PPFS-YOLO: Physics-Prior Frequency-Spatial Fusion for Robust Container Surface Damage Detection

Jingze Liu and Feng Gao *

Posted Date: 24 March 2026

doi: 10.20944/preprints202603.1880.v1

Keywords: container damage detection; machine vision sensing; non-contact optical inspection; infrastructure inspection; defect detection; frequency-spatial fusion; physics-prior regularization; YOLO; Fourier spectral masking

 doi:10.20944/preprints202603.1880.v1

Article

PPFS-YOLO: Physics-Prior Frequency-Spatial Fusion for Robust Container Surface Damage Detection

Jingze Liu [1] and Feng Gao [2,*]

[1] Faculty of Information Science and Engineering, Ocean University of China, Qingdao 266100, China
[2] College of Computer Science and Technology, Faculty of Information Science and Engineering, Ocean University of China, Qingdao 266100, China
* Correspondence: gaofeng@ouc.edu.cn

Highlights

- Main finding: PPFS-YOLO combines frequency-spatial fusion with Sobel-guided edge-prior regularization in a YOLOv12s detector for container surface damage inspection.
- Main finding: On the held-out benchmark, PPFS-YOLO improves mAP@50 by 12.10 percentage points over YOLO12s and raises Hole AP@50 by 20.93 percentage points with limited parameter overhead.
- Implication: Spectral feature reweighting and lightweight edge-prior supervision provide a practical way to reduce pseudo-texture interference in container damage detection.
- Implication: The same design may be useful for other visual inspection tasks affected by structured background texture, although broader cross-domain validation is still needed.

Abstract

Container surface damage detection remains challenging for vision-based inspection because rust stains, specular reflections, and paint weathering often resemble true damage, while puncture-type defects (Hole) are underrepresented in available data. We present PPFS-YOLO, a YOLOv12s-based detector for non-contact optical inspection that combines frequency-spatial fusion with edge-prior regularization. The Frequency-Spatial Fusion (FSF) module applies a learnable Fourier-domain mask and gated fusion to complement spatial features. The Physics-Informed Module (FIM) predicts edge maps and aligns them with Sobel-derived priors through an additional L_1 loss, encouraging sharper and more consistent damage boundaries. Three FSF–FIM pairs are inserted at the P3, P4, and P4-head stages of YOLOv12s. On a container damage dataset with 7,013 images and three classes (Dent, Hole, Rusty), PPFS-YOLO reaches 64.86% mAP@50, improving over YOLO12s by 12.10 percentage points with only 0.79 M additional parameters. Ablation results show that the structural modules alone provide limited gains, whereas adding the edge-prior loss yields the full improvement, indicating that the prior term is important for stable joint optimization. These results support PPFS-YOLO as a practical machine-vision sensing approach for automated container inspection.

Keywords: container damage detection; machine vision sensing; non-contact optical inspection; infrastructure inspection; defect detection; frequency-spatial fusion; physics-prior regularization; YOLO; Fourier spectral masking

1. Introduction

Shipping containers constitute the backbone of global freight logistics, with over 800 million TEU (twenty-foot equivalent units) transported annually [1]. Surface damage—including dents from handling impact, puncture holes, and corrosion—can compromise structural integrity, lead to cargo loss, and pose safety hazards at ports. Current inspection practices rely heavily on manual

 doi:10.20944/preprints202603.1880.v1

visual assessment conducted under time pressure, resulting in inconsistent accuracy and limited throughput [2]. Automated visual inspection powered by deep learning offers a scalable alternative, yet a deployable camera-based sensing pipeline must separate safety-critical damage cues from nuisance appearance variations under uncontrolled field conditions.

Recent sensor-centered AI studies further suggest that robust performance depends not only on stronger backbones, but also on task-aware sensing and explicit analysis of failure patterns. Uncertainty-aware view selection can improve 3D scene acquisition efficiency, and systematic analysis of where models fail can reveal error modes that remain hidden under aggregate metrics [3,4]. These observations motivate a container inspection approach that injects domain priors into the detection pipeline rather than relying solely on larger generic detectors.

Deep learning-based object detection, particularly the YOLO family of single-stage detectors [5–11], has been widely adopted for industrial surface defect detection tasks such as steel strip inspection [12–14], PCB defect recognition [15,16], wind turbine blade assessment [17], and boiler inner-wall detection [18]. These methods achieve real-time inference while maintaining competitive detection accuracy. Meanwhile, Transformer-based detectors such as RT-DETR [19] and DINO [20] have demonstrated the potential of attention mechanisms for detection. However, applying these detectors directly to container damage detection presents two domain-specific challenges:

1. **Pseudo-texture interference.** Container surfaces exhibit complex visual patterns—rust stains, paint peeling, specular reflections, and embossed logos—that share low-level feature characteristics with genuine damage. Purely spatial-domain convolutions struggle to disentangle these confounding textures from structural defects, leading to elevated false-positive rates. From a signal-processing perspective, pseudo-textures occupy characteristic frequency bands [21] that overlap with, but are distinct from, genuine damage signatures; this distinction is invisible to standard spatial convolutions.
2. **Minority-class instability.** Puncture-type defects (Hole) are inherently rare in service and in available datasets (approximately 12% of annotated instances), causing detectors to underrepresent this safety-critical category during training. Approaches such as focal loss [5], seesaw loss [22], and hard example mining [23] partially alleviate class imbalance but do not leverage the distinctive physical signatures of hole-type damage.

Recent work has explored frequency-domain analysis for visual recognition and sensing. Fast Fourier Convolution [24] demonstrated the effectiveness of spectral-domain operations with global receptive fields. FcaNet [25] generalized channel attention to multi-spectral representations, proving that global average pooling is a special case of DCT-domain decomposition. In camouflaged object detection, frequency-aware methods [26,27] exploit the complementarity of spatial and spectral features to separate objects from confounding backgrounds—a scenario directly analogous to the pseudo-texture problem in container inspection. In aerial and multimodal sensing, spatial-frequency feature fusion has been applied to oriented object detection [28] and multimodal detection [29]. In the wood panel defect domain, FDADNet [30] employs frequency-domain transformation with adaptive downsampling. Early work on tile defect detection also combined spatial-frequency enhancement with region growing [31]. To the best of our knowledge, prior work has not reported the combination of frequency-spatial fusion and explicit physics-prior regularization within an end-to-end YOLO detection framework.

Physics-informed neural networks (PINNs), first formalized by Raissi et al. [32], encode physical laws as soft constraints within neural network training and have been effective in scientific computing [33]. In defect analysis, physics-informed approaches have been applied to crack monitoring via guided wave signals [34,35], alternating current field measurement [36], magnetic flux leakage estimation [37], and material internal structure analysis [38]. However, these approaches target reconstruction or measurement tasks rather than visual object detection. The concept of encoding domain-specific physical knowledge—such as edge continuity and boundary sharpness [39,40]—as differentiable loss terms within a detection pipeline remains largely unexplored.

 doi:10.20944/preprints202603.1880.v1

In this paper, we develop PPFS-YOLO (**P**hysics-**P**rior **F**requency-**S**patial YOLO), a framework that addresses both pseudo-texture interference and edge-prior-guided feature learning for container surface damage detection. Our key contributions are:

1. We design the **Frequency-Spatial Fusion (FSF)** module, which performs learnable spectral masking in the 2D Fourier domain followed by gated spatial-frequency feature fusion, enabling the network to selectively suppress pseudo-texture frequency components while preserving damage-related signals.
2. We propose the **Physics-Informed Module (FIM)**, which encodes Sobel-derived edge priors as a differentiable L_1 loss ($\mathcal{L}_{\text{phy}}$) and applies edge-guided residual refinement, steering the network toward physically plausible damage representations.
3. We provide an ablation study that separates the effects of FSF, FIM, and the edge-prior loss. The results show that FSF+FIM without $\mathcal{L}_{\text{phy}}$ yields only +0.83 pp mAP@50, whereas enabling $\mathcal{L}_{\text{phy}}$ raises the gain to +12.10 pp.
4. Among the evaluated detectors, PPFS-YOLO achieves the best performance on this dataset (64.86% mAP@50), with large gains on the minority Hole class (+20.93 pp AP@50) and in overall precision (+12.96 pp).

The remainder of this paper is organized as follows: Section 2 reviews related work on YOLO-based defect detection, frequency-domain feature analysis, and prior-guided defect modeling. Section 3 presents the PPFS-YOLO architecture and its two core modules with detailed mathematical derivations. Section 4 describes the experimental setup and results. Section 5 provides an in-depth analysis of the findings. Section 6 concludes the paper.

2. Related Work

2.1. Evolution of Real-Time Object Detectors

Real-time object detection has been driven by two parallel paradigms: CNN-based single-stage detectors and Transformer-based end-to-end detectors. The YOLO (You Only Look Once) family [5] epitomizes the former, evolving through successive architectural innovations. YOLOv7 [7] introduced trainable bag-of-freebies with extended efficient layer aggregation; YOLOv8 [6] adopted an anchor-free decoupled head with task-aligned assignment; YOLOv9 [8] proposed programmable gradient information for enhanced feature learning; YOLOv10 [9] eliminated non-maximum suppression via consistent dual assignments; YOLO11 [10] further optimized the backbone with C3k2 modules; and the recent YOLOv12 [11] adopted attention-centric blocks (A2C2f) that harness the representation capacity of attention mechanisms while maintaining real-time speed.

On the Transformer side, DINO [20] introduced improved denoising anchor boxes for DETR-based detection, and RT-DETR [19] extended this paradigm to real-time detection with a hybrid encoder design. Complementary to architecture innovations, loss function design has been crucial: focal loss [5] addressed foreground-background imbalance; generalized focal loss [41] unified quality estimation with classification; task-aligned one-stage detection (TOOD) [42] jointly optimized classification and localization; and balanced learning strategies [43,44] improved training sample selection.

Feature pyramid networks (FPN) [45] and path aggregation networks (PANet) [46] established the multi-scale feature fusion paradigm, later refined by BiFPN [47] with learnable weighted fusion. Channel attention (SE-Net [48]), spatial attention (CBAM [49]), and their combinations have become standard components. However, all these attention mechanisms operate in the spatial domain and do not exploit frequency-domain representations—a gap our FSF module addresses.

2.2. YOLO-Based Industrial Defect Detection

YOLO variants have been extensively adapted for industrial defect detection across diverse domains. **Steel surface inspection:** MSFT-YOLO [12] integrated Transformer modules into YOLOv5 for defect detection on steel surfaces. An improved YOLOv4 [13] targeted steel strip defects using enhanced backbone features. Multi-scale feature fusion with attention residual blocks [14] advanced

 doi:10.20944/preprints202603.1880.v1

hot-rolled steel detection. Mixed receptive field augmentation [50] and improved YOLOX [51] have also been applied. YOLOv8-MGVS [52] and ASD-YOLO [53] further pushed the performance boundary with multi-module collaborative optimization. **PCB and electronics:** YOLOv4-MN3 [15] combined MobileNetv3 with YOLOv4 for PCB defects; PCB-YOLO [16] enhanced YOLOv5 specifically for circuit board inspection. **Other domains:** Wind turbine blade defects [17], boiler inner-wall damage [18], industrial parts [54], particleboard surfaces [55], magnetic tiles [56], bearing surfaces [57], and fabric defects [58] have all been addressed by improved YOLO detectors. MAS-YOLO [59] improved YOLOv12 for PCB defects using median-enhanced attention. A recent transmission line defect detector [60] combined BiFPN with channel-position collaborative attention on YOLOv12.

For container-specific damage, YOLO-NAS [2] automated detection but without addressing the pseudo-texture false-positive problem. A systematic survey [61] and a deep learning survey on surface defects [62] have confirmed the growing complexity of industrial inspection scenarios; yet frequency-domain exploitation and physics-prior integration remain absent from existing YOLO-based defect detectors—a dual gap our PPFS-YOLO addresses.

2.3. Frequency-Domain Feature Analysis in Visual Recognition

Frequency-domain representations provide complementary information to spatial features, with particular advantages for distinguishing textural patterns from structural signals [21]. From the foundational Parseval's theorem [63], the energy content preserved across spatial and spectral domains ensures that frequency-domain filtering does not inherently discard information but rather reorganizes it.

General vision: Fast Fourier Convolution (FFC) [24] introduced spectral convolutions with global receptive fields for image generation, enabling non-local feature interactions without the quadratic cost of self-attention. FcaNet [25] generalized channel attention via discrete cosine transform decomposition, mathematically proving that global average pooling is a special case of frequency-domain feature compression.

Camouflaged object detection: This sub-field presents a problem highly analogous to pseudo-texture confusion. Zhong et al. [26] proposed a frequency enhancement module (FEM) with offline DCT followed by learnable enhancement and high-order relation modules for rich feature fusion. FBNet [27] designed frequency-aware context aggregation (FACA) to suppress confounding high-frequency textures and adaptive frequency attention (AFA) to enhance discriminative frequency components.

