performance gains are primarily attributed to the proposed multi-scale feature fusion mechanism and dynamic attention weighting strategy, effectively enhancing the model's capability to identify subtle defects on container surfaces.

Visualization of prediction results

Figure 11 presents a visual comparison of container damage detection predictions. The first row shows the ground truth labels, indicating the actual locations and types of damage on the containers. The second and third rows display the prediction results of YOLOv11 and the proposed UPAD-YOLO models, respectively. As shown in the figure, both models are capable of detecting common types of damage, such as dents and deformations. However, UPAD-YOLO demonstrates superior performance in several aspects. Higher

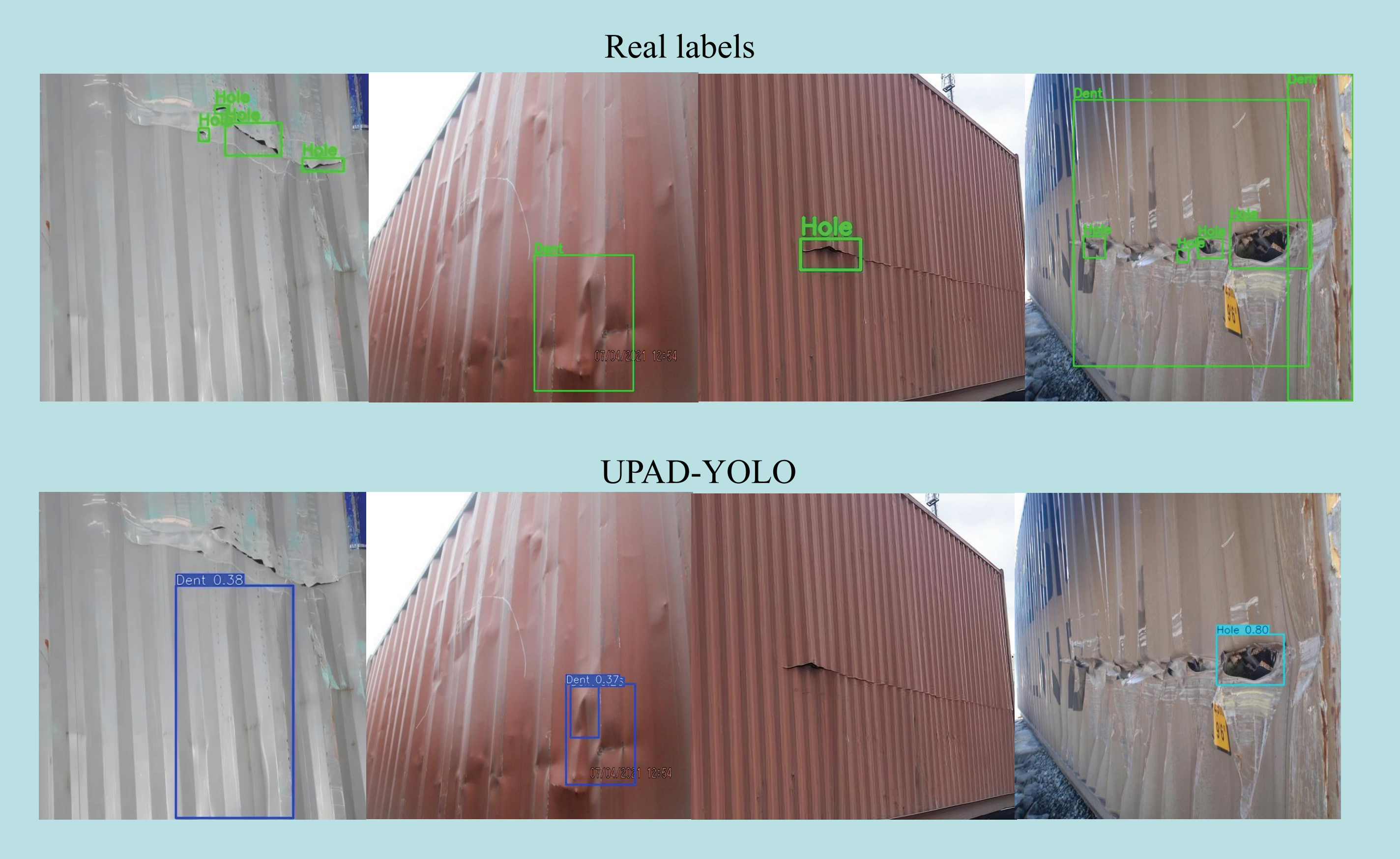

Figure 12 Qualitative comparison between real labels (top row) and UPAD-YOLO predictions (bottom row). Representative failure cases illustrate partial detection and confidence degradation under complex lighting and deformation conditions.

