

Article

A Lightweight Parallel Attention U-Net for Surface Defect Segmentation of Wind Turbine Towers in Visible-Light Images

Fanqiang Zeng ¹, Renchaogetu Wu ², Yinan Ma ¹, Yu Zhang ¹, Wanpeng Ping ¹, Songbin Yang ¹ and Qingfei Gao ^{2,*}

¹ Heilongjiang Tietou Prefabricated Components Co., Ltd., Harbin 150000, China; giunti@sina.com (F.Z.); 15846611125@163.com (Y.M.); 15945955114@163.com (Y.Z.); 15045699433@163.com (W.P.); yangsongbin0451@126.com (S.Y.)

² School of Transportation Science and Engineering, Harbin Institute of Technology, Harbin 150090, China; 25s132040@stu.hit.edu.cn

* Correspondence: gaoqingfei@hit.edu.cn

Abstract

Wind turbine towers operate in complex outdoor environments, where visible surface anomalies such as cracks, pitting, and honeycombing can develop. Field-acquired visible-light images are commonly affected by illumination variation, shadows, local reflections, surface textures, and structural joints, which makes pixel-level anomaly segmentation difficult. This study proposes a task-oriented lightweight U-Net, termed LPAU-Net, that combines DWConv–PWConv feature extraction, parallel channel–spatial attention with learnable scalar fusion, grouped multi-level feature aggregation, and multi-scale decoding. Defect-free images are included during training, and defective and defect-free samples are evaluated separately to distinguish anomaly segmentation from false-positive suppression. The dataset contains 762 original field images collected from the same nine wind turbine towers at one wind farm during five time-separated acquisition campaigns. Campaigns 1–3 were used for training, Campaign 4 for validation, checkpoint selection, and threshold determination, and Campaign 5 for final evaluation. Thus, Campaign 5 is a later acquisition batch from the same towers and site, rather than unseen-tower or cross-wind-farm validation. Across three independent random seeds, LPAU-Net achieved $89.84 \pm 0.08\%$ Precision, $89.24 \pm 0.08\%$ Recall, $89.54 \pm 0.08\%$ F1-score, and $81.07 \pm 0.13\%$ Defect IoU on defective Campaign 5 images, with 4.34 M parameters, 14.8 G FLOPs, and 32.81 FPS under the reported desktop-GPU benchmark. On the 30 defect-free Campaign 5 images, the average false-positive area ratio was $0.42 \pm 0.03\%$, and the image-level false-alarm rate was $10.00 \pm 3.33\%$. The results indicate a balanced accuracy–complexity trade-off within the evaluated cross-time-campaign setting. However, strong light, low light, shadows, and reflections were not evaluated as independent subsets, so condition-specific robustness improvement cannot be quantified. Because all visible anomalies were merged into one binary defect class, the model localizes anomalous regions but does not classify cracks, pitting, honeycombing, or other defect types.

Keywords: wind turbine tower; surface defect segmentation; visible-light imaging; lightweight U-Net; parallel attention; defect-free samples; false-positive suppression

Academic Editor: Eugeniusz Koda

Received: 27 May 2026

Revised: 9 July 2026

Accepted: 14 July 2026

Published: 16 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Wind turbine towers are key load-bearing components, and their surface condition affects durability, inspection planning, and maintenance cost [1,2]. Long-term exposure to wind, temperature and humidity variation, ultraviolet radiation, precipitation, and operational vibration can produce visible anomalies such as cracks, pitting, honeycombing, local spalling, contaminant adhesion, and coating degradation [3]. Because early anomalies are often small, irregular, and visually subtle, repeatable surface inspection is important for monitoring deterioration and supporting maintenance decisions [4].

Current tower surface inspection still relies heavily on direct manual examination or manual interpretation of acquired images. These practices are labor-intensive and subjective, and they are difficult to standardize across towers, operators, and acquisition conditions. They can also involve safety risks when work at height is required [5]. In large wind farms, manual inspection provides limited support for frequent and fine-grained monitoring; therefore, automated vision methods are being investigated to improve inspection consistency and reduce manual workload [6,7].

Traditional image-processing methods usually rely on hand-crafted edges, thresholds, morphology, or texture descriptors [8]. Although such methods can be effective under controlled illumination and simple backgrounds, their performance is sensitive to viewing distance, curved-surface shading, reflections, structural joints, coating texture, and surrounding background. Their dependence on manually selected rules limits transferability when acquisition conditions or anomaly appearances change.

Deep convolutional networks and semantic segmentation have advanced crack detection, concrete surface damage recognition, and industrial defect analysis [9–12]. In contrast to image-level classification or bounding-box detection [13], semantic segmentation provides pixel-level location, shape, and boundary information [14–17]. U-Net and related encoder–decoder architectures are especially relevant because skip connections combine deep semantic information with shallow spatial detail [18,19].

Recent studies have extended structural inspection from binary crack recognition toward multi-class segmentation, quantitative measurement, and field-oriented acquisition. Wang et al. combined GAN-based augmentation with multi-class damage segmentation, geometric quantification, and severity assessment for reinforced-concrete structures [20]. Chen et al. developed an attention-enhanced dual-stage framework for tunnel-crack localization and quantitative measurement [21]. Other studies integrated UAV acquisition with multi-level damage classification [22] and examined classical and generative augmentation for multi-type reinforced-concrete damage segmentation under limited and imbalanced data [23]. Related work has also examined building surface crack segmentation, U-Net-based concrete-crack localization, and real-time crack detection [24–26], as well as encoder–decoder crack segmentation for concrete inspection [27]. These studies emphasize data diversity, category-aware evaluation, and downstream quantification. However, their targets, acquisition geometry, and deployment objectives differ from binary anomaly localization on curved wind turbine tower surfaces. The present study therefore focuses on a lightweight within-site segmentation framework and separately evaluates anomaly segmentation and false-positive behavior on defective and defect-free images.