Detection tasks across sensing modalities: SFFD [28] developed a layer-wise frequency-domain analysis (L-FDA) module for oriented object detection in aerial imagery, demonstrating that frequency features capture rotation-invariant signatures. FDTNet [64] employed dual-stream Transformers for frequency-aware prohibited object detection in X-ray images. For multimodal aerial sensing, an adaptive frequency-domain gate [29] dynamically learns the dependence on frequency-filtered features.

Industrial defect detection: FDADNet [30] applied multi-axis frequency-domain weighted information representation for wood panel defect detection. A spatial-frequency enhancement method [31] combined Gabor filtering with region growing for tile defect detection.

Our FSF module differs from prior frequency-domain approaches in three key aspects: (1) it employs a *fully learnable* 2D spectral mask $\boldsymbol{M}_f(u, v)$ with per-channel scaling rather than fixed frequency filters or offline DCT; (2) it uses bilinear interpolation from a compact base resolution (40×21), enabling resolution-agnostic deployment; and (3) it is tightly coupled with a physics-prior module via shared $\mathcal{L}_{\text{phy}}$ supervision, creating a synergistic effect that exceeds the sum of individual components.

2.4. Physics-Informed and Prior-Guided Defect Analysis

Physics-informed neural networks (PINNs), first formalized by Raissi et al. [32] for solving partial differential equations, encode physical laws as soft constraints within neural network training. A comprehensive survey by Cuomo et al. [33] categorized PINNs into vanilla, physics-constrained

 doi:10.20944/preprints202603.1880.v1

(PCNN), variational (hp-VPINN), and conservative (CPINN) variants, noting that most advances focus on customizing activation functions, gradient optimization, and loss structures.

In defect analysis and non-destructive evaluation:

- **Crack monitoring:** Chen et al. [34] employed a physics-informed LSTM for real-time fatigue crack quantification from harmonic parameters, achieving 0.498 mm RMSE—significantly outperforming pure LSTM (3.205 mm) and Paris' Law (3.641 mm).
- **Guided wave testing:** GuwNet [35] integrated ultrasonic guided wave physics into a deep neural network for microcrack quantification, reducing quantification errors by over 80% compared to non-physics methods.
- **Magnetic flux leakage:** DfedResNet [37] proposed a physics-informed doubly-fed cross-residual network, achieving 1–2 orders of magnitude improvement in defect depth estimation.
- **Electromagnetic testing:** An end-to-end PINN [36] for rotating alternating current field measurement achieved 0.9982 mAP for defect identification with 3D reconstruction.
- **Material analysis:** Zhang et al. [38] presented a general PINN framework for identifying internal voids and inclusions across linear elastic, hyperelastic, and plastic material models.

These studies show that differentiable domain constraints can improve robustness and sample efficiency in defect-related tasks. However, most of them target *reconstruction and quantification tasks* (e.g., estimating crack depth from sensor signals) rather than visual object detection.

Our FIM takes inspiration from this line of work but adopts a lighter formulation suitable for detection: it regularizes feature responses using Sobel-derived edge priors, rather than a full governing-equation model, within an end-to-end detector. To the best of our knowledge, prior YOLO-based defect detectors have not combined this type of edge-prior regularization with frequency-spatial fusion in a single architecture. Table 1 summarizes the positioning of PPFS-YOLO relative to representative prior work.

Table 1. Comparison of PPFS-YOLO with representative prior methods across three design dimensions. ✔ = supported; ✗ = not supported.

Method	Domain	Freq. Fusion	Physics Prior	End-to-End Det.
FcaNet [25]	General	✔	✗	✗
FFC [24]	General	✔	✗	✗
FreqCOD [26]	Camouflage	✔	✗	✗
SFFD [28]	Remote Sens.	✔	✗	✔
FDADNet [30]	Wood Defect	✔	✗	✔
GuwNet [35]	NDT	✗	✔	✗
DfedResNet [37]	MFL	✗	✔	✗
YOLO-NAS [2]	Container	✗	✗	✔
MAS-YOLO [59]	PCB	✗	✗	✔
PPFS-YOLO (Ours)	Container	✔	✔	✔

3. Method

3.1. Overall Architecture

PPFS-YOLO is built upon the YOLOv12s architecture [11] and augments the detection pipeline with two plug-in modules: Frequency-Spatial Fusion (FSF) and Physics-Informed Module (FIM). As illustrated in Figure 1, the network is organized into three columns—Backbone, Top-Down Neck, and Bottom-Up Head—with 27 indexed layers (0–26) plus a final Detect head.

 doi:10.20944/preprints202603.1880.v1

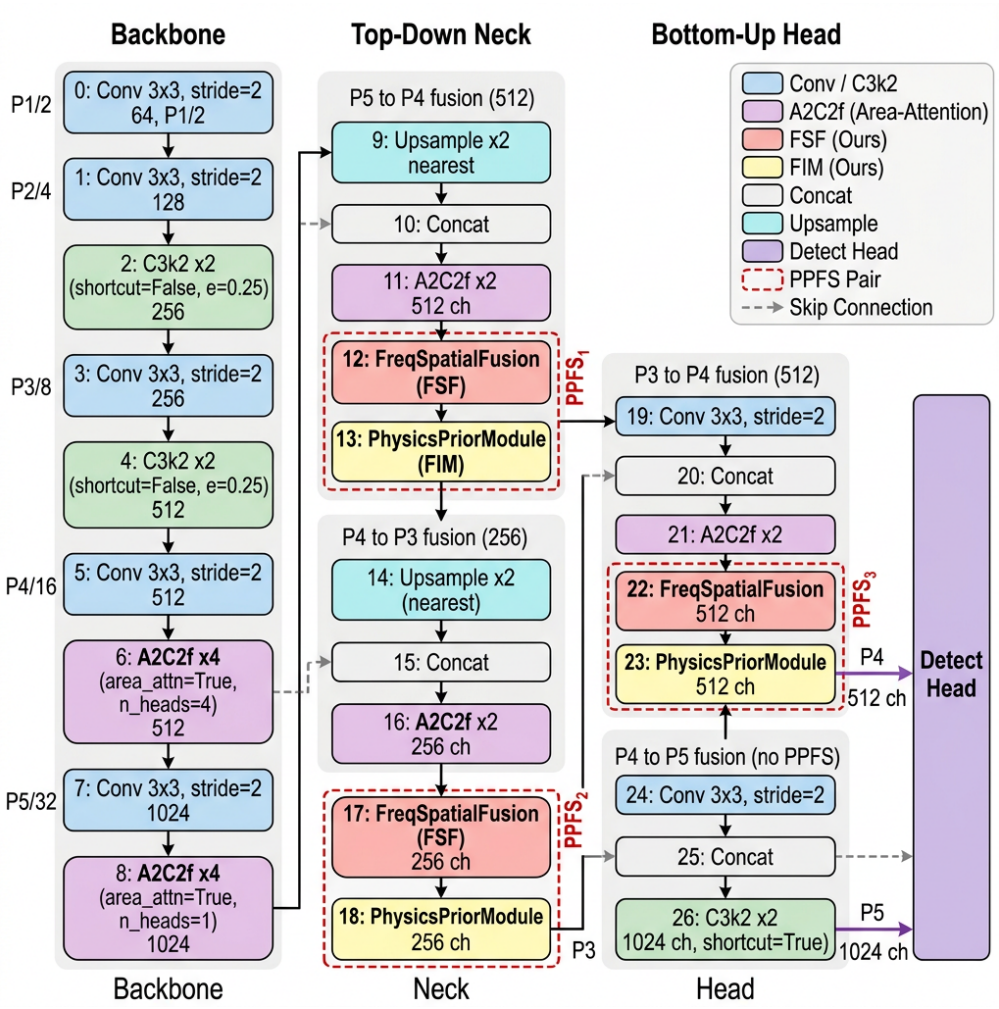


Figure 1. Overall architecture of PPFS-YOLO. The network is organized into three columns: the Backbone (layers 0–8), the Top-Down Neck (layers 9–18), and the Bottom-Up Head (layers 19–27). Color-coded blocks indicate module types: blue = Conv/C3k2, pink = A2C2f, red = FSF, yellow = FIM, white = Concat, teal = Upsample, purple = Detect. Three PPFS pairs (dashed groups $PPFS_1$–$PPFS_3$) are inserted at layers 12–13 (P4 neck), 17–18 (P3 neck), and 22–23 (P4 head). Skip connections from backbone layers 4, 6, and 8 feed into the neck and head via Concat nodes. The Detect head outputs predictions at P3, P4, and P5 scales.

The backbone follows the YOLOv12s design with Conv–C3k2–A2C2f blocks, producing feature maps at three scales: P3 ($H/8 \times W/8$, 256 channels), P4 ($H/16 \times W/16$, 512 channels), and P5 ($H/32 \times W/32$, 1024 channels). In the neck, a top-down and bottom-up feature pyramid network [45,46] fuses multi-scale features through concatenation and A2C2f blocks. Table 2 details the complete 27-layer architecture.

Three FSF–FIM pairs are inserted at strategically chosen positions:

- **P4 Neck (Layers 12–13):** After the first A2C2f block in the top-down path (512 channels).
- **P3 Neck (Layers 17–18):** After the second A2C2f block in the top-down path (256 channels).
- **P4 Head (Layers 22–23):** After the A2C2f block in the bottom-up path (512 channels).

This placement ensures that frequency-spatial fusion and physics-prior regularization are applied to both medium-scale (P4) and fine-scale (P3) features, where pseudo-texture interference and small defect details are most prominent. The total parameter overhead is only +0.79 M (from 9.23 M to 10.02 M), and the computational cost increases by +1.7 GFLOPs (from 10.8 to 12.5 GFLOPs). Table 3 provides a per-module breakdown.

 doi:10.20944/preprints202603.1880.v1

Table 2. PPFS-YOLO layer-by-layer architecture. Inserted FSF and FIM modules are highlighted with background. "−1" denotes the preceding layer; bracketed indices denote concatenation sources.

Layer	Module	From	Channels	Role
Backbone				
0	Conv 3×3 s2	image	64	stem
1	Conv 3×3 s2	−1	128	down
2	C3k2 ($n=2$)	−1	256	feature
3	Conv 3×3 s2	−1	256	down
4	C3k2 ($n=2$)	−1	512	feature
5	Conv 3×3 s2	−1	512	down
6	A2C2f ($n=2$)	−1	512	attn
7	Conv 3×3 s2	−1	1024	down
8	A2C2f ($n=2$)	−1	1024	attn
Neck (top-down)				
9	Upsample 2×	−1	1024	up
10	Concat	[−1, 6]	1536	fuse
11	A2C2f ($n=2$)	−1	512	refine
12	FreqSpatialFusion	−1	512	**FSF (P4)**
13	PhysicsPriorModule	−1	512	**FIM (P4)**
14	Upsample 2×	−1	512	up
15	Concat	[−1, 4]	768	fuse
16	A2C2f ($n=2$)	−1	256	refine
17	FreqSpatialFusion	−1	256	**FSF (P3)**
18	PhysicsPriorModule	−1	256	**FIM (P3)**
Head (bottom-up)				
19	Conv 3×3 s2	−1	256	down
20	Concat	[−1, 13]	768	fuse
21	A2C2f ($n=2$)	−1	512	refine
22	FreqSpatialFusion	−1	512	**FSF (P4-head)**
23	PhysicsPriorModule	−1	512	**FIM (P4-head)**
24	Conv 3×3 s2	−1	512	down
25	Concat	[−1, 8]	1536	fuse
26	A2C2f ($n=2$)	−1	1024	refine
27	Detect	[18, 23, 26]	—	output

Table 3. Parameter and FLOP breakdown of PPFS modules (per instance). C denotes the input channel count; values are computed at 640×640 input resolution.