Full-size DOI: 10.7717/peerj-cs.3627/fig-12

localization accuracy is observed in the first two images, where the bounding boxes predicted by UPAD-YOLO exhibit greater overlap with the ground truth, while YOLOv11 fails to accurately localize certain defects. In contrast, UPAD-YOLO's bounding boxes more precisely cover the actual damaged areas. Lower false positive rate is evident in the third and fourth images. YOLOv11 produces several false positives by predicting damage where none exists, whereas UPAD-YOLO significantly reduces such errors, yielding cleaner and more reliable predictions. Improved detection of small and subtle defects can be seen in the fifth image, where UPAD-YOLO successfully identifies minor damages that YOLOv11 tends to miss, demonstrating enhanced capability in handling challenging defect cases. Overall, the visual results clearly demonstrate that UPAD-YOLO outperforms YOLOv11 in terms of detection accuracy, reliability, and robustness, making it well-suited for container damage detection applications.

Nevertheless, as illustrated in Fig. 12, certain partial detection and confidence degradation cases can still be observed under complex conditions. In particular, when surface reflections, uneven illumination, or overlapping deformations occur, UPAD-YOLO occasionally fails to capture subtle edge distortions or assigns lower confidence scores to small defect regions. For example, in the left and middle samples, low-contrast metallic surfaces cause partial misses of dent boundaries, while in the right example, strong lighting variation leads to incomplete bounding boxes and reduced confidence for hole-type defects. These observations indicate that although the proposed model maintains strong robustness and generalization in real port environments, its sensitivity to small or low-contrast defect boundaries could be further improved through adaptive illumination correction and refined uncertainty modeling in future work.

CONCLUSIONS

To address the challenges posed by multi-scale object detection, complex background interference, and stringent real-time constraints in container damage identification, this study introduces an enhanced single-stage object detection framework, termed UPAD-YOLO, which builds upon the YOLOv11 architecture. In the context of container damage detection, the proposed UDMConv module addresses key limitations of traditional convolution operations, including a restricted receptive field, inflexible multi-scale feature fusion, and suboptimal computational efficiency. To further enhance the model's focus on critical damage features and improve multi-scale target fusion, the PPA and ASFF modules are integrated into the framework. This study presents UPAD-YOLO, a high-precision and high-efficiency single-stage detection model tailored for container damage detection, designed to tackle the challenges of multi-scale defect recognition, complex background interference, and real-time detection constraints inherent in traditional approaches. To strengthen feature representation and detection robustness, UPAD-YOLO incorporates the novel UDMConv module alongside the PPA and ASFF modules. By introducing dynamic dilation rates and channel-level modulation, the UDMConv module significantly enhances the model's capacity to adapt the receptive field to varying defect scales, thereby improving multi-scale feature representation. The PPA module employs multi-branch feature extraction and integrated attention mechanisms to guide the model's focus toward defect regions while suppressing background noise, thereby enhancing the precision and consistency of feature selection. The ASFF module fuses multi-scale spatial features and dynamically reweights information across hierarchical levels, effectively enhancing the detection head's sensitivity to object boundaries and geometries. Experimental evaluations on a self-constructed container defect dataset demonstrate that UPAD-YOLO outperforms YOLOv11 and other state-of-the-art models across multiple metrics, including precision (92.8%), recall (91.2%), mAP50 (95.4%), and F1-score (92.0%). Ablation studies further validate the contribution of each module, while visualizations illustrate UPAD-YOLO's enhanced capacity to accurately localize real defect regions, particularly in handling multi-scale targets and intricate structural damages. Moreover, UPAD-YOLO achieves high detection accuracy alongside efficient inference and model compactness, underscoring its strong potential for real-world industrial deployment. This approach not only delivers a more efficient and robust solution for container damage detection but also provides transferable insights for the design of object detection models in broader industrial applications.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the helpful feedback provided by the reviewers and editors.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by the National Natural Science Foundation of China (62176140, 62376139); the Central Guidance on Local Science and Technology Development Fund of Shandong Province (YDZX2022093); the Technology Innovation Program of Yantai City (2023XDRH001, 2023JCYJ004, 2022XDRH014); and the Shandong Provincial Natural Science Foundation (ZR2023MF014). There was no additional external funding received for this study. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
National Natural Science Foundation of China: 62176140, 62376139.
The Central Guidance on Local Science and Technology Development Fund of Shandong Province: YDZX2022093.
The Technology Innovation Program of Yantai City: 2023XDRH001, 2023JCYJ004, 2022XDRH014.
The Shandong Provincial Natural Science Foundation: ZR2023MF014.