Several task-specific challenges remain when established segmentation networks are applied to wind turbine tower images. Curved tower surfaces can produce nonuniform illumination and local reflections; structural joints and coating textures may resemble anomalies; and cracks, pitting, and honeycombing can be small, low-contrast, and irregular. Practical inspection also favors models with moderate computational cost. These characteristics motivate a problem-driven combination of lightweight feature extraction, attention-based feature recalibration, and multi-scale detail recovery. The contribution is

incremental and task-oriented: the individual operations are established, whereas their grouped configuration and evaluation are tailored to the present inspection setting. The overall research framework, including dataset construction, LPAU-Net training, and representative evaluation on defective and defect-free images, is shown in Figure 1.

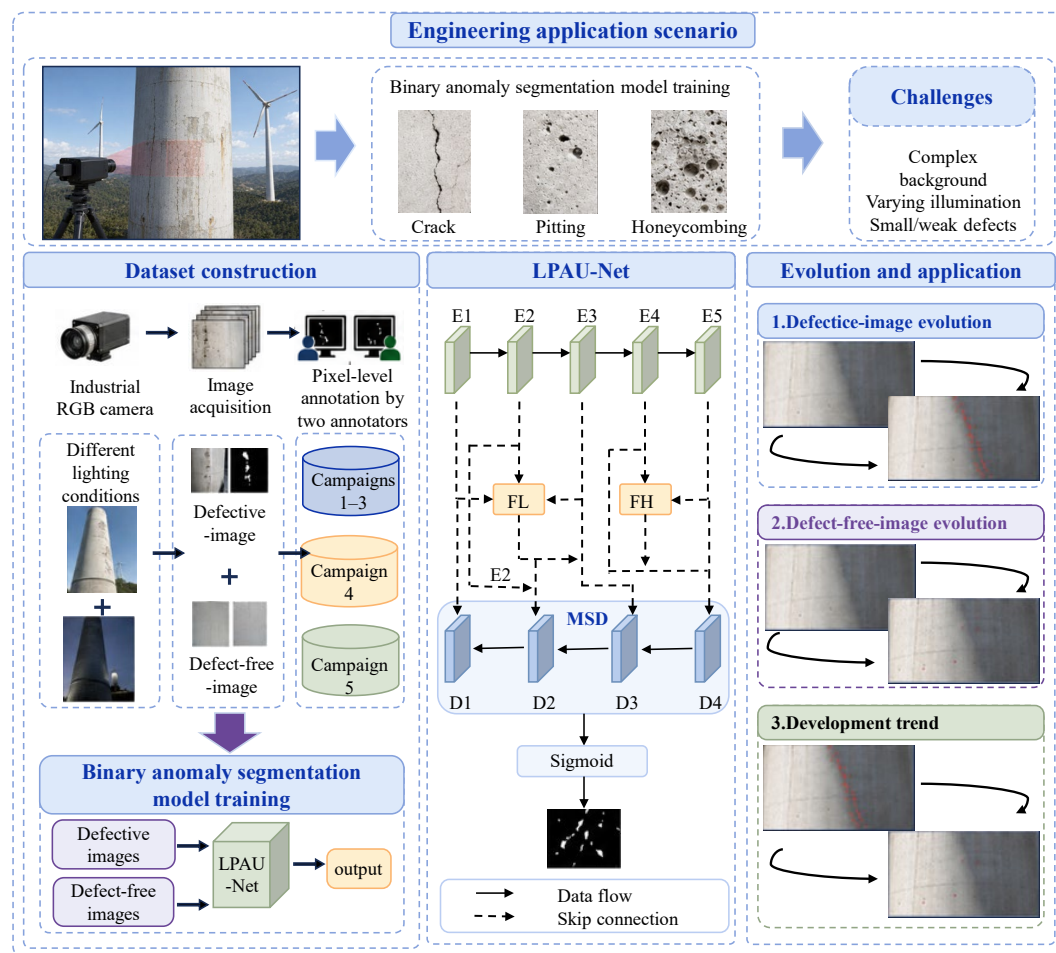


Figure 1. Overall research framework.

Figure 1 Overall research framework for surface-defect segmentation of wind turbine towers. The framework covers the engineering inspection scenario and associated challenges, image acquisition and pixel-level annotation, campaign-based dataset partitioning, joint model training using defective and defect-free images, the LPAU-Net architecture, and representative prediction results. Campaigns 1–3, Campaign 4, and Campaign 5 were used for model training, threshold selection, and held-out evaluation, respectively. The red overlays in the right-hand panels indicate pixels predicted as defects and illustrate defect localization, false-positive responses on defect-free images, and the intended application of the model to tower-surface inspection.

Accordingly, this study formulates tower surface inspection as binary pixel-level segmentation, in which visible cracks, pitting, honeycombing, and related anomalies are merged into one defect class. LPAU-Net combines DWConv–PWConv units for low-cost feature extraction, parallel channel–spatial attention for feature recalibration, grouped low- and high-level feature interactions, and multi-scale decoding for detail recovery. The objective is to localize whether and where a visible anomaly is present while maintaining a practical accuracy–complexity balance under the evaluated acquisition conditions. Because category-specific labels are not used, the model does not classify the detected region as a crack, pitting, honeycombing, or another defect type.

Field inspection streams also contain many images without visible defects. If training uses only defective samples, the model may respond to shadows, joints, reflections, and normal surface textures. Defect-free images are therefore included with all-background masks to provide explicit supervision for normal tower surfaces. This use of normal samples is consistent with anomaly-detection settings in which normal appearance must be modeled explicitly [28]. Section 4.5 evaluates both the benefit and the cost of this strategy: false-positive responses are reduced, while a small decrease in Recall shows that false-alarm suppression and sensitivity must be considered jointly.

The contributions are fourfold. First, established lightweight, attention, and multi-scale operations are configured in a task-oriented U-Net for binary anomaly localization in wind turbine tower imagery. Second, low- and high-level PAM branches use independent learnable scalar fusion to distinguish detail enhancement from semantic background suppression. Third, defective and defect-free images are jointly trained and evaluated through separate segmentation and false-alarm indicators. Fourth, the campaign-level protocol evaluates a later, time-separated acquisition batch from the same nine towers and wind farm. It therefore measures within-site cross-time-campaign performance and is not presented as unseen-tower, cross-wind-farm, or external validation.