Module	C	Params (K)	GFLOPs	Component Details
FSF (P3)	256	33.5	0.11	mask (40×21), channel scale, gate conv
FSF (P4, P4-head)	512	132.4	0.10	mask (40×21), channel scale, gate conv
FIM (P3)	256	71.4	0.17	edge predictor, DW-PW refine ×2
FIM (P4, P4-head)	512	283.9	0.51	edge predictor, DW-PW refine ×2
Total (3 pairs)	—	**790**	**1.70**	—

3.2. Frequency-Spatial Fusion Module

The FSF module addresses the pseudo-texture problem by performing learnable spectral filtering in the Fourier domain and fusing the result with spatial features through a gated mechanism. As shown in Figure 2, the module consists of a frequency path (top) and a spatial identity path (bottom), joined by a gated fusion block.

 doi:10.20944/preprints202603.1880.v1

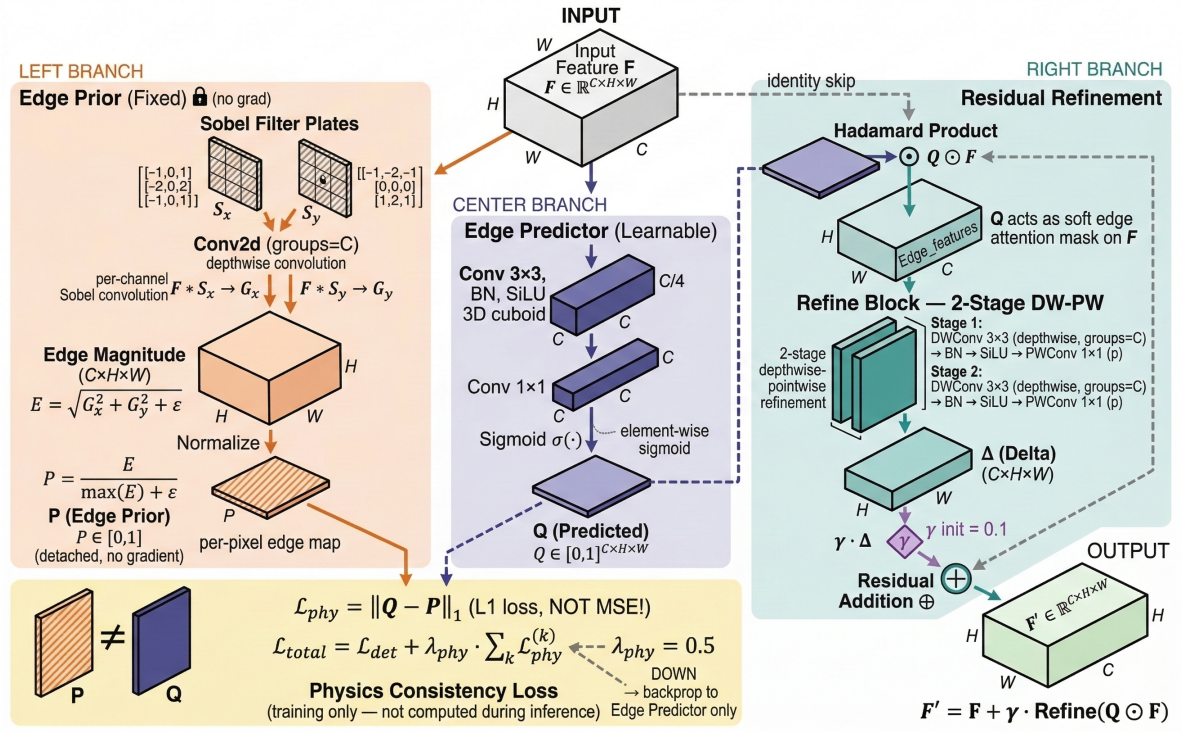


Figure 2. Architecture of the Frequency-Spatial Fusion (FSF) module. The input feature $\boldsymbol{F}_s$ is transformed via FFT2 and decomposed into amplitude $\boldsymbol{A}$ and phase $\boldsymbol{\Phi}$. A learnable 2D frequency mask $\boldsymbol{M}_f$ (base size 40×21, bilinearly interpolated) together with a per-channel scale $\boldsymbol{S}_c$ reweights the amplitude spectrum to produce the masked amplitude $\tilde{\boldsymbol{A}}$. The reconstructed signal is obtained via IFFT2, yielding $\boldsymbol{F}_{\text{freq}}$. The spatial path passes $\boldsymbol{F}_s$ through as identity. A gated fusion mechanism concatenates $\boldsymbol{F}_s$ and $\boldsymbol{F}_{\text{freq}}$, applies $\text{Conv}_{1\times1}$–BN–SiLU–$\text{Conv}_{1\times1}$–Sigmoid to produce $\boldsymbol{\alpha}$ (bias-initialized to $+1.0$, giving $\alpha_0 \approx 0.73$), and outputs $\boldsymbol{\alpha} \odot \boldsymbol{F}_s + (1-\boldsymbol{\alpha}) \odot \boldsymbol{F}_{\text{freq}}$.

Given an input spatial feature map $\boldsymbol{F}_s \in \mathbb{R}^{C\times H\times W}$, the FSF module operates as follows.

3.2.1. Theoretical Foundation: Parseval's Theorem

The design rationale for FSF rests on *Parseval's theorem* [21,63], which guarantees energy equivalence between the spatial and frequency domains. For a discrete 2D signal $f[m,n]$ of size $H\times W$ and its DFT $\hat{F}[u,v]$:

$$\sum_{m=0}^{H-1}\sum_{n=0}^{W-1} |f[m,n]|^2 = \frac{1}{HW}\sum_{u=0}^{H-1}\sum_{v=0}^{W-1} |\hat{F}[u,v]|^2. \tag{1}$$

Proof sketch. By definition, $\hat{F}[u,v] = \sum_{m,n} f[m,n]\, e^{-j2\pi(um/H+vn/W)}$. Computing $\sum_{u,v} |\hat{F}[u,v]|^2$:

$$\begin{aligned}
\sum_{u,v} |\hat{F}|^2 &= \sum_{u,v} \hat{F}[u,v]\, \hat{F}^*[u,v] \\
&= \sum_{u,v}\sum_{m,n}\sum_{m',n'} f[m,n]\, f^*[m',n']\, e^{-j2\pi u(m-m')/H}\, e^{-j2\pi v(n-n')/W} \\
&= \sum_{m,n}\sum_{m',n'} f[m,n]\, f^*[m',n'] \underbrace{\sum_{u} e^{-j2\pi u(m-m')/H}}_{H\,\delta_{m,m'}} \underbrace{\sum_{v} e^{-j2\pi v(n-n')/W}}_{W\,\delta_{n,n'}} \\
&= HW \sum_{m,n} |f[m,n]|^2.
\end{aligned} \tag{2}$$

Equation (1) follows directly. Importantly, this means that modulating $|\hat{F}[u,v]|$ by a mask $M_f(u,v)$ in the frequency domain is equivalent to an energy redistribution in the spatial domain. Specifically, if $M_f(u,v) \approx 0$ for frequency bands associated with pseudo-textures, the corresponding spatial energy is suppressed. Unlike spatial convolutions with local receptive fields, this spectral masking provides

 doi:10.20944/preprints202603.1880.v1

global spatial context at each frequency component, making it naturally suited for suppressing spatially repeated pseudo-texture patterns.

3.2.2. Forward Computation

Step 1: Frequency-Domain Transform. The 2D FFT is applied channel-wise:

$$\hat{\boldsymbol{F}}[c,u,v] = \sum_{m=0}^{H-1}\sum_{n=0}^{W-1} \boldsymbol{F}_s[c,m,n]\, e^{-j2\pi\left(\frac{um}{H}+\frac{vn}{W}\right)}, \quad \hat{\boldsymbol{F}} \in \mathbb{C}^{C\times H\times W}. \tag{3}$$

The amplitude and phase components are separated as $\boldsymbol{A} = |\hat{\boldsymbol{F}}|$ and $\boldsymbol{\Phi} = \angle(\hat{\boldsymbol{F}})$.

Step 2: Learnable Spectral Masking. A 2D learnable frequency mask $\boldsymbol{M}_f \in \mathbb{R}^{H_b\times W_b}$ is maintained at a compact base resolution ($H_b = 40$, $W_b = 21$) and bilinearly interpolated to match the input spatial dimensions. This low-rank parameterization reduces the mask parameters from $H\times W$ to $H_b\times W_b$ (e.g., from $80\times 40 = 3200$ to $40\times 21 = 840$ for P3), providing implicit low-pass regularization on the mask itself. A per-channel scaling vector $\boldsymbol{S}_c \in \mathbb{R}^C$ modulates the mask across channels. The masked amplitude is:

$$\tilde{\boldsymbol{A}}[c,u,v] = S_c[c]\cdot \mathrm{Interp}(\boldsymbol{M}_f)[u,v]\cdot \boldsymbol{A}[c,u,v], \tag{4}$$

where $\mathrm{Interp}(\cdot)$ denotes bilinear interpolation from (H_b, W_b) to (H, W).

Step 3: Inverse Transform. The frequency-enhanced feature map is reconstructed via inverse FFT:

$$\boldsymbol{F}_{\text{freq}}[c,m,n] = \frac{1}{HW}\sum_{u=0}^{H-1}\sum_{v=0}^{W-1} \tilde{\boldsymbol{A}}[c,u,v]\, e^{j\boldsymbol{\Phi}[c,u,v]}\, e^{j2\pi\left(\frac{um}{H}+\frac{vn}{W}\right)}. \tag{5}$$

Step 4: Gated Fusion. The spatial and frequency features are fused through a learnable gating mechanism:

$$\boldsymbol{\alpha} = \sigma\Big(\mathrm{Conv}_{1\times 1}\Big([\boldsymbol{F}_s;\ \boldsymbol{F}_{\text{freq}}]\Big) + b_{\text{gate}}\Big) \in \mathbb{R}^{C\times H\times W}, \tag{6}$$

$$\boldsymbol{F}^* = \boldsymbol{\alpha}\odot \boldsymbol{F}_s + (1-\boldsymbol{\alpha})\odot \boldsymbol{F}_{\text{freq}}, \tag{7}$$

where $[\cdot;\cdot]$ denotes channel-wise concatenation, $\mathrm{Conv}_{1\times 1}$ reduces $2C$ channels to C (via an intermediate $C/4$ bottleneck), and $\sigma(\cdot)$ is the sigmoid function.

3.2.3. Gate Initialization Analysis

The gate bias is initialized to $b_{\text{gate}} = +1.0$. Since at initialization the convolution weights yield approximately zero-mean outputs, the initial gate value is:

$$\boldsymbol{\alpha}_0 \approx \sigma(1.0) = \frac{1}{1+e^{-1}} \approx 0.731. \tag{8}$$

This ensures that ≈73% of the initial output comes from the spatial pathway, preserving the pretrained backbone representations during early training and allowing the frequency pathway to gradually increase its contribution as the mask $\boldsymbol{M}_f$ is optimized. The gradient of the gate with respect to its input $z = \mathrm{Conv}(\cdot) + b_{\text{gate}}$ is:

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)(1-\sigma(z)) \approx 0.731\times 0.269 \approx 0.197, \tag{9}$$

which lies in the high-sensitivity region of the sigmoid, ensuring that gradients flow effectively to update the gating parameters.

 doi:10.20944/preprints202603.1880.v1

Algorithm 1 FSF Module Forward Pass

Require: Input feature $\boldsymbol{F}_s \in \mathbb{R}^{C\times H\times W}$; learnable mask $\boldsymbol{M}_f \in \mathbb{R}^{H_b\times W_b}$; channel scale $\boldsymbol{S}_c \in \mathbb{R}^{C}$; gate bias b_{gate}

Ensure: Fused feature $\boldsymbol{F}^* \in \mathbb{R}^{C\times H\times W}$

$\hat{\boldsymbol{F}} \leftarrow \text{FFT2}(\boldsymbol{F}_s)$ ▷ 2D FFT

$\boldsymbol{A},\ \boldsymbol{\Phi} \leftarrow |\hat{\boldsymbol{F}}|,\ \angle(\hat{\boldsymbol{F}})$ ▷ Amplitude & Phase

$\boldsymbol{M} \leftarrow \text{BilinearInterp}(\boldsymbol{M}_f,\ H,\ W)$ ▷ Upsample mask

$\tilde{\boldsymbol{A}}[c,u,v] \leftarrow S_c[c]\cdot \boldsymbol{M}[u,v]\cdot \boldsymbol{A}[c,u,v]$ ▷ Spectral masking

$\hat{\boldsymbol{F}}_{\text{masked}} \leftarrow \tilde{\boldsymbol{A}}\cdot e^{j\boldsymbol{\Phi}}$ ▷ Reconstruct complex spectrum

$\boldsymbol{F}_{\text{freq}} \leftarrow \text{Re}\big(\text{IFFT2}(\hat{\boldsymbol{F}}_{\text{masked}})\big)$ ▷ Inverse FFT

$\boldsymbol{\alpha} \leftarrow \sigma\Big(\text{Conv}_{1\times1}\Big([\boldsymbol{F}_s;\ \boldsymbol{F}_{\text{freq}}]\Big) + b_{\text{gate}}\Big)$ ▷ Gated fusion

$\boldsymbol{F}^* \leftarrow \boldsymbol{\alpha}\odot \boldsymbol{F}_s + (1-\boldsymbol{\alpha})\odot \boldsymbol{F}_{\text{freq}}$

return $\boldsymbol{F}^*$

3.3. Physics-Informed Module

The FIM module encodes the physical prior that genuine structural damage exhibits sharp, continuous edge boundaries, whereas pseudo-textures produce diffuse or irregular edge responses. In this paper, "physics prior" refers to a structural prior derived from boundary sharpness and continuity rather than an explicit PDE or constitutive law. As shown in Figure 3, the module comprises three parallel branches—Edge Prior (left), Edge Predictor (center), and Residual Refinement (right)—whose internal data flows and loss connections are annotated in the diagram.