Competing Interests

Zhipeng Liang is employed by Yantai Huadong Electronic Technology Co., Ltd. The authors declare that there are no other competing interests.

Author Contributions

- Benshuo Zhang conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, and approved the final draft.
- Dapeng Cheng analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Zhipeng Liang analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Xinhao Li analyzed the data, prepared figures and/or tables, and approved the final draft.
- Feng Zhao performed the experiments, analyzed the data, authored or reviewed drafts of the article, and approved the final draft.
- Zhiyong An performed the computation work, authored or reviewed drafts of the article, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The Container Damage Dataset is available at Kaggle: Zijie. (2025). Container Damage Dataset [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/12436314.

The UPAD-YOLO Source Code Repository is available at Zenodo: zbs-spec. (2025). zbs-spec/UPAD-YOLO: UPAD-YOLO (v1.0.0). Zenodo. https://doi.org/10.5281/zenodo.15860658.

REFERENCES

Bahrami Z, Zhang R, Wang T, Liu Z. 2022. An end-to-end framework for shipping container corrosion defect inspection. *IEEE Transactions on Instrumentation and Measurement* **71**:1–14 DOI 10.1109/tim.2022.3204091.

Bandong S, Nazaruddin YY, Joelianto E. 2021. Container detection system using CNN-based object detectors. In: *Proceedings of the 2021 International Conference on Instrumentation, Control, and Automation (ICA)*. Piscataway: IEEE, 106–111.

Chen J, Dong C, Wan Y. 2024. Enhancing container damage detection with improved YOLOv5 model: integrating swin transformer. In: *Proceedings of the 2024 International Conference on Intelligent Computing (ICIC), Poster Papers*. Poster paper.

Chen P, Gao L, Shi X, Allen K, Yang L. 2019. Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss. *Computerized Medical Imaging and Graphics* **75(9)**:84–92 DOI 10.1016/j.compmedimag.2019.06.002.

Defect C. 2024. Defect container detection dataset. *Roboflow Universe. Available at https://universe.roboflow.com/container-defect/defect-container-detection* (accessed 6 November 2025).

Enikeev M, Gubaydullin I, Maleeva M. 2017. Analysis of corrosion process development on metals by means of computer vision. *Engineering Journal* **21(4)**:183–192 DOI 10.4186/ej.2017.21.4.183.

Fianna. 2024. Containers damage detect v2 dataset. *Roboflow Universe. Available at https://universe.roboflow.com/fianna-0apa5/containers-damage-detect-v2* (accessed 6 November 2025).

Fouseki K. 2023. US shipping in the global economy and the economic impact of the COVID-19. Master's thesis. University of Piraeus, Athens, Greece.

Girshick R. 2015. Fast R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Piscataway: IEEE, 1440–1448.

Grzelakowski A. 2019. Global container shipping market development and its impact on mega logistics system. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation* **13(3)**:529–535 DOI 10.12716/1001.13.03.06.

Haralambides HE. 2019. Gigantism in container shipping, ports and global logistics: a time-lapse into the future. *Maritime Economics & Logistics* **21(1)**:1–60 DOI 10.1057/s41278-018-00116-0.

Huang X, Zhu J, Huo Y. 2024. SSA-YOLO: an improved YOLO for hot-rolled strip steel surface defect detection. *IEEE Transactions on Instrumentation and Measurement* **73(10)**:1–17 DOI 10.1109/tim.2024.3488136.

İmamoğlu ZE, Tuğlular T, Baştanlar Y. 2020. Container damage detection and classification using container images. In: *Proceedings of the 2020 28th Signal Processing and Communications Applications Conference (SIU)*. Piscataway: IEEE.

Jocher G, Stoken A, Borovec J, Changyu L, Hogan A, Diaconu L, Ingham F, Poznanski J, Fang J, Yu L. 2020. ultralytics/YOLOv5: v3.1-bug fixes and performance improvements. Zenodo DOI 10.5281/zenodo.4154370.

Kattainen E. 2019. Object detection for container corner detection. PhD thesis. Tampere University, Tampere.

Khanam R, Hussain M. 2024. YOLOv11: an overview of the key architectural enhancements. ArXiv DOI 10.48550/arXiv.2410.17725.

Lin M, Changming Z, Rigui Z. 2021. Container damage detection method based on yolov4 algorithm. *Journal of Shanghai Maritime University* **42(4)**:114–118 DOI 10.13340/j.jsmu.2021.04.018.