2. Methodology

2.1. Problem Formulation and Overall Architecture

Field-acquired tower images combine small or weak-boundary anomalies with non-defective structures that may show similar contrast and texture. LPAU-Net is therefore designed to localize visible anomalies while controlling computational cost under the present desktop-GPU setting. Images were acquired using an MV-CS050-10GC 5 MP GigE industrial RGB camera (Hangzhou Hikrobot Co., Ltd., Hangzhou, China). Pixel-level annotations were manually prepared by two annotators using Labelme, version 5.4.1.

The task is defined as binary semantic segmentation. Cracks, pitting, honeycombing, and other visible surface anomalies are assigned to one defect class, and all remaining pixels are assigned to the background class. The output is an image-aligned anomaly-probability map rather than a category prediction. Consequently, the model can localize visible anomalous regions, but it cannot determine whether a region corresponds to a crack, pitting, honeycombing, or another defect type. It also does not infer defect mechanism, severity, or maintenance priority.

The encoder uses DWConv–PWConv units to reduce architecture-level complexity. PAM recalibrates features through parallel channel and spatial branches to reduce undesired responses to joints, shadows, reflections, and normal textures. Grouped multi-level fusion and the multi-scale decoder reuse shallow detail and deep semantic information to recover small or weak-boundary regions. Their individual and combined effects are evaluated through the controlled ablation configurations in Section 4.3.

Given an input image

$$X \in \mathbb{R}^{3 \times 512 \times 512} \quad (1)$$

the model outputs a single-channel defect probability map:

$$Y = f_{\theta}(X), \hat{Y} \in [0, 1]^{1 \times 512 \times 512} \quad (2)$$

where $f_{\theta}(\cdot)$ denotes the mapping implemented by LPAU-Net, θ represents the learnable parameters, and $\hat{Y}$ denotes the predicted probability that each pixel belongs to the defect class. During inference, the Sigmoid function is used to generate the defect probability map. The pixel-level binarization threshold was selected on the Campaign 4 validation set by maximizing Defect IoU on the defective validation images. The resulting

threshold was 0.5 and was fixed before evaluation on the Campaign 5 test set. The detailed threshold selection procedure and image-level decision strategy are described in Section 3.3.4.

$$M(x,y)=\begin{cases} 1, \hat{Y}(x,y) \geq 0.5 \\ 0, \hat{Y}(x,y) < 0.5 \end{cases} \quad (3)$$

The architecture follows three task-specific hypotheses. Replacing standard convolution with DWConv–PWConv units is expected to reduce computational cost while retaining local texture extraction; the A–B ablation comparison evaluates this trade-off. Parallel channel–spatial attention is intended to recalibrate anomaly-related responses and reduce visually similar background interference; this is examined by the B–C comparison and the learned-coefficient analysis. Grouped feature fusion with multi-scale decoding is intended to recover spatial detail lost during downsampling; this is examined by the B–D and C/D–E comparisons. These experiments evaluate the coordinated design, but they do not by themselves demonstrate external generalization or embedded-device readiness.

The binary formulation serves as a first-stage visible anomaly localization task. It identifies whether and where an anomaly is present in image coordinates, but it cannot classify the anomaly into crack, pitting, honeycombing, or other defect categories. It also does not infer physical dimensions or assign maintenance grades. Category classification, geometric measurement, and engineering grading require category-specific labels, geometric calibration, and additional decision criteria and are outside the present scope.

2.2. Lightweight Encoder with DWConv–PWConv Units

The encoder is designed to reduce computation without excessively weakening the representation of thin, low-contrast, or discontinuous anomalies. DWConv–PWConv units separate within-channel spatial filtering from cross-channel mixing, providing a lower-cost alternative to standard convolution while retaining hierarchical feature extraction. This factorization follows established depthwise-separable and mobile-network design principles [29–32].

The encoder contains five stages with progressively increasing channel depth and decreasing spatial resolution. Each stage applies two DW–PW units followed by 2×2 max pooling with stride 2. The resulting feature hierarchy provides skip features for the decoder and grouped inputs for the low- and high-level PAM branches. The complete LPAU-Net architecture is illustrated in Figure 2.

$$\text{DWConv} \rightarrow \text{PWConv} \rightarrow \text{LN_He} \rightarrow \text{Activation}$$

where DWConv denotes depthwise convolution for within-channel spatial filtering and PWConv denotes pointwise convolution for cross-channel interaction and channel adjustment. For compactness, LN_He denotes the implementation in which Layer Normalization is applied to the intermediate feature and convolutional weights are initialized using He initialization. GELU is used in the first three encoder stages, whereas H-Swish is used in the subsequent deeper encoder stages.

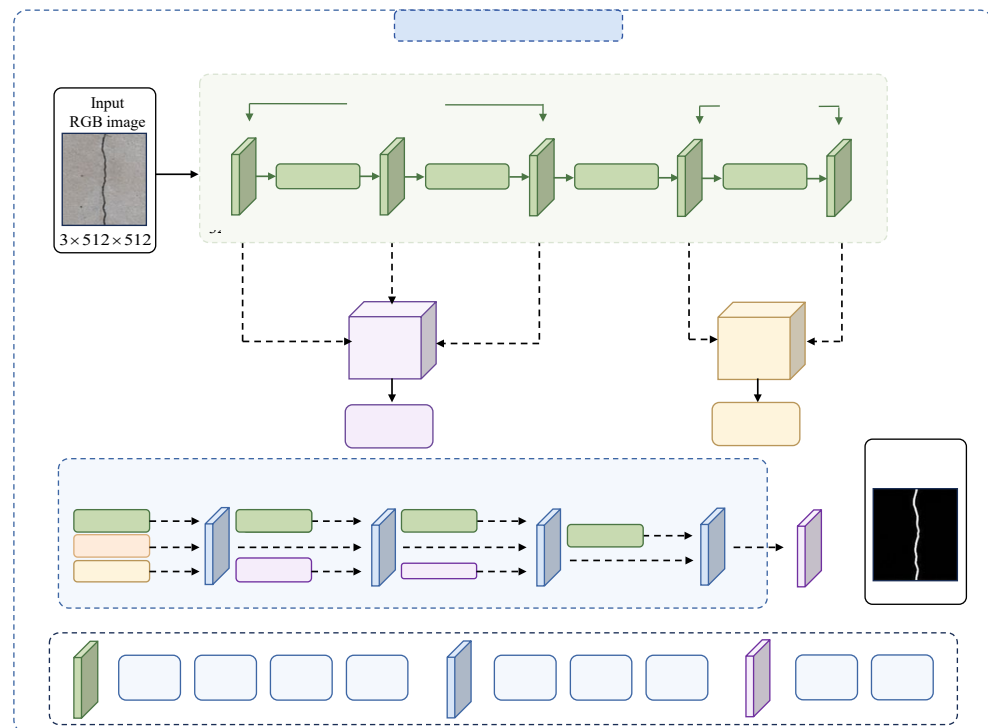


Figure 2. Overall architecture of LPAU-Net.