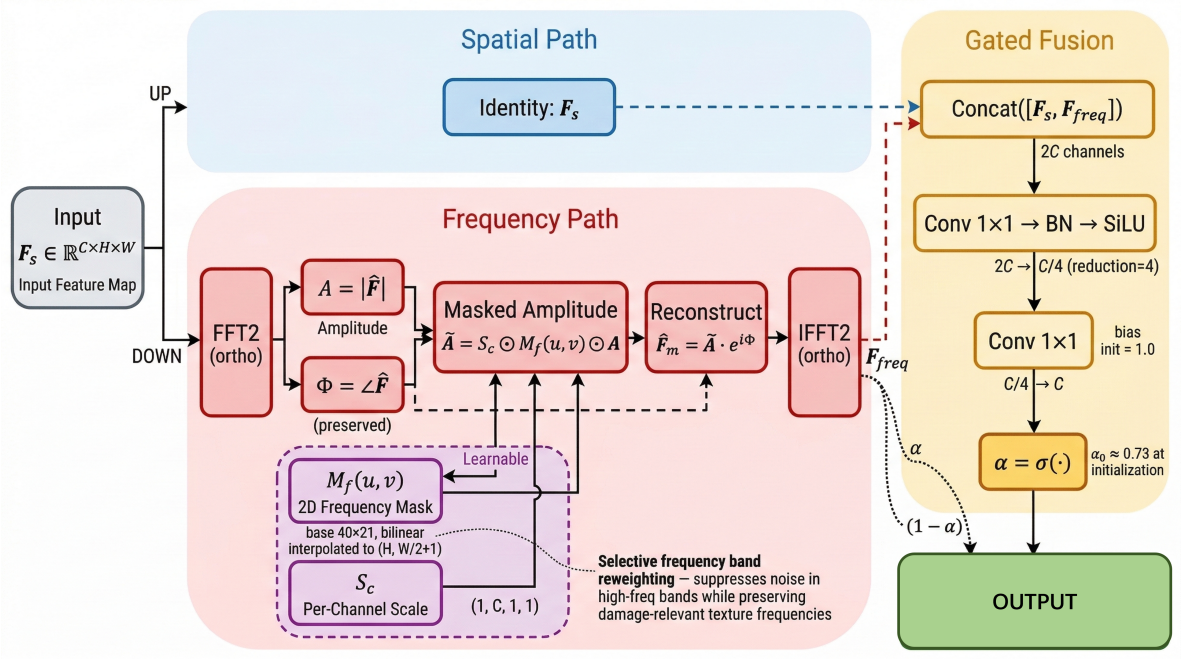


Figure 3. Architecture of the Physics-Informed Module (FIM). The module comprises three branches. **Left – Edge Prior**: fixed Sobel filters (no gradient) applied as depthwise convolutions produce gradient magnitudes that are normalized to $[0,1]$, yielding the edge prior map $\boldsymbol{P}$. **Center – Edge Predictor**: a learnable $\text{Conv}_{3\times3}$–$\text{Conv}_{1\times1}$–Sigmoid pathway predicts the edge map $\boldsymbol{Q}$ from the input feature $\boldsymbol{F}$. **Right – Residual Refinement**: the Hadamard product $\boldsymbol{Q}\odot\boldsymbol{F}$ is passed through a two-stage DW–PW refine block producing $\boldsymbol{\Delta}$, which is added back as $\boldsymbol{F}' = \boldsymbol{F} + \gamma\cdot\boldsymbol{\Delta}$. The physics-prior loss $\mathcal{L}_{\text{phy}} = \|\boldsymbol{Q} - \boldsymbol{P}\|_1$ supervises the edge predictor, and the total loss $\mathcal{L}_{\text{total}}$ combines detection and physics terms.

Given an input feature $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$:

 doi:10.20944/preprints202603.1880.v1

3.3.1. Edge Prior via Sobel Operators

The Sobel operator [40] approximates the image gradient using separable 3×3 kernels. For horizontal and vertical gradients:

$$\boldsymbol{K}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \qquad \boldsymbol{K}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}. \tag{10}$$

These are applied as depthwise convolutions (shared across C channels, no learnable parameters). The gradient computation is performed on a detached copy $\boldsymbol{F}_{\text{det}} = \text{sg}(\boldsymbol{F})$ (where sg denotes the stop-gradient operator) to prevent the physics constraint from directly altering the backbone features:

$$\boldsymbol{G}_x = \boldsymbol{K}_x * \boldsymbol{F}_{\text{det}}, \quad \boldsymbol{G}_y = \boldsymbol{K}_y * \boldsymbol{F}_{\text{det}}, \quad \boldsymbol{G}_x, \boldsymbol{G}_y \in \mathbb{R}^{C\times H\times W}, \tag{11}$$

where $*$ denotes the convolution operator. The edge magnitude prior is:

$$\boldsymbol{P} = \sigma\Big(\sqrt{\boldsymbol{G}_x^2 + \boldsymbol{G}_y^2 + \epsilon}\Big), \quad \boldsymbol{P} \in [0,1]^{C\times H\times W}, \tag{12}$$

where $\epsilon = 10^{-6}$ prevents numerical instability and $\sigma(\cdot)$ normalizes the magnitude to $[0,1]$.

The physical interpretation is as follows: regions with genuine structural damage (dents, holes) produce strong, coherent gradient responses across multiple feature channels, while pseudo-textures yield spatially diffuse gradients or responses confined to specific channels. This signal structure motivates a channel-wise edge alignment loss.

3.3.2. Learnable Edge Prediction

A lightweight predictor network h_θ estimates a physics-consistent edge map:

$$\boldsymbol{Q} = \sigma(\text{PW}_{C\to C}(\text{SiLU}(\text{BN}(\text{DW}_{3\times3}(\boldsymbol{F}))))), \quad \boldsymbol{Q} \in [0,1]^{C\times H\times W}, \tag{13}$$

where $\text{DW}_{3\times3}$ denotes a depthwise 3×3 convolution (capturing local spatial patterns with $9C$ parameters) and $\text{PW}_{C\to C}$ is a pointwise 1×1 convolution (cross-channel mixing with C^2 parameters).

3.3.3. Physics-Prior Loss and Gradient Analysis

The alignment between the predicted and actual edge maps is enforced via the L_1 loss:

$$\mathcal{L}_{\text{phy}}^{(i)} = \frac{1}{CHW} \sum_{c,m,n} |Q^{(i)}[c,m,n] - P^{(i)}[c,m,n]|, \quad i \in \{1,2,3\}. \tag{14}$$

Gradient flow analysis. The gradient of $\mathcal{L}_{\text{phy}}^{(i)}$ with respect to the predictor parameters θ is:

$$\frac{\partial \mathcal{L}_{\text{phy}}^{(i)}}{\partial \theta} = \frac{1}{CHW} \sum_{c,m,n} \text{sign}\Big(Q_{c,m,n}^{(i)} - P_{c,m,n}^{(i)}\Big) \cdot \frac{\partial Q_{c,m,n}^{(i)}}{\partial \theta}. \tag{15}$$

The L_1 loss is specifically chosen over L_2 because the sign function provides constant-magnitude gradients regardless of the residual size. This prevents the gradient vanishing that occurs with L_2 when $|\boldsymbol{Q} - \boldsymbol{P}| \to 0$, ensuring that the physics alignment signal remains strong throughout training. Furthermore, L_1 is robust to outlier edge responses that may arise from container surface specular reflections.

Why stop-gradient on $\boldsymbol{P}$? The edge prior $\boldsymbol{P}$ is computed from the detached feature $\text{sg}(\boldsymbol{F})$. If gradient were allowed to flow through $\boldsymbol{P}$, the network could trivially minimize $\mathcal{L}_{\text{phy}}$ by making $\boldsymbol{F}$ smooth (zero-gradient features everywhere), which would destroy the representation quality. By detaching

 doi:10.20944/preprints202603.1880.v1

$\boldsymbol{P}$, the physics loss exclusively trains the predictor h_θ to predict edge structure, creating a *knowledge distillation*-like setup where the Sobel operator acts as a fixed "teacher" and h_θ is the "student."

3.3.4. Edge-Guided Residual Refinement

The predicted edge map modulates the input feature, and a two-stage depthwise-pointwise convolutional refinement is applied:

$$\boldsymbol{F}' = \boldsymbol{F} + \gamma \cdot \underbrace{\mathrm{PW}(\mathrm{SiLU}(\mathrm{BN}(\mathrm{DW}(\mathrm{PW}(\mathrm{SiLU}(\mathrm{BN}(\mathrm{DW}(\boldsymbol{Q} \odot \boldsymbol{F}))))))))}_{\mathrm{Refine}(\boldsymbol{Q} \odot \boldsymbol{F})}, \tag{16}$$

where γ is a learnable scalar initialized to $\gamma_0 = 0.1$. The small initial γ_0 is critical: it ensures that the refinement branch produces near-zero modifications at the start of training, preventing the randomly initialized edge predictor from corrupting the feature representations. As training progresses and $\boldsymbol{Q}$ converges toward meaningful edge maps, γ grows to allow stronger edge-guided modulation.

Algorithm 2 FIM Module Forward Pass

Require: Input feature $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$; Sobel kernels $\boldsymbol{K}_x, \boldsymbol{K}_y$; edge predictor h_θ; residual scale γ
Ensure: Refined feature $\boldsymbol{F}' \in \mathbb{R}^{C\times H\times W}$; physics loss $\mathcal{L}_{\mathrm{phy}}$

$\boldsymbol{F}_{\mathrm{det}} \leftarrow \mathrm{sg}(\boldsymbol{F})$ ▷ Stop gradient on feature
$\boldsymbol{G}_x,\ \boldsymbol{G}_y \leftarrow \boldsymbol{K}_x * \boldsymbol{F}_{\mathrm{det}},\ \boldsymbol{K}_y * \boldsymbol{F}_{\mathrm{det}}$ ▷ Sobel depthwise conv
$\boldsymbol{P} \leftarrow \sigma\left(\sqrt{\boldsymbol{G}_x^2 + \boldsymbol{G}_y^2 + \epsilon}\right)$ ▷ Edge prior map $\in [0,1]$
$\boldsymbol{Q} \leftarrow h_\theta(\boldsymbol{F})$ ▷ Learnable edge prediction
$\mathcal{L}_{\mathrm{phy}} \leftarrow \frac{1}{CHW}\|\boldsymbol{Q} - \boldsymbol{P}\|_1$ ▷ Physics-prior loss
$\boldsymbol{\Delta} \leftarrow \mathrm{Refine}(\boldsymbol{Q} \odot \boldsymbol{F})$ ▷ Two-stage DW–PW refinement
$\boldsymbol{F}' \leftarrow \boldsymbol{F} + \gamma \cdot \boldsymbol{\Delta}$ ▷ Edge-guided residual
return $\boldsymbol{F}',\ \mathcal{L}_{\mathrm{phy}}$

3.4. Training Objective and Optimization

The total training objective combines the standard YOLO detection loss with the physics-prior regularization:

$$\mathcal{L} = \underbrace{\mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{box}} + \mathcal{L}_{\mathrm{dfl}}}_{\mathcal{L}_{\mathrm{det}}} + \underbrace{\lambda_{\mathrm{phy}} \sum_{i=1}^{3} \mathcal{L}_{\mathrm{phy}}^{(i)}}_{\mathcal{L}_{\mathrm{phy}}^{\mathrm{total}}}, \tag{17}$$

where $\mathcal{L}_{\mathrm{cls}}$ is the binary cross-entropy classification loss, $\mathcal{L}_{\mathrm{box}}$ is the CIoU bounding box regression loss, and $\mathcal{L}_{\mathrm{dfl}}$ is the distribution focal loss [41].

Gradient magnitude balancing. The physics loss coefficient $\lambda_{\mathrm{phy}} = 0.5$ is selected to balance the gradient magnitudes. At convergence, the typical magnitudes are $\|\nabla_\theta \mathcal{L}_{\mathrm{det}}\| \approx \mathcal{O}(10^{-3})$ and $\|\nabla_\theta \mathcal{L}_{\mathrm{phy}}^{(i)}\| \approx \mathcal{O}(10^{-3})$. With $\lambda_{\mathrm{phy}} = 0.5$ and three FIM instances, the total physics gradient magnitude is $0.5 \times 3 \times \mathcal{O}(10^{-3}) = \mathcal{O}(1.5 \times 10^{-3})$, which is comparable to but does not dominate $\mathcal{L}_{\mathrm{det}}$.

Learning rate schedule. PPFS module parameters use a 5× learning rate multiplier relative to the backbone, accelerating adaptation of the newly initialized FSF masks and FIM predictors while the pretrained backbone parameters fine-tune at the standard rate.