Liu S, Huang D, Wang Y. 2019. Learning spatial fusion for single-shot object detection. ArXiv DOI 10.48550/arXiv.1911.09516.

Liu Y, Jia F, Yu L, Li X, Liu Y, Chen G, Wu G, Yin Y. 2024. Research on container defect detection based on DB-YOLOv8. *Academic Journal of Computing & Information Science* **7(11)**:135–141 DOI 10.25236/ajcis.2024.071118.

Liu Y, Zhang G, Li X. 2025. Hint-DETR: a transfer learning model based on DETR for few-shot defect detection. *IEEE Transactions on Instrumentation and Measurement* **74**:1–11 DOI 10.1109/tim.2025.3568997.

Nguyen H-N, Kam T-Y, Cheng P-Y. 2014. An automatic approach for accurate edge detection of concrete crack utilizing 2D geometric features of crack. *Journal of Signal Processing Systems* **77(3)**:221–240 DOI 10.1007/s11265-013-0813-8.

Nguyen Thi Phuong T, Cho GS, Chatterjee I. 2025. Automating container damage detection with the YOLO-nas deep learning model. *Science Progress* **108(1)**:368504251314084 DOI 10.1177/00368504251314084.

Odiegwu CL. 2022. Effectiveness of containerization on global transport. SSRN DOI 10.2139/ssrn.4189666.

Pham D-L, Chang T-W. 2023. A YOLO-based real-time packaging defect detection system. *Procedia Computer Science* **217(1)**:886–894 DOI 10.1016/j.procs.2022.12.285.

Rodrigue J-P, Notteboom T. 2015. Looking inside the box: evidence from the containerization of commodities and the cold chain. *Maritime Policy & Management* **42(3)**:207–227 DOI 10.1080/03088839.2014.932925.

Roy AM, Bhaduri J. 2023. DenseSPH-YOLOv5: an automated damage detection model based on DenseNet and swin-transformer prediction head-enabled YOLOv5 with attention mechanism. *Advanced Engineering Informatics* **56(12)**:102007 DOI 10.1016/j.aei.2023.102007.

Sami AA, Sakib S, Deb K, Sarker IH. 2023. Improved YOLOv5-based real-time road pavement damage detection in road infrastructure management. *Algorithms* **16(9)**:452 DOI 10.3390/a16090452.

Skender HP, Host A, Nuhanović M. 2016. The role of logistics service providers in international trade. *Business Logistics in Modern Management* **16**:21–37.

Varghese R, Sambath M. 2024. YOLOv8: a novel object detection algorithm with enhanced performance and robustness. In: *Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. Piscataway: IEEE.

Wang M. 2018. Pavement crack detection based on mobile laser scanning data. ArXiv DOI 10.48550/arXiv.1806.02002.

Wang A, Chen H, Liu L, Chen K, Lin Z, Han J, Ding G. 2024. YOLOv10: real-time end-to-end object detection. *Advances in Neural Information Processing Systems* **37**:107984–108011 DOI 10.52202/079017-3429.

Wang Z, Gao J, Zeng Q, Sun Y. 2021. Multitype damage detection of container using CNN based on transfer learning. *Mathematical Problems in Engineering* **2021(1)**:5395494 DOI 10.1155/2021/5395494.

Xu S, Zheng S, Xu W, Xu R, Wang C, Zhang J, Teng X, Li A, Guo L. 2024. HCF-Net: hierarchical context fusion network for infrared small object detection. In: *Proceedings of the 2024 IEEE International Conference on Multimedia and Expo (ICME)*. Piscataway: IEEE.

Yeon JH, Seo YU, Kim SW, Oh SY, Jeong JH, Park JH, Kim S-H, Youn J. 2022. Shipping container load state and accident risk detection techniques based deep learning. *KIPS Transactions on Computer and Communication Systems* **11(11)**:411–418.

Zhao W, Chen F, Huang H, Li D, Cheng W. 2021. A new steel defect detection algorithm based on deep learning. *Computational Intelligence and Neuroscience* **2021(1)**:5592878 DOI 10.1155/2021/5592878.

Zhao Y, Lv W, Xu S, Wei J, Wang G, Dang Q, Liu Y, Chen J. 2024. DETRs beat YOLOs on real-time object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway: IEEE, 16965–16974.

Zhou Q, Chen X, Tang J. 2024. GANs fostering data augmentation for automated surface inspection with adaptive learning bias. *The International Journal of Advanced Manufacturing Technology* **131(3–4)**:1629–1649.