DW-PW factorization replaces dense spatial-channel convolution with separate spatial and channel operations. This design reduces architecture-level parameter and operation counts while retaining local edge and texture processing. Its practical effect is not assumed from the derivation alone; it is evaluated against the standard convolution baseline in the ablation study.

To further illustrate the lightweight mechanism of the DW-PW structure, let the input feature map be

$$X \in \mathbb{R}^{C \times H \times W} \quad (4)$$

where X denotes the input feature map; C , H , and W are the input channel count, feature-map height, and feature-map width; O is the number of output channels; and $k \times k$ is the depthwise convolution kernel size. For standard convolution, the parameter count and computational cost are expressed as follows:

$$P_{Conv} = COk^2 \quad (5)$$

$$F_{\text{Conv}} = CHW\sigma k^2 \quad (6)$$

In the DW-PW lightweight convolutional unit adopted in this study, depthwise convolution first performs spatial convolution separately on each input channel; therefore, its number of parameters and computational cost are

$$P_{DW} = Ck^2 \quad (7)$$

$$F_{DW} = CHWk^2 \quad (8)$$

Then, pointwise convolution maps the input channels to the output channels, and its number of parameters and computational cost are

$$P_{PW} = CO \quad (9)$$

$$F_{PW} = CHWO \quad (10)$$

Therefore, the total number of parameters and computational cost of the DW-PW lightweight convolutional unit can be expressed as

$$P_{DW-PW} = Ck^2 + CO = C(k^2 + O) \quad (11)$$

$$F_{DW-PW} = CHWk^2 + CHWO = CHW(k^2 + O) \quad (12)$$

Compared with standard convolution, the parameter ratio is

$$\frac{P_{DW-PW}}{P_{Conv}} = \frac{Ck^2 + CO}{COK^2} = \frac{1}{O} + \frac{1}{k^2} \quad (13)$$

The computational-cost ratio is

$$\frac{F_{DW-PW}}{F_{Conv}} = \frac{CHW(k^2 + O)}{CHWOK^2} = \frac{1}{O} + \frac{1}{k^2} \quad (14)$$

When depthwise convolution is adopted in this study, the corresponding relationship is

$$\frac{F_{DW-PW}}{F_{Conv}} = \frac{1}{O} + \frac{1}{9} \quad (15)$$

The derivation shows that DWConv-PWConv requires fewer parameters and operations than standard convolution for the same input and output dimensions. In the proposed encoder, this factorization provides the lightweight backbone, whereas the measured accuracy-efficiency trade-off is determined empirically in Section 4.3.

Stage-specific activation functions are used: GELU [33] in the first three stages and H-Swish in the last two stages. Layer Normalization is applied to intermediate features, and the convolutional weights are initialized using He initialization [34]. The compact notation LN_He is retained in the equations and figures to represent this implementation combination.

$$\text{GELU}(x) = x\Phi(x) \quad (16)$$

where Phi denotes the cumulative distribution function of the standard normal distribution. A commonly used approximation is

$$\text{GELU}(x) \approx 0.5x \left[1 + \tanh \left(\sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right) \right] \quad (17)$$

H-Swish is defined as

$$\text{H_Swish}(x) = x \frac{\text{ReLU6}(x+3)}{6} \quad (18)$$

$$\text{ReLU6}(x) = \min(\max(x, 0), 6) \quad (19)$$

Using the notation adopted in this manuscript, LN_He computes the mean and variance over the channel and spatial dimensions of each sample:

$$\mu = \frac{1}{CHW} \sum_{c=1}^C \sum_{h=1}^H \sum_{w=1}^W X_{c,h,w} \quad (20)$$

$$\sigma^2 = \frac{1}{CHW} \sum_{c=1}^C \sum_{h=1}^H \sum_{w=1}^W (X_{c,h,w} - \mu)^2 \quad (21)$$

The normalized output is

$$\text{LN}_{\text{He}}(X_{c,h,w}) = \gamma \frac{X_{c,h,w} - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \quad (22)$$

where γ and β denote the learnable scale and shift parameters, respectively; epsilon prevents division by zero.

2.3. Learnable Weighted Parallel-Attention Module

Tower joints, shadow boundaries, local reflections, and coating textures may produce feature responses similar to cracks or pitting. Without explicit feature recalibration, a feature extractor can preserve both true anomalies and non-defective interference, which may increase false positives or fragment weak-boundary regions.

PAM addresses this issue through parallel channel and spatial branches. Channel attention reweights feature channels associated with anomaly discrimination, while spatial attention emphasizes candidate locations. Channel recalibration is related to squeeze-and-excitation mechanisms [35], whereas the parallel channel-spatial formulation follows the attention design cited in [36]. Their outputs are combined with the original feature through a residual path. The structure of the learnable weighted PAM is shown in Figure 3.