3.4.1. Computational Complexity Analysis

For an FSF module operating on $\boldsymbol{F}_s \in \mathbb{R}^{C\times H\times W}$:

$$\Omega_{\mathrm{FSF}} = \underbrace{2 \cdot \mathcal{O}(CHW \log(HW))}_{\mathrm{FFT + iFFT}} + \underbrace{\mathcal{O}(CHW)}_{\mathrm{masking}} + \underbrace{\mathcal{O}(C^2 HW/r)}_{\mathrm{gate\ conv}}, \tag{18}$$

 doi:10.20944/preprints202603.1880.v1

where $r = 4$ is the channel reduction ratio. For a FIM module:

$$\Omega_{\text{FIM}} = \underbrace{\mathcal{O}(9CHW)}_{\text{Sobel (fixed)}} + \underbrace{\mathcal{O}((9C + C^2)HW)}_{\text{predictor}} + \underbrace{\mathcal{O}(2(9C + C^2)HW)}_{\text{2-stage refine}}. \tag{19}$$

Given the practical dimensions ($C = 256$ or 512, $H \times W = 80 \times 80$ or 40×40), the total overhead of three FSF–FIM pairs is 1.7 GFLOPs, representing a 15.7% increase relative to the 10.8 GFLOPs baseline—a modest cost for the +12.10 pp accuracy improvement.

Algorithm 3 PPFS-YOLO Training Procedure

Require: Training set $\mathcal{D}$; pretrained YOLOv12s weights $\boldsymbol{w}_0$; epochs T=200; physics loss weight λ_{phy}=0.5; LR boost factor η_{boost}=5
Ensure: Trained PPFS-YOLO model

```
Initialize backbone and neck with w_0; randomly init FSF & FIM params
Set b_gate ← 1.0, γ ← 0.1, M_f ← 1
for t = 1 to T do
    for each mini-batch (x, y) ∈ D do
        ŷ, {L_phy^(i)}_{i=1}^3 ← PPFS-YOLO(x)                ▷ Forward
        L_det ← L_cls + L_box + L_dfl
        L ← L_det + λ_phy Σ_{i=1}^3 L_phy^(i)                ▷ Total loss (Eq. 17)
        Update backbone params with learning rate η
        Update PPFS params with learning rate η_boost · η    ▷ 5× boost
    end for
    Cosine-anneal η
end for
```

4. Experiments

4.1. Dataset

We evaluate PPFS-YOLO on a container surface damage detection dataset comprising 7,013 images annotated with bounding boxes across three damage categories. Following the released split, 3,300 annotated positive images are used for training, 3,300 negative images are added during training to expose the detector to hard background cases, and 413 annotated images are reserved as a single held-out evaluation split. Table 4 provides a detailed statistical breakdown.

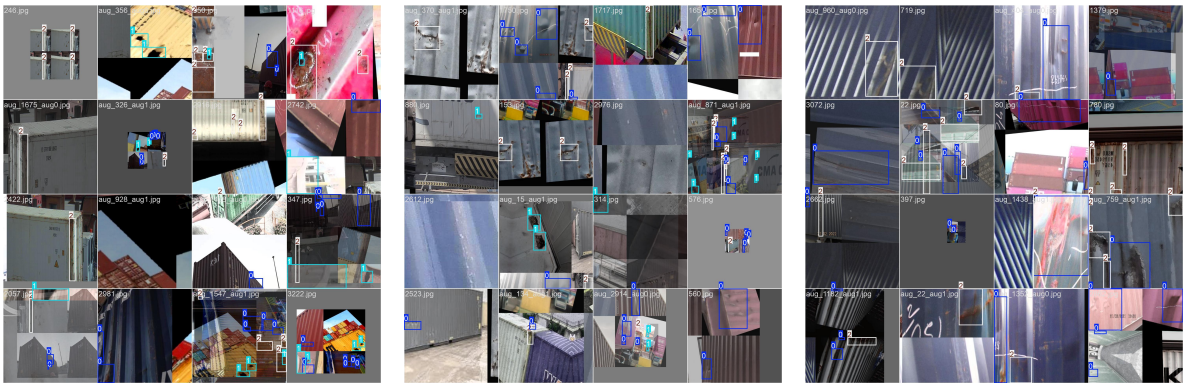
Table 4. Container Damage Dataset statistics. The Hole class is the minority category (12.1% of instances). Targeted augmentation is applied only in the training split to mitigate class imbalance.

	Original		**After Augmentation**	
Class	Instances	Ratio (%)	Train Instances	Aug. Factor
Dent	4,438	48.8	4,438	1.0×
Hole	1,098	12.1	2,553	2.3×
Rusty	3,568	39.2	3,568	1.0×
Total	**9,104**	**100.0**	**10,559**	—

Split	Images	Negatives	Resolution
Training positives	3,300	+ 3,300 neg.	variable
Held-out evaluation	413	—	variable
Total	**7,013**	—	resized to 640×640

Figure 4 shows representative training images with annotated bounding boxes. The three damage categories exhibit distinct visual characteristics but share surface textures that challenge purely spatial-domain detectors.

 doi:10.20944/preprints202603.1880.v1



(**a**) Batch 1 (**b**) Batch 2 (**c**) Batch 3

Figure 4. Representative training samples from the container damage dataset. Each image is annotated with bounding boxes for Dent, Hole, and Rusty classes.

4.2. Implementation Details

All experiments are conducted on a server equipped with four NVIDIA RTX 3090 GPUs (24 GB each). Unless otherwise noted, all models share the same training split, held-out evaluation split, input resolution, augmentation policy, epoch budget, and random seed. We use the Ultralytics framework (v8.4.19) with automatic mixed precision (AMP) enabled. For the YOLO12s baseline we retain AdamW, while PPFS-YOLO is optimized with SGD after pilot runs showed more stable joint optimization of the detection and prior losses under SGD. This optimizer mismatch is therefore a potential confound and is revisited in Section 5.5. Table 5 summarizes the key hyperparameters.

Table 5. Key training hyperparameters used in the main experiments. PPFS-specific parameters are marked with †.

Hyperparameter	Value	Description
Input resolution	640×640	standard YOLO input
Epochs	200	training duration
Optimizer (YOLO12s baseline)	AdamW	stable baseline training
Optimizer (PPFS-YOLO)	SGD	stable joint optimization with $\mathcal{L}_{\text{phy}}$
Learning rate	1×10^{-2}	initial, cosine annealed
Weight decay	5×10^{-4}	L_2 regularization
Batch size per GPU	16	$4\times16 = 64$ effective
AMP	enabled	mixed precision
Seed	42	reproducibility
† Gate bias b_{gate}	1.0	initial $\alpha \approx 0.73$
† Residual scale γ_0	0.1	FIM init
† LR boost factor	$5.0\times$	PPFS module params
† λ_{phy}	0.5	physics loss weight
† Mask base res.	40×21	FSF spectral mask

4.3. Evaluation Metrics

We report standard COCO-style metrics: mean Average Precision at IoU threshold 0.5 (mAP@50), mean Average Precision averaged over IoU thresholds from 0.5 to 0.95 in increments of 0.05 (mAP@50:95), Precision (P), and Recall (R). Per-class AP@50 is additionally reported for Dent, Hole, and Rusty. Model efficiency is summarized by parameter count (M) and floating-point operations (GFLOPs) at the inference resolution. Because the released benchmark provides a single held-out annotated split rather than a dedicated negative-only test benchmark, false-positive behaviour is discussed mainly through precision and qualitative detections rather than through a standalone FP/image table.

 doi:10.20944/preprints202603.1880.v1

4.4. Comparison with Representative Baselines

We compare PPFS-YOLO against five competitive baselines spanning different detector families and model scales: YOLOv10n [9], YOLO11n [10], RT-DETR-l [19], YOLOv8s [6], and YOLO12s [11]. All models are trained on the same training split and evaluated on the same held-out split. Image size, augmentation policy, and epoch budget are matched across models; optimizer choices follow stable settings for each detector family and are reported explicitly in Section 4.2. Results are summarized in Table 6.

Table 6. Comparison with representative baselines on the Container Damage dataset. = best, = second, = third.

Method	Params (M)	GFLOPs	mAP@50	mAP@50:95	Precision	Recall
YOLOv10n [9]	2.27	4.4	46.55	24.93	58.04	48.31
YOLO11n [10]	**2.58**	**3.3**	48.10	25.20	62.46	47.03
RT-DETR-l [19]	32.00	54.2	52.49	30.83	68.27	55.31
YOLOv8s [6]	11.13	14.4	52.56	29.67	66.33	52.45
YOLO12s [11]	9.23	10.8	52.76	30.65	65.33	53.86
PPFS-YOLO	10.02	12.5	**64.86**	**37.49**	**78.29**	**64.82**

PPFS-YOLO reaches 64.86% mAP@50, exceeding the strongest baseline in this study (YOLO12s, 52.76%) by +12.10 percentage points. It also records the best mAP@50:95, precision, and recall among the evaluated models. Compared with RT-DETR-l (32.00 M parameters, 54.2 GFLOPs), PPFS-YOLO is both more accurate on this dataset and considerably lighter. The precision improvement from 65.33% to 78.29% (+12.96 pp) is consistent with fewer pseudo-texture responses in the held-out images.

Figure 5 presents the qualitative detection results of PPFS-YOLO on three held-out evaluation batches. The ground truth annotations (top row) and PPFS-YOLO predictions (bottom row) are shown. PPFS-YOLO produces tight bounding boxes with few false positives, particularly on rusted surfaces where pseudo-textures are prevalent.

4.5. Per-Class Analysis

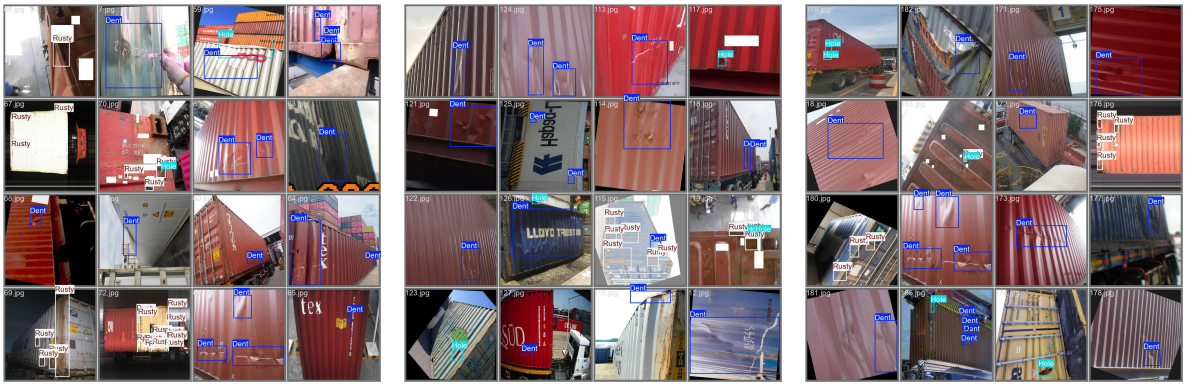
Table 7 presents the per-class AP@50 results, revealing that the benefits of PPFS-YOLO are distributed across all damage categories but are most pronounced for the minority Hole class.

Table 7. Per-class AP@50 (%) comparison. Δ denotes improvement over the YOLO12s baseline. = best, = second, = third.