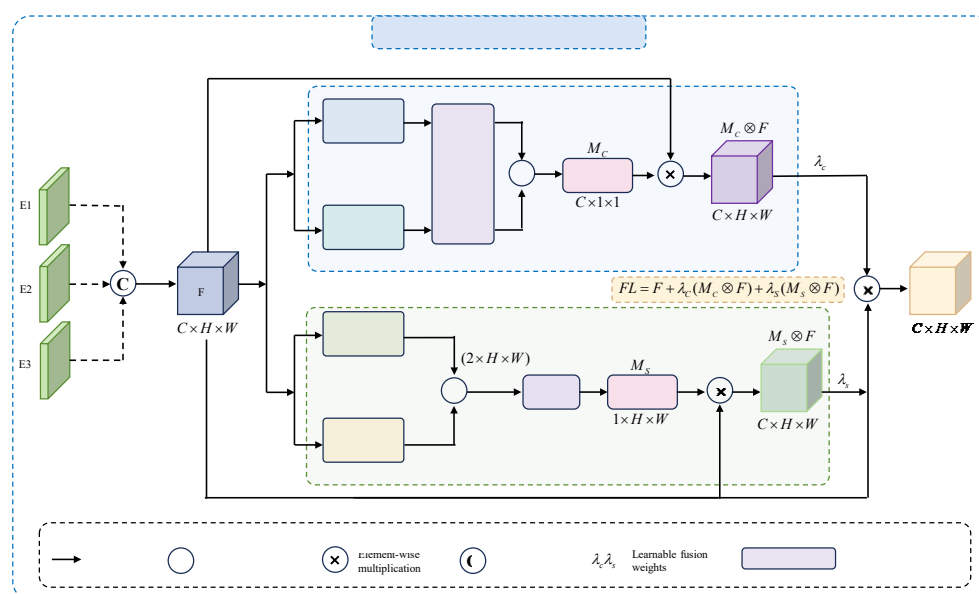


Figure 3. Structure of the learnable weighted PAM illustrated with the low-level fusion feature FL.

PAM contains parallel channel- and spatial-attention branches. The branches act on the same input feature and are fused with the residual feature [37], so branch reweighting can be learned without removing the original information path.

Given an input feature

$$F \in \mathbb{R}^{C \times H \times W} \quad (23)$$

for the channel attention module, global average pooling is defined as

$$F_{\text{avg}}^c = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W F_{c,h,w} \quad (24)$$

Global max pooling is defined as

$$F_{\text{max}}^c = \max_{1 \leq h \leq H, 1 \leq w \leq W} F_{c,h,w} \quad (25)$$

The channel attention weight is

$$M_c = \sigma(g_c(F_{avg}^c) + g_c(F_{max}^c)) \quad (26)$$

where $g_c(\cdot)$ denotes the channel mapping function and $\sigma(\cdot)$ denotes the Sigmoid function, respectively.

For the spatial attention module, channel-wise average pooling is defined as

$$F_{avg}^s(h, w) = \frac{1}{C} \sum_{c=1}^C F_{c,h,w} \quad (27)$$

Channel-wise max pooling is defined as

$$F_{max}^s(h, w) = \max_{1 \leq c \leq C} F_{c,h,w} \quad (28)$$

The spatial attention weight is

$$M_s = \sigma(g_s(\text{Concat}(F_{avg}^s, F_{max}^s))) \quad (29)$$

where M_s denotes the spatial mapping function.

The channel-attention-enhanced feature and the spatial-attention-enhanced feature are defined as

$$F_c = M_c \otimes F \quad (30)$$

$$F_s = M_s \otimes F \quad (31)$$

Fixed additive fusion implicitly assigns equal contributions to the residual, channel-attention, and spatial-attention terms. Because the relative usefulness of semantic channels and spatial locations may vary with feature level and imaging condition, equal weighting is not assumed to be optimal.

Learnable scalar weights are therefore introduced at the PAM fusion stage. They are optimized jointly with the remaining network parameters, allowing the relative channel and spatial contributions to adapt to the training objective. The improved PAM output is defined as

$$F_{out} = F + \lambda_c F_c + \lambda_s F_s \quad (32)$$

where λ_c and λ_s denote the fusion weights of the channel- and spatial-attention branches, respectively. To retain the original fixed fusion form at initialization while allowing later adaptation, they are defined as

$$\lambda_c = 2\sigma(a_c) \quad (33)$$

$$\lambda_s = 2\sigma(a_s) \quad (34)$$

where a_c and a_s are learnable scalar parameters initialized to 0. Since

$$\sigma(0) = 0.5 \quad (35)$$

at the initial training stage

$$\lambda_c = \lambda_s = 1 \quad (36)$$

At this point, the fusion form of the improved PAM is equivalent to the original fixed additive fusion:

$$F_{out} = F + F_c + F_s \quad (37)$$

The learnable scalar parameters are updated jointly with the remaining network weights through backpropagation of the segmentation loss:

$$a_c^{t+1} = a_c^t - \eta \frac{\partial L}{\partial a_c} \quad (38)$$

$$a_s^{t+1} = a_s^t - \eta \frac{\partial L}{\partial a_s} \quad (39)$$

where η denotes the learning rate and L denotes the total loss function. The channel and spatial fusion weights are therefore optimized by the same objective as the remaining network parameters rather than fixed manually.

Finally, the improved PAM output can be written as

$$F_{out} = F + 2\sigma(a_c)(M_c \otimes F) + 2\sigma(a_s)(M_s \otimes F) \quad (40)$$

This formulation retains the residual path and adds only four scalar parameters across the low- and high-level PAM modules. Its value is assessed by the fixed-versus-learnable comparison in Section 4.3. The learned coefficients describe branch weighting within this dataset and model; they should not be interpreted as universal or as a substitute for spatial feature visualization.

2.4. Grouped Multi-Scale PAM Feature Fusion

Shallow encoder features preserve edges, texture, and local shape, whereas deeper features provide stronger semantic discrimination. Applying attention at only one resolution may therefore favor either detail recovery or background suppression. The encoder outputs are consequently divided into low- and high-level groups, each with an independent PAM.

The low-level group contains E1, E2, and E3. Their spatial resolutions are aligned to that of E3, concatenated along the channel dimension, compressed by a 1×1 convolution, and passed to the low-level PAM:

$$F_L = \text{PAM}_L(\text{Conv}_{1 \times 1}(\text{Concat}(\text{Down}(E_1), \text{Down}(E_2), E_3))) \quad (41)$$

The output size is

$$F_L \in \mathbb{R}^{128 \times 64 \times 64} \quad (42)$$

The low-level branch is intended to retain edge, texture, and local morphology cues that are relevant to fine or weak-boundary anomalies. Independent channel and spatial scalars allow the branch balance to be learned rather than fixed.

$$F_L^{out} = F_L + 2\sigma(a_c^L)(M_c^L \otimes F_L) + 2\sigma(a_s^L)(M_s^L \otimes F_L) \quad (43)$$

where a_c^L and a_s^L denote the learnable scalar parameters of the channel- and spatial-attention branches in the low-level PAM; both are initialized to 0. The corresponding fusion weights are

$$\lambda_c^L = 2\sigma(a_c^L) \quad (44)$$

$$\lambda_s^L = 2\sigma(a_s^L) \quad (45)$$

At the initial training stage,

$$\lambda_c^L = \lambda_s^L = 1 \quad (46)$$

During optimization, the low-level fusion weights are adjusted jointly with the remaining parameters according to the same segmentation objective.

The high-level group contains E4 and E5. E5 is upsampled to the spatial resolution of E4, after which the features are concatenated, compressed by a 1×1 convolution, and processed by the high-level PAM:

$$F_H = \text{PAM}_H(\text{Conv}_{1 \times 1}(\text{Concat}(E_4, \text{Up}(E_5)))) \quad (47)$$

The output size is

$$F_H \in \mathbb{R}^{256 \times 32 \times 32} \quad (48)$$

The high-level branch is intended to strengthen semantic discrimination between visible anomalies and non-defective interference such as joints, shadows, and reflections. It uses independent channel and spatial scalar parameters.

$$F_H^{\text{out}} = F_H + 2\sigma(a_c^H)(M_c^H \otimes F_H) + 2\sigma(a_s^H)(M_s^H \otimes F_H) \quad (49)$$

where a_c^H and a_s^H denote the learnable scalar parameters of the channel- and spatial-attention branches in the high-level PAM; both are initialized to 0. The corresponding fusion weights are

$$\lambda_c^H = 2\sigma(a_c^H) \quad (50)$$

$$\lambda_s^H = 2\sigma(a_s^H) \quad (51)$$

At the initial training stage,

$$\lambda_c^H = \lambda_s^H = 1 \quad (52)$$

The high-level fusion weights are likewise optimized jointly with the remaining network parameters.

Accordingly, independent learnable fusion parameters are set in the low-level PAM and high-level PAM:

$$a_c^L, a_s^L, a_c^H, a_s^H \quad (53)$$

Only four trainable scalar parameters are added across the two PAM modules. This permits different low- and high-level branch balances without materially changing model size; the learned patterns are reported descriptively in Section 4.3.

The PAM-enhanced low- and high-level features are subsequently reused by the decoder. For concise notation, they retain the low- and high-level fusion symbols in the following equations.

2.5. Multi-Scale Decoder for Boundary Recovery

Repeated downsampling strengthens semantic representation but reduces spatial resolution, which can fragment thin cracks and blur pitting or honeycombing boundaries. The multi-scale decoder therefore combines conventional skip features with the PAM-enhanced low- and high-level features. This design is intended to restore spatial detail while preserving semantic context; its contribution is evaluated separately in the ablation study.

The decoding process of MSD is expressed as

$$D_4 = \text{DSConv}(\text{Conv}_{1 \times 1}(\text{Concat}(\text{Up}(E_5), E_4, F_H))) \quad (54)$$

$$D_3 = \text{DSConv}(\text{Conv}_{1 \times 1}(\text{Concat}(\text{Up}(D_4), E_3, F_L))) \quad (55)$$

$$D_2 = \text{DSConv}(\text{Conv}_{1 \times 1}(\text{Concat}(\text{Up}(D_3), E_2, \text{Up}(F_L)))) \quad (56)$$

$$D_1 = \text{DSConv}(\text{Conv}_{1 \times 1}(\text{Concat}(\text{Up}(D_2), E_1))) \quad (57)$$

where $\text{Up}(\cdot)$ denotes bilinear upsampling, $\text{Concat}(\cdot)$ denotes channel-wise concatenation, $\text{Conv}_{1 \times 1}(\cdot)$ denotes channel compression and feature reorganization, and $\text{DSConv}(\cdot)$ denotes lightweight feature fusion. The feature sizes of the encoder, grouped attention features, and decoder stages are shown in Table 1.

Table 1. Feature sizes of the encoder, grouped PAM fusion, and decoder.

Stage	Source	Processing Operation	Output	Purpose
E1	Input	Encoder module 1	$32 \times 256 \times 256$	Skip connection to D1; low-level feature fusion
E2	E1	Encoder module 2	$64 \times 128 \times 128$	Skip connection to D2; low-level feature fusion
E3	E2	Encoder module 3	$128 \times 64 \times 64$	Skip connection to D3; low-level feature fusion
E4	E3	Encoder module 4	$256 \times 32 \times 32$	Skip connection to D4; high-level feature fusion
E5	E4	Encoder module 5	$512 \times 16 \times 16$	Bottleneck feature; high-level feature fusion
FL	E1, E2, E3	Size alignment + concatenation + 1×1 Conv + PAM	$128 \times 64 \times 64$	Used in decoder stages D3 and D2
FH	E4, E5	Size alignment + concatenation + 1×1 Conv + PAM	$256 \times 32 \times 32$	Used in decoder stage D4
D4	Concat(Up(E5)), E4, FH	Concatenation + 1×1 Conv + DW-PW	$256 \times 32 \times 32$	Deep-detail recovery
D3	Up(D4), E3, FL	Concatenation + 1×1 Conv + DW-PW	$128 \times 64 \times 64$	Medium-scale defect recovery
D2	Up(D3), E2, Up(FL)	Concatenation + 1×1 Conv + DW-PW	$64 \times 128 \times 128$	Shallow-detail refinement
D1	Up(D2), E1	Concatenation + 1×1 Conv + DW-PW	$32 \times 256 \times 256$	Boundary reconstruction
Output	D1	Upsampling + 1×1 Conv + Sigmoid	$1 \times 512 \times 512$	Defect probability map

E_i denotes the encoder feature at stage i , D_i denotes the decoder feature at stage i , and FL and FH denote the low- and high-level PAM-enhanced fusion features, respectively. All feature sizes are expressed as $C \times H \times W$.