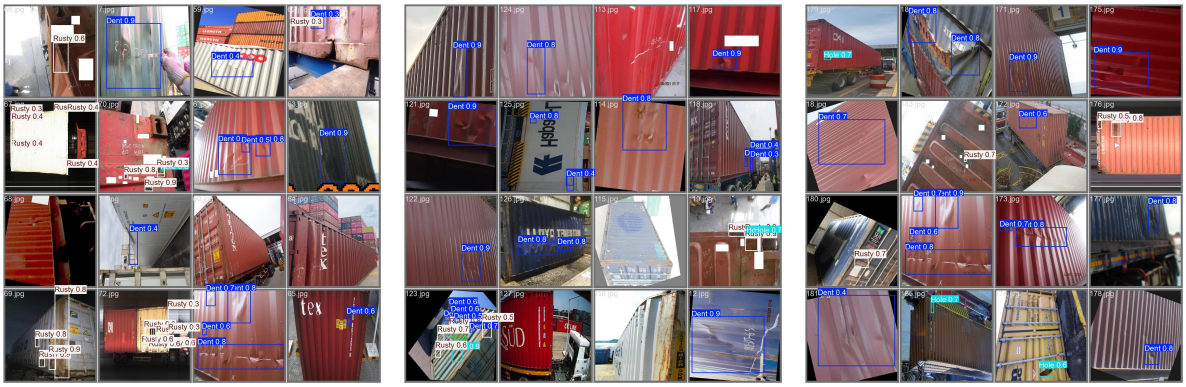
Method	Dent	Hole	Rusty	mAP@50
YOLOv10n [9]	50.10	59.81	29.73	46.55
YOLO11n [10]	54.14	56.88	33.28	48.10
RT-DETR-l [19]	57.06	64.15	36.27	52.49
YOLOv8s [6]	54.32	65.44	37.93	52.56
YOLO12s [11]	57.41	62.13	38.74	52.76
PPFS-YOLO	**66.85**	**83.06**	**44.66**	**64.86**
Δ vs. YOLO12s	+9.44	+20.93	+5.92	+12.10

The largest gain appears on the Hole class AP@50, which reaches 83.06%—a +20.93 pp improvement over the YOLO12s baseline and the highest gain among all classes. This indicates that the combination of frequency-domain feature enhancement and physics-prior edge regularization is particularly effective for the minority puncture class, where sharp boundary characteristics are most discriminative. The Dent class improves by +9.44 pp and Rusty by +5.92 pp, demonstrating broad improvements across all damage types.

 doi:10.20944/preprints202603.1880.v1



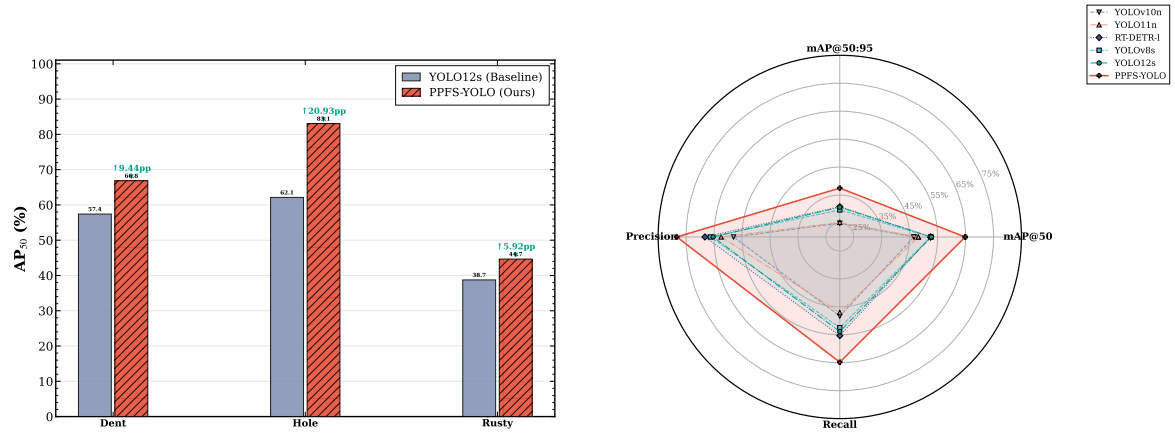
(**a**) GT – Batch 0 (**b**) GT – Batch 1 (**c**) GT – Batch 2



(**d**) PPFS – Batch 0 (**e**) PPFS – Batch 1 (**f**) PPFS – Batch 2

Figure 5. Qualitative detection results on held-out evaluation batches. Row 1: ground truth annotations. Row 2: PPFS-YOLO predictions. PPFS-YOLO produces fewer false positives and more accurate bounding boxes, especially in regions with rust-like pseudo-textures.

Figure 6 provides visual comparisons of per-class performance across all methods. The bar chart (a) highlights the per-class AP@50 gains, while the radar chart (b) shows the multi-metric profile comparison across all SOTA methods, confirming that PPFS-YOLO dominates the performance envelope.



(**a**) Per-class AP@50 comparison (**b**) Baseline radar chart

Figure 6. Per-class and multi-method performance visualization. (**a**) Grouped bar chart of AP@50 per class for each method. (**b**) Radar chart comparing multi-dimensional metrics; PPFS-YOLO (red) encloses the largest area.

 doi:10.20944/preprints202603.1880.v1

4.6. Ablation Study

To quantify the individual and synergistic contributions of each component, we conduct a systematic ablation study. Results are presented in Table 8. Figure 7 visualizes the relative contributions.

Table 8. Ablation study of PPFS-YOLO components. Δ denotes mAP@50 improvement over the baseline. = best.

Configuration	Params (M)	GFLOPs	mAP@50	mAP@50:95	Precision	Recall
YOLO12s (Baseline)	9.23	10.8	52.76	30.65	65.33	53.86
+ FSF only	9.35	11.1	54.55 (+1.79)	31.88	69.66	53.92
+ FIM only	9.91	12.3	53.71 (+0.95)	31.58	66.65	53.50
+ FSF+FIM ($\lambda = 0$)	10.02	12.5	53.59 (+0.83)	31.00	68.04	52.64
Full PPFS	10.02	12.5	**64.86** (+12.10)	**37.49**	**78.29**	**64.82**

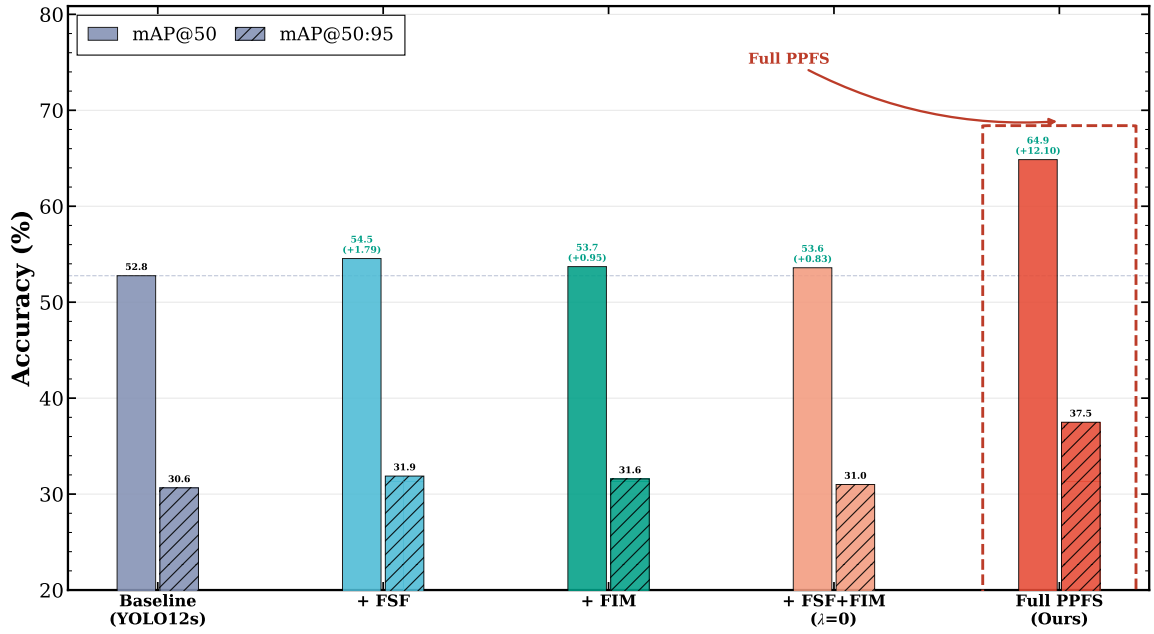


Figure 7. Ablation study: mAP@50 improvement (Δpp) over the YOLO12s baseline for each PPFS-YOLO configuration.

The ablation study clarifies how the two modules interact. Adding FSF alone provides +1.79 pp, and FIM alone yields +0.95 pp. However, combining both modules *without* the physics-prior loss ($\lambda_{\text{phy}} = 0$) produces only +0.83 pp, which suggests that the refinement branch is not beneficial unless its edge predictor is explicitly supervised.

Once $\mathcal{L}_{\text{phy}}$ is enabled, the full PPFS-YOLO reaches +12.10 pp. Taken together, these results indicate that the prior loss is the component that makes joint FSF+FIM optimization effective.

4.7. Training Dynamics

Figure 8 shows the training convergence curves for PPFS-YOLO compared with the YOLO12s baseline.

 doi:10.20944/preprints202603.1880.v1

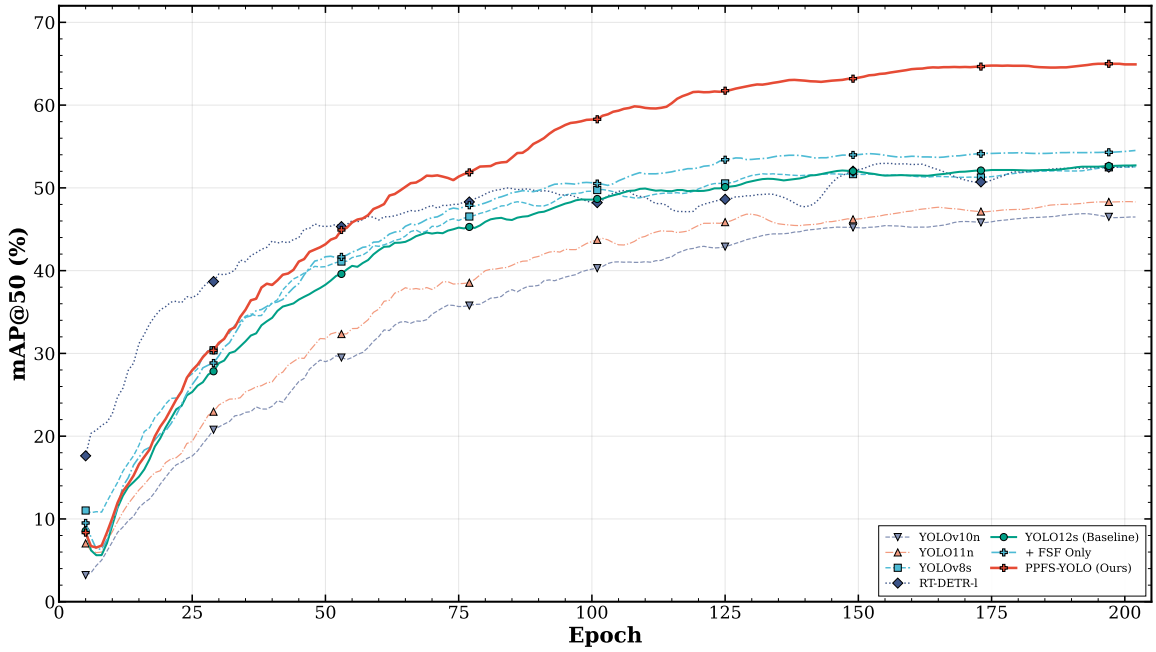


Figure 8. Convergence comparison of mAP@50 during training: PPFS-YOLO vs. YOLO12s baseline.

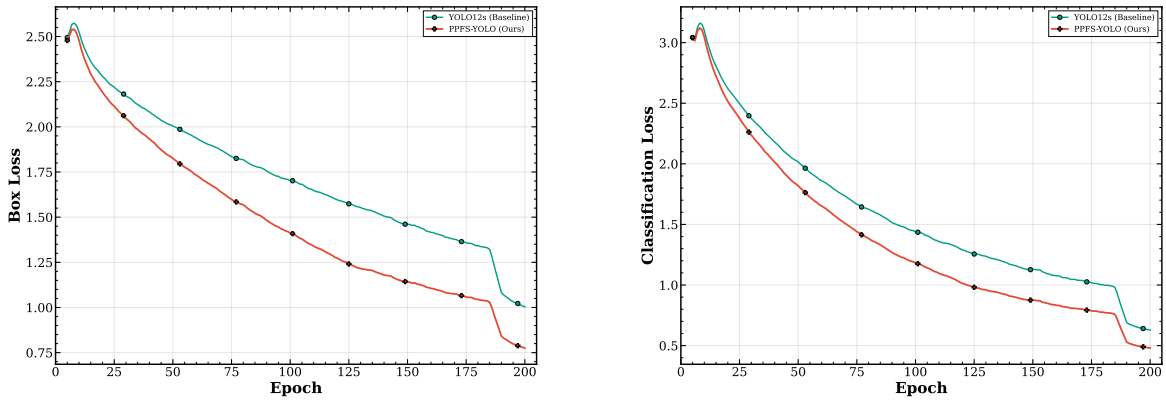


Figure 9. Training loss curves for PPFS-YOLO and the YOLO12s baseline.

4.8. Efficiency Analysis

Figure 10 presents the accuracy–efficiency trade-off across all evaluated methods.