2.6. Output Layer

Because the final decoder feature has a spatial resolution of 256×256 , the output layer further uses bilinear upsampling to restore it to the input image size and generates a single-channel prediction map through a 1×1 convolution:

$$Y = \text{Conv}_{1 \times 1}(\text{Up}(D_1)) \quad (58)$$

The defect probability map is then obtained through the Sigmoid function:

$$\hat{Y} = \text{Sigmoid}(Y) \quad (59)$$

The final output is

$$\hat{Y} \in \mathbb{R}^{1 \times 512 \times 512} \quad (60)$$

2.7. Joint Loss Function

Defect pixels occupy a small fraction of most images, creating substantial foreground–background imbalance. The focal weighting principle was introduced to reduce the influence of easy examples [38]. In this study, the training objective combines BCE, Dice Loss [39], and Focal Tversky Loss, with the latter selected because it can place additional emphasis on hard or missed foreground regions under class imbalance [40,41].

$$L=0.3L_{\text{BCE}}+0.4L_{\text{Dice}}+0.3L_{\text{FTL}} \quad (61)$$

where L_{BCE} , L_{Dice} and L_{FTL} denote the BCE, Dice, and Focal Tversky terms, respectively. They constrain pixel classification, region overlap, and false-negative-sensitive learning. The joint loss acts only during optimization and does not change the inference graph, parameter count, or forward computational path.

BCE Loss is defined as

$$L_{\text{BCE}}=-\frac{1}{N}\sum_{i=1}^N[y_i\log(\hat{y}_i)+(1-y_i)\log(1-\hat{y}_i)] \quad (62)$$

where N denotes the total number of pixels, y_i denotes the ground-truth label of the corresponding pixel, and $\hat{y}_i$ denotes the predicted probability for that pixel, respectively.

Dice Loss is defined as

$$\text{Dice}=\frac{2\sum_{i=1}^Ny_i\hat{y}_i+\varepsilon}{\sum_{i=1}^Ny_i+\sum_{i=1}^N\hat{y}_i+\varepsilon} \quad (63)$$

$$L_{\text{Dice}}=1-\text{Dice} \quad (64)$$

Focal Tversky Loss is defined as [41]

$$\text{TI}=\frac{\text{TP}+\varepsilon}{\text{TP}+\alpha\text{FP}+\beta\text{FN}+\varepsilon} \quad (65)$$

$$L_{\text{FTL}}=(1-\text{TI})^\gamma \quad (66)$$

where TP, FP, and FN denote the true-positive, false-positive, and false-negative pixel counts, respectively. For defect-free training images, the ground-truth mask is empty. The smoothing terms keep Dice and Focal Tversky losses numerically stable when both prediction and target approach zero; nonzero defect responses increase the false-positive contribution and are penalized. The parameters used in this study are defined below:

$$\alpha=0.3,\beta=0.7,\gamma=0.75 \quad (67)$$

The relative weights of BCE, Dice, and Focal Tversky Loss were selected through a limited Campaign 4 sensitivity analysis. Three predefined combinations were evaluated with all other settings fixed, and 0.3/0.4/0.3 produced the highest validation F1-score and Defect IoU among those candidates. This is a local comparison rather than an exhaustive hyperparameter search.

The Focal Tversky parameters were fixed at $\alpha = 0.3$, $\beta = 0.7$, and $\gamma = 0.75$ for all experiments. They were not exhaustively optimized. Under the adopted formulation, the larger false-negative weight increases the penalty assigned to missed defect pixels.

3. Dataset and Experimental Protocol

3.1. Wind Turbine Tower Surface Defect Dataset

This study constructed a wind turbine tower surface defect dataset for training and evaluating LPAU-Net. The dataset comprises 762 independently acquired visible-light images collected from nine physical wind turbine towers within one wind farm during five separately organized field acquisition campaigns conducted at different time periods. Only RGB visible-light information was used; auxiliary sensing modalities such as depth imaging, infrared imaging, laser ranging, or positioning measurements were not involved.

During image acquisition, the camera-to-surface distance was approximately 2–5 m and was adjusted according to tower surface location, site accessibility, and field conditions rather than fixed at a constant value. The imaging viewpoint was also not rigidly standardized because accessible shooting positions varied among surface locations. The original image resolution was not treated as an experimental control variable. Before network input, all images and corresponding annotation masks were uniformly resized to 512×512 pixels.

No tower-region cropping, manual background removal, or foreground masking was performed before resizing. Thus, the network inputs retained the complete captured field of view, including tower surface textures, structural joints, shadows, illumination transitions, local reflections, and surrounding background interference. Image acquisition was conducted under natural outdoor conditions and included normal illumination, strong illumination, low illumination, shadowed regions, and local reflections. These non-standardized conditions were retained to reflect common visual interference during field inspection. The dataset acquisition and coverage information is summarized in Table 2.

Table 2. Dataset acquisition and coverage summary.

Acquisition Item	Dataset Information
Image source	Field-acquired visible-light images of wind turbine towers
Acquisition device	RGB visible-light camera; no auxiliary depth, infrared, laser, or positioning modality
Independent original images	762 images: 610 defective and 152 defect-free
Input resolution	All images resized to 512×512 before network input
Wind farms	One wind farm
Wind turbine towers	Nine physical towers (T1–T9)
Independent inspection campaigns	Five time-separated acquisition campaigns (C1–C5)
Documented illumination/background conditions	Natural outdoor illumination, shadows, local reflections, structural joints, and retained background interference

The coverage summary reports only verified acquisition information. The original image resolution was not treated as an experimental control variable, and augmented derivatives are not counted as independently acquired field samples.