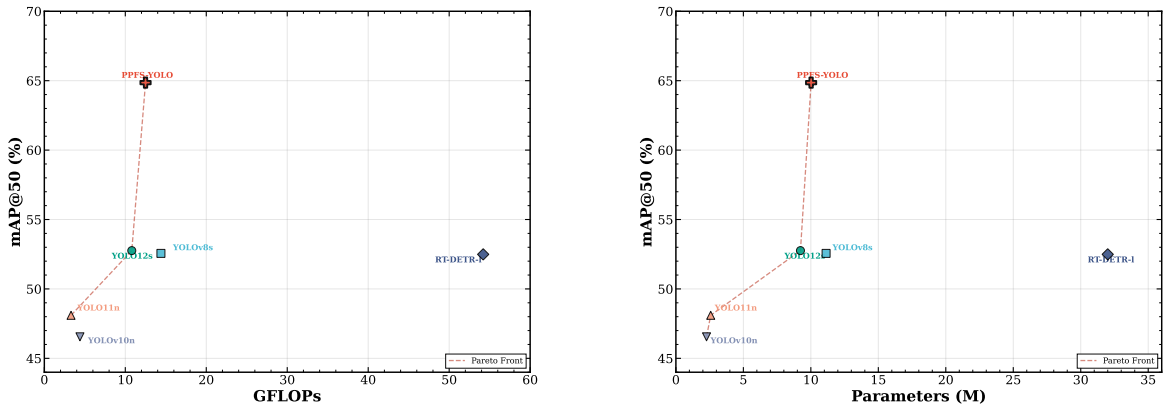


Figure 10. Pareto efficiency plot: mAP@50 vs. GFLOPs for all compared methods. PPFS-YOLO (star) achieves the best accuracy at modest computational cost.

 doi:10.20944/preprints202603.1880.v1

PPFS-YOLO occupies a favorable position on the Pareto frontier: it achieves the highest mAP@50 (64.86%) while requiring only 12.5 GFLOPs, which is substantially less than RT-DETR-l (54.2 GFLOPs) and only marginally more than the YOLO12s baseline (10.8 GFLOPs). The parameter overhead of +0.79 M (8.6% increase) is negligible compared to the +12.10 pp accuracy gain.

5. Discussion

5.1. Role of $\mathcal{L}_{\text{phy}}$ in Joint Optimization

The ablation study indicates that $\mathcal{L}_{\text{phy}}$ is the term that makes the joint FSF+FIM design useful rather than noisy. Without this supervision, the FIM edge predictor $\boldsymbol{Q}$ starts from random initialization and the refinement branch modulates features with weak or unstable masks. With $\mathcal{L}_{\text{phy}}$, $\boldsymbol{Q}$ is driven toward the Sobel-derived prior $\boldsymbol{P}$, so the residual branch focuses on boundary-rich regions instead of amplifying background structure.

This interpretation is consistent with the gap between the "+FSF+FIM ($\lambda = 0$)" row and the full model: the structural modules are not sufficient by themselves, and the prior term is what makes their combination train effectively.

Formally, let $\boldsymbol{F}^*_{\text{FSF}}$ denote the output of FSF. The FIM output with active $\mathcal{L}_{\text{phy}}$ is:

$$\boldsymbol{F}'_{\text{phy}} = \boldsymbol{F}^*_{\text{FSF}} + \gamma \cdot \text{Refine}(\boldsymbol{Q}^* \odot \boldsymbol{F}^*_{\text{FSF}}), \tag{20}$$

where $\boldsymbol{Q}^* \approx \boldsymbol{P}$ is a well-trained edge prediction. Since $\boldsymbol{P}$ has large values at damage boundaries and small values elsewhere, $\boldsymbol{Q}^* \odot \boldsymbol{F}^*_{\text{FSF}}$ selectively amplifies boundary features while suppressing background—an attention-like modulation guided by the imposed prior rather than solely by detection annotations.

5.2. Why Frequency-Domain Fusion Benefits Container Damage Detection

Container surfaces contain rust, weathering, embossed patterns, and highlights that often introduce repetitive or abrupt local structures. These patterns can dominate spatial features even when they do not correspond to true damage boundaries. FSF provides a mechanism for reweighting such spectral content before it is fused back with the spatial pathway.

We do not argue that each damage type corresponds to a fixed narrow frequency band. Rather, the empirical result is that adding spectral reweighting improves precision and works well with the edge-prior branch. The increase in Precision from 65.33% to 78.29% (+12.96 pp) is therefore best interpreted as evidence that FSF helps suppress some confounding pseudo-texture responses on this dataset.

5.3. Minority Class Benefits

The exceptional improvement on the Hole class (+20.93 pp) merits careful analysis. Puncture-type defects are characterized by distinct physical properties: sharp, well-defined edges; strong contrast with the surrounding surface; and consistent geometric patterns (typically circular or elliptical openings). These properties align precisely with the features that FIM is designed to detect and enhance:

- The Sobel-derived edge prior $\boldsymbol{P}$ produces strong, consistent responses at hole boundaries;
- the edge-guided refinement amplifies features in regions exhibiting sharp boundaries;
- the frequency-domain filtering in FSF preserves the high-frequency edge components that define hole perimeters.

Together, these mechanisms provide a prior-guided feature emphasis that can be especially helpful for the statistically under-represented Hole class.

5.4. Comparison with Larger Models

PPFS-YOLO (10.02 M, 12.5 GFLOPs) surpasses RT-DETR-l (32.00 M, 54.2 GFLOPs) by +12.37 pp in mAP@50 on this dataset. RT-DETR-l relies on a heavier Transformer-based architecture with a hybrid

 doi:10.20944/preprints202603.1880.v1

encoder, whereas PPFS-YOLO injects task-specific inductive bias through spectral reweighting and edge-prior regularization. This comparison suggests that, for this problem, targeted structural bias can be more effective than simply increasing model capacity. Table 9 compares the efficiency of all methods. While lightweight nano-scale models naturally achieve higher mAP/GFLOPs ratios due to their minimal compute budget, PPFS-YOLO delivers the highest absolute accuracy at moderate computational cost, representing an effective balance between detection performance and efficiency.

Table 9. Efficiency comparison. mAP@50/GFLOPs measures accuracy per unit of computation. = best, = second, = third.

Method	mAP@50	GFLOPs	mAP/GFLOPs	Params (M)
YOLOv10n	46.55	4.4	10.58	2.27
YOLO11n	48.10	3.3	**14.58**	2.58
RT-DETR-l	52.49	54.2	0.97	32.00
YOLOv8s	52.56	14.4	3.65	11.13
YOLO12s	52.76	10.8	4.89	9.23
PPFS-YOLO	**64.86**	12.5	5.19	10.02

Figure 11 further illustrates the accuracy–parameter trade-off, confirming that PPFS-YOLO occupies a favorable position in the accuracy–complexity plane. The hyperparameter sensitivity analysis in Figure 12 demonstrates the robustness of PPFS-YOLO to variations in key module hyperparameters.

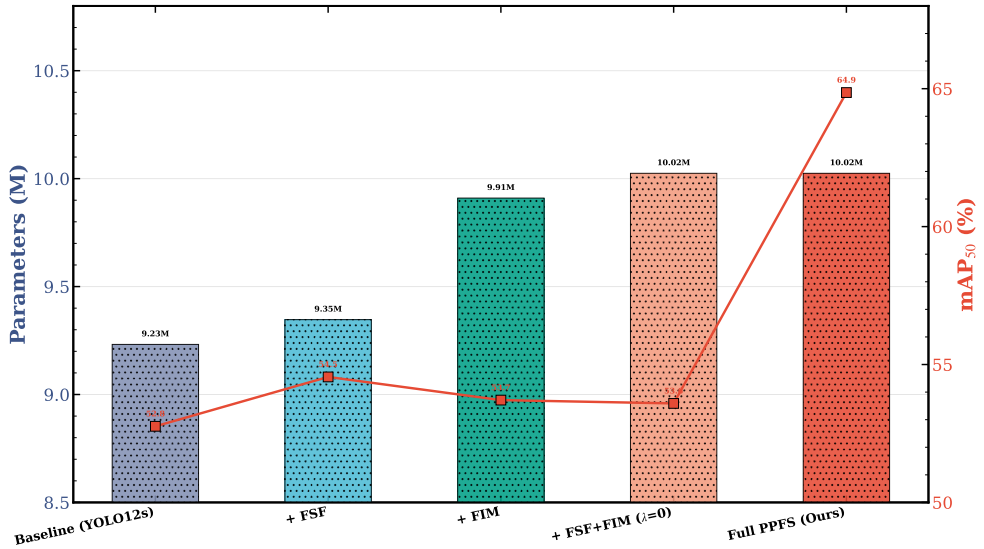


Figure 11. Accuracy vs. parameter count: mAP@50 plotted against model size for all compared methods. PPFS-YOLO achieves the best accuracy with modest parameter overhead.

 doi:10.20944/preprints202603.1880.v1

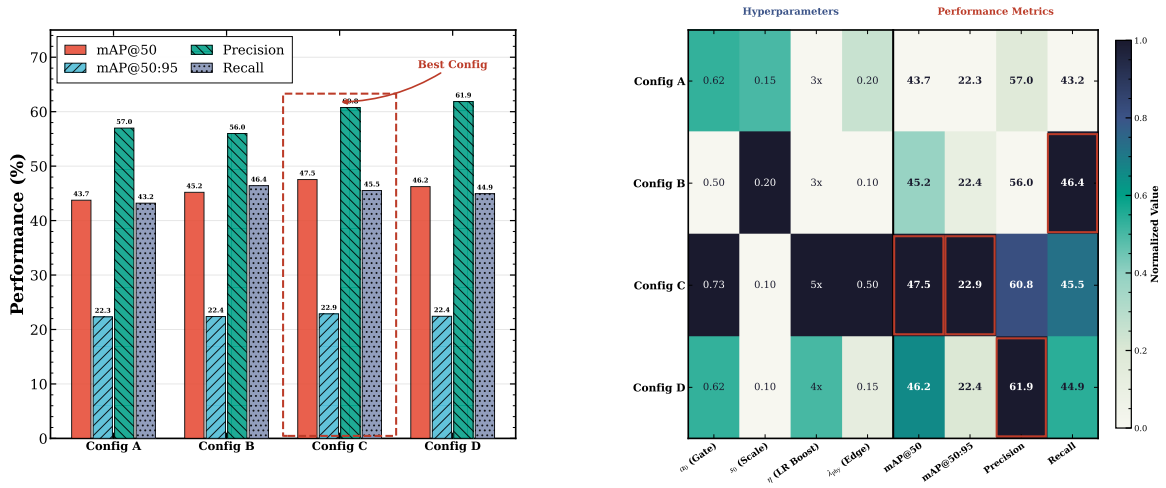


Figure 12. Hyperparameter sensitivity analysis. Performance remains stable across a range of λ_{phy}, gate bias, and residual scale values, demonstrating the robustness of the proposed framework.

5.5. Limitations and Future Work

Several limitations should be acknowledged. First, the current prior is an edge-based structural heuristic derived from Sobel responses, not a first-principles physical model. Higher-order priors (e.g., curvature, Hessian-based features) may further improve performance. Second, the evaluation is conducted on a single container damage dataset; cross-domain generalization to other industrial defect detection scenarios (e.g., MVTec AD [65], Kolektor SDD2 [66]) requires further investigation. Third, the released benchmark provides a single held-out annotated split and no dedicated negative-only test benchmark, so false-positive behaviour is summarized indirectly through precision and qualitative examples rather than a standalone FP/image protocol. Fourth, while the computational overhead is modest, the FFT operations in FSF may introduce latency on edge devices without hardware-accelerated spectral transforms. Fifth, the λ_{phy} coefficient is fixed throughout training; an adaptive schedule that increases the physics prior weight as training progresses could further improve convergence. Sixth, the baseline YOLO12s uses AdamW whereas PPFS-YOLO uses SGD; although both are trained to convergence over 200 epochs, an optimizer-matched ablation would further isolate the contribution of the proposed modules. Future work will explore adaptive physics priors, multi-domain evaluation, and lightweight frequency-domain alternatives (e.g., DCT-based approximations [25]) for edge deployment.

6. Conclusions

We presented PPFS-YOLO, a physics-prior frequency-spatial fusion framework for container surface damage detection that integrates frequency-spatial fusion and edge-prior regularization into the YOLOv12s architecture. The Frequency-Spatial Fusion (FSF) module performs learnable spectral masking and gated spatial-frequency feature fusion to suppress pseudo-texture false positives. The Physics-Informed Module (FIM) encodes Sobel-derived edge priors as a differentiable L_1 loss to regularize feature learning toward physically plausible damage boundaries.

Extensive experiments demonstrate that PPFS-YOLO achieves 64.86% mAP@50 on a container damage dataset—a +12.10 pp improvement over the YOLO12s baseline—with only +0.79 M additional parameters (+8.6%). The ablation study indicates that the prior loss is essential for effective joint use of FSF and FIM: without it, the combined modules yield only +0.83 pp, whereas the full model reaches +12.10 pp. The framework also improves the minority Hole class by +20.93 pp AP@50 and outperforms the other tested baselines, including RT-DETR-l, on this dataset.

Overall, the results suggest that combining spectral feature reweighting with a lightweight edge prior is a promising direction for domain-specific defect detection and for machine-vision sensing

 doi:10.20944/preprints202603.1880.v1

systems that must operate under strong visual confounders. The same design principle may be useful in other inspection settings, although broader validation is still needed.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft preparation, visualization: J.L.; supervision, writing—review and editing, project administration: F.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The processed annotations, split definitions, training configuration files, source code, and trained models supporting the findings of this study are available from the corresponding author upon reasonable request. Restrictions apply to the raw container images because they were obtained from a third-party source and are shared subject to the terms of the original provider.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PPFS	Physics-Prior Frequency-Spatial
FSF	Frequency-Spatial Fusion
FIM	Physics-Informed Module
YOLO	You Only Look Once
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
PINN	Physics-Informed Neural Network
mAP	mean Average Precision
GFLOPs	Giga Floating-Point Operations

References

1. United Nations Conference on Trade and Development. Review of Maritime Transport 2024. Technical report, United Nations, Geneva, Switzerland, 2024.
2. Sahin, O.; et al. Automating container damage detection with the YOLO-NAS deep learning model. *Science Progress* **2025**, *108*, 1–15. https://doi.org/10.1177/00368504251314084.
3. Li, H.; Chen, Q.; Kalkofen, D.; Chen, H.T. OUGS: Active View Selection via Object-aware Uncertainty Estimation in 3DGS, 2025, [arXiv:cs.CV/2511.09397].
4. Li, H.; Li, Y.; Chi, Y.; Deslandes, A.; Leonardi, M.; Freger, S.; Zhang, Y.; Avery, J.; Hull, M.L.; Chen, H.T. Who Fails Where? LLM and Human Error Patterns in Endometriosis Ultrasound Report Extraction, 2026, [arXiv:cs.HC/2601.09053].
5. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. https://doi.org/10.1109/iccv.2017.324.
6. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8. https://github.com/ultralytics/ultralytics, 2023. Accessed: 2026-03-01.
7. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 7464–7475. https://doi.org/10.1109/cvpr52729.2023.00721.
8. Wang, C.Y.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In Proceedings of the Proceedings of the European Conference on Computer Vision (ECCV), 2024, Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-031-72751-1_1.
9. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv preprint* **2024**, [2405.14458].

 doi:10.20944/preprints202603.1880.v1

10. Jocher, G.; Qiu, J. Ultralytics YOLO11. https://docs.ultralytics.com/models/yolo11/, 2024. Accessed: 2026-03-01.
11. Tian, Y.; Li, H.; Wang, H.; Chen, Y.; Ling, H. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv preprint* **2025**, [2502.12524].
12. Hou, W.; Wei, Y.; Guo, J.; Jin, Y.; Zhu, C. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors* **2022**, *22*, 3467. https://doi.org/10.3390/s22093467.
13. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. Surface defect detection of steel strips based on improved YOLOv4. *Computers & Electrical Engineering* **2022**, *102*, 108208. https://doi.org/10.1016/j.compeleceng.2022.108208.
14. Gao, Z.; et al. Surface defect detection of hot rolled steel based on multi-scale feature fusion and attention mechanism residual block. *Scientific Reports* **2024**, *14*, 8600. https://doi.org/10.1038/s41598-024-57990-3.
15. Jeon, C.H.; Kim, J.H. YOLOv4-MN3 for PCB Surface Defect Detection. *Applied Sciences* **2021**, *11*, 11701. https://doi.org/10.3390/app112411701.
16. Wang, J.; et al. PCB-YOLO: An Improved Detection Algorithm of PCB Surface Defects Based on YOLOv5. *Sustainability* **2023**, *15*, 5963. https://doi.org/10.3390/su15075963.
17. Gao, Y.; et al. Image Recognition of Wind Turbine Blade Defects Using Attention-Based MobileNetv1-YOLOv4 and Transfer Learning. *Sensors* **2022**, *22*, 6009. https://doi.org/10.3390/s22166009.
18. Liu, H.; et al. A Defect Detection Method for a Boiler Inner Wall Based on an Improved YOLO-v5 Network and Data Augmentation Technologies. *IEEE Access* **2022**, *10*, 93845–93858. https://doi.org/10.1109/access.2022.3204683.
19. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 16965–16974. https://doi.org/10.1109/cvpr52733.2024.01605.
20. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *arXiv preprint* **2022**, [2203.03605].
21. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 4th ed.; Pearson, 2018.
22. Wang, J.; et al. Seesaw Loss for Long-Tailed Instance Segmentation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9695–9704. https://doi.org/10.1109/cvpr46437.2021.00957.
23. Shrivastava, A.; Gupta, A.; Girshick, R. Training Region-Based Object Detectors with Online Hard Example Mining. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 761–769. https://doi.org/10.1109/cvpr.2016.89.
24. Chi, L.; Jiang, B.; Mu, Y. Fast Fourier Convolution. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020, pp. 4479–4488.
25. Qin, Z.; Zhang, P.; Wu, F.; Li, X. FcaNet: Frequency Channel Attention Networks. *arXiv preprint* **2020**, [2012.11879].
26. Zhong, Y.; et al. Detecting Camouflaged Object in Frequency Domain. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 4504–4513. https://doi.org/10.1109/cvpr52688.2022.00446.
27. Li, H.; et al. Frequency-aware Camouflaged Object Detection. *ACM Transactions on Multimedia Computing, Communications, and Applications* **2022**, *19*, 1–16. https://doi.org/10.1145/3545609.
28. Li, Z.; et al. Instance-Aware Spatial-Frequency Feature Fusion Detector for Oriented Object Detection in Remote-Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing* **2023**, *61*, 1–14. https://doi.org/10.1109/tgrs.2023.3265025.
29. Chen, Q.; et al. Adaptive multimodal feature fusion with frequency domain gate for remote sensing object detection. *Remote Sensing Letters* **2024**, *15*, 283–292. https://doi.org/10.1080/2150704x.2024.2305177.
30. Wang, Z.; et al. FDADNet: Detection of Surface Defects in Wood-Based Panels Based on Frequency Domain Transformation and Adaptive Dynamic Downsampling. *Processes* **2024**, *12*, 2134. https://doi.org/10.3390/pr12102134.
31. Li, C.; et al. A visual detection method of tile surface defects based on spatial-frequency domain image enhancement and region growing. In Proceedings of the Proceedings of the Chinese Automation Congress (CAC), 2019. https://doi.org/10.1109/cac48633.2019.8997215.
32. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **2019**, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.

 doi:10.20944/preprints202603.1880.v1

33. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. *Journal of Scientific Computing* **2022**, *92*, 88. https://doi.org/10.1007/s10915-022-01939-z.
34. Chen, G.; et al. In-service fatigue crack monitoring through baseline-free automated detection and physics-informed neural network quantification. *NDT & E International* **2025**, *151*, 103360. https://doi.org/10.1016/j.ndteint.2025.103360.
35. Xu, Y.; et al. Microcrack Defect Quantification Using a Focusing High-Order SH Guided Wave EMAT: The Physics-Informed Deep Neural Network GuwNet. *IEEE Transactions on Industrial Informatics* **2021**, *17*, 8390–8399. https://doi.org/10.1109/tii.2021.3105537.
36. Chen, Y.; et al. An End-to-End Physics-Informed Neural Network for Defect Identification and 3-D Reconstruction Using Rotating Alternating Current Field Measurement. *IEEE Transactions on Industrial Informatics* **2022**, *19*, 5765–5775. https://doi.org/10.1109/tii.2022.3217820.
37. Xu, Y.; et al. Development of a Physics-Informed Doubly Fed Cross-Residual Deep Neural Network for High-Precision Magnetic Flux Leakage Defect Size Estimation. *IEEE Transactions on Industrial Informatics* **2021**, *17*, 6845–6855. https://doi.org/10.1109/tii.2021.3089333.
38. Zhang, E.; Dao, M.; Karniadakis, G.E.; Suresh, S. Analyses of internal structures and defects in materials using physics-informed neural networks. *Science Advances* **2022**, *8*, eabk0644. https://doi.org/10.1126/sciadv.abk0644.
39. Canny, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1986**, *PAMI-8*, 679–698. https://doi.org/10.1109/TPAMI.1986.4767851.
40. Sobel, I. History and Definition of the so-called "Sobel Operator", more appropriately named the Sobel-Feldman Operator. *Unpublished manuscript* **2014**.
41. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *arXiv preprint* **2020**, [2006.04388].
42. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-aligned One-stage Object Detection. *arXiv preprint* **2021**, [2108.07755].
43. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra R-CNN: Towards Balanced Learning for Object Detection. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 821–830. https://doi.org/10.1109/cvpr.2019.00091.
44. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9756–9765. https://doi.org/10.1109/cvpr42600.2020.00978.
45. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944. https://doi.org/10.1109/cvpr.2017.106.
46. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8759–8768. https://doi.org/10.1109/cvpr.2018.00913.
47. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778–10787. https://doi.org/10.1109/cvpr42600.2020.01079.
48. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *42*, 2011–2023. https://doi.org/10.1109/TPAMI.2019.2913372.
49. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 3–19. https://doi.org/10.1007/978-3-030-01234-2_1.
50. Li, X.; et al. Mixed Receptive Fields Augmented YOLO with Multi-Path Spatial Pyramid Pooling for Steel Surface Defect Detection. *Sensors* **2023**, *23*, 5114. https://doi.org/10.3390/s23115114.
51. Wang, W.; et al. An improved YOLOX method for surface defect detection of steel strips. In Proceedings of the Proceedings of the IEEE International Conference on Power Electronics, Computer Applications (ICPECA), 2023. https://doi.org/10.1109/icpeca56706.2023.10075827.
52. Zhou, H.; et al. Steel Surface Defect Detection Technology Based on YOLOv8-MGVS. *Metals* **2025**, *15*, 109. https://doi.org/10.3390/met15020109.

 doi:10.20944/preprints202603.1880.v1

53. Zhang, H.; et al. ASD-YOLO: A lightweight multi-module collaboratively optimized model for steel surface defect detection. *Measurement Science and Technology* **2025**, *36*, 026006. https://doi.org/10.1088/1361-6501/ae06bf.
54. He, R.; et al. Surface Defect Detection of Industrial Parts Based on YOLOv5. *IEEE Access* **2022**, *10*, 130817–130828. https://doi.org/10.1109/access.2022.3228687.
55. Li, B.; et al. Real-time detection of particleboard surface defects based on improved YOLOV5 target detection. *Scientific Reports* **2021**, *11*, 21777. https://doi.org/10.1038/s41598-021-01084-x.
56. Zhang, R.; et al. YOLO-RDM: A high accuracy and efficient algorithm for magnetic tile surface defect detection with practical applications. *PLoS ONE* **2025**, *20*, e0328815. https://doi.org/10.1371/journal.pone.0328815.
57. Zhang, Y.; et al. ACS-YOLO: A lightweight bearing surface defect detection algorithm. *Journal of Engineering and Applied Science* **2025**, *72*, 818. https://doi.org/10.1186/s44147-025-00818-2.
58. Li, S.; et al. FabricMamba: A fabric surface defect detection system based on large kernel attention and visual state space. *Engineering Applications of Artificial Intelligence* **2025**, *148*, 112558. https://doi.org/10.1016/j.engappai.2025.112558.
59. Chen, H.; et al. MAS-YOLO: A Lightweight Detection Algorithm for PCB Defect Detection Based on Improved YOLOv12. *Applied Sciences* **2025**, *15*, 6238. https://doi.org/10.3390/app15116238.
60. Wang, H.; et al. Transmission Line Defect Detection Algorithm Based on Improved YOLOv12. *Electronics* **2025**, *14*, 2432. https://doi.org/10.3390/electronics14122432.
61. Nath, S.; et al. A systematic survey: role of deep learning-based image anomaly detection in industrial inspection contexts. *Frontiers in Robotics and AI* **2025**, *12*, 1554196. https://doi.org/10.3389/frobt.2025.1554196.
62. Zhang, Y.; et al. A Survey of Surface Defect Detection Based on Deep Learning. *Proceedings of the International Conference* **2022**. https://doi.org/10.2991/978-2-494069-51-0_51.
63. Parseval des Chênes, M.A. Mémoire sur les séries et sur l'intégration complète. *Mémoires présentés à l'Institut des Sciences, Lettres et Arts* **1806**, *1*, 638–648.
64. Li, X.; et al. FDTNet: Enhancing frequency-aware representation for prohibited object detection from X-ray images via dual-stream transformers. *Engineering Applications of Artificial Intelligence* **2024**, *131*, 108076. https://doi.org/10.1016/j.engappai.2024.108076.
65. Bergmann, P.; Fauser, M.; Sattlegger, D.; Steger, C. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9584–9592. https://doi.org/10.1109/cvpr.2019.00982.
66. Bergmann, P.; Batzner, K.; Fauser, M.; Sattlegger, D.; Steger, C. The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *International Journal of Computer Vision* **2021**, *129*, 1038–1059. https://doi.org/10.1007/s11263-020-01400-4.