Of the 762 original images, 610 contain visible anomalies and 152 are defect-free, corresponding to a defect-free proportion of 19.95%. The training, validation, and test subsets contain 92, 30, and 30 defect-free images, respectively. Defective masks merge cracks, pitting, honeycombing, and related visible anomalies into one defect class, whereas defect-free masks contain background only. Two annotators divided the initial pixel-level labeling work and then cross-reviewed each other's masks for omitted regions, boundary errors, and background interference. Disagreements were jointly re-examined and resolved to obtain consensus masks. Because the independent pre-consensus masks were not retained as a separate evaluation set, inter-annotator agreement cannot be quantified retro-

spectively; annotation uncertainty at weak or ambiguous boundaries should therefore be recognized as a limitation.

Images were retained only when the tower surface and anomaly boundaries were sufficiently visible for pixel-level annotation; severely blurred, substantially occluded, or indeterminate samples were excluded. The final dataset comprises one wind farm, the same nine physical towers, and five time-separated acquisition campaigns. Campaigns 1–3 contain 141, 168, and 149 images and form the 458-image training set. Campaign 4 contains 152 images for validation, checkpoint selection, and threshold determination. Campaign 5 contains 152 images acquired during a later time-separated campaign from the same nine towers and the same wind farm, and it is used only for final evaluation. Excluding severely degraded images improves annotation reliability but narrows the evaluated operating range.

All subsets contain Towers T1–T9, but each time-separated acquisition campaign is assigned exclusively to one subset. Photographed surface locations were generally different across campaigns, which reduces but does not eliminate the possibility that the same physical defect location appears in more than one subset. No augmented derivative is included in validation or testing. The protocol therefore measures within-site cross-time-campaign performance on the same physical towers and wind farm. It is not an unseen-tower split, cross-wind-farm test, or external validation of new coatings or imaging devices. The subset composition and training augmentation counts are summarized in Table 3.

Table 3. Original field images and training augmentation.

Subset	Acquisition Campaign	Number of Towers Represented	Original Images	Defect-Free Originals	Augmentation	Total Inputs	Defect-Free Inputs
Training set	C1–C3	T1–T9	458	92	One additional image per image	916	184
Validation set	C4	T1–T9	152	30	No augmentation	152	30
Test set	C5 (later time-separated campaign)	T1–T9 (same towers)	152	30	No augmentation	152	30
Total	C1–C5	T1–T9	762	152	Training set only	1220	244

The dataset contains 762 independently acquired field images. The total of 1220 denotes the effective experimental inputs after one offline derivative was generated for each of the 458 training images; it must not be interpreted as 1220 independent field samples.

3.2. Data Augmentation Strategy

The limited number of independent field images and the small anomaly-area ratio increase the risk of overfitting to the available acquisition conditions. Data augmentation is therefore used as a training-time regularization strategy [42,43]. It is applied only to the training set, whereas validation and test images remain unaugmented. Albumentations [44] applies each geometric transformation synchronously to the image and pixel-level mask. General evidence on augmentation under limited image data is also provided in [45].

All images and masks are first resized to 512×512 pixels. Offline augmentation then applies geometric transformations, brightness/contrast perturbation, and noise according to Table 4. One randomly generated derivative is created for each of the 458 original training images, including 366 defective and 92 defect-free images. The effective training set therefore contains 916 inputs: 732 defective and 184 defect-free. These derivatives are used only for optimization and are not treated as additional independent observations.

Table 4. Data augmentation parameters.

Augmentation operation	Parameter Setting	Probability
Resize	512 × 512	-
Horizontal flip	-	0.5
Random rotation	±15°	0.5
Brightness/contrast perturbation	0.2	0.5
Gaussian noise	Noise intensity range 10–50	0.3

All transformations are restricted to the training set and are applied synchronously to each image–mask pair to preserve pixel-level alignment.

3.3. Experimental Setup and Evaluation Protocol

3.3.1. Training Configuration, Implementation Environment, and Inference Benchmark

All training, ablation, and comparison experiments used the same campaign-level split, 512 × 512 input size, and offline-augmented training set. Each model was trained for 150 epochs with a batch size of 4 using AdamW [46]. The initial learning rate and weight-decay coefficient were both set to 1×10^{-4} . Three independent runs used random seeds; each seed controlled initialization, data shuffling, and other stochastic training operations, while the split and remaining hyperparameters were unchanged. For each run, the checkpoint with the highest Defect IoU on Campaign 4 was retained; if several checkpoints achieved the same Defect IoU, the checkpoint with the lower validation loss was selected.

Experiments were conducted on Ubuntu 20.04 with an Intel Core i7-10700K CPU, an NVIDIA GeForce RTX 3060 GPU with 12 GB memory, and 32 GB system memory. The implementation used Python 3.8, PyTorch 1.13.1 [47], CUDA 11.7, cuDNN 8.6, OpenCV 4.7, and Albumentations 1.3.0. MultiStepLR reduced the learning rate by a factor of 0.5 at epochs 60, 100, and 130 to reduce late-stage parameter fluctuations. The schedule is expressed below.

Let the initial learning rate and the learning rate at epoch e be denoted by the symbols in the following expression:

$$\eta_t = \eta_0 \times \gamma^{m(t)} \quad (68)$$

where γ denotes the learning rate decay factor. The exponent counts the decay milestones passed by the current epoch. The learning rate is defined piecewise as follows:

$$\eta_t = \begin{cases} 1.0 \times 10^{-4}, & 0 \leq t < 60 \\ 5.0 \times 10^{-5}, & 60 \leq t < 100 \\ 2.5 \times 10^{-5}, & 100 \leq t < 130 \\ 1.25 \times 10^{-5}, & 130 \leq t \leq 150 \end{cases} \quad (69)$$

Figure 4 summarizes the learning rate schedule and a representative training tendency. Unless otherwise stated, numerical segmentation results in the result tables are reported as mean ± sample standard deviation over the three independent random-seed runs. In contrast, convergence curves, threshold-sensitivity examples, and qualitative prediction figures are representative seed-1 results used for illustration and threshold discussion, not three-run averages. Parameter count and FLOPs are architecture-level quantities. All FPS values were obtained under the same RTX 3060, 512 × 512, PyTorch evaluation environment and are used only for within-study comparison; image acquisition, annotation, and result visualization are outside this benchmark. Because no embedded or UAV-mounted platform was tested, the reported desktop-GPU throughput does not establish field-deployment performance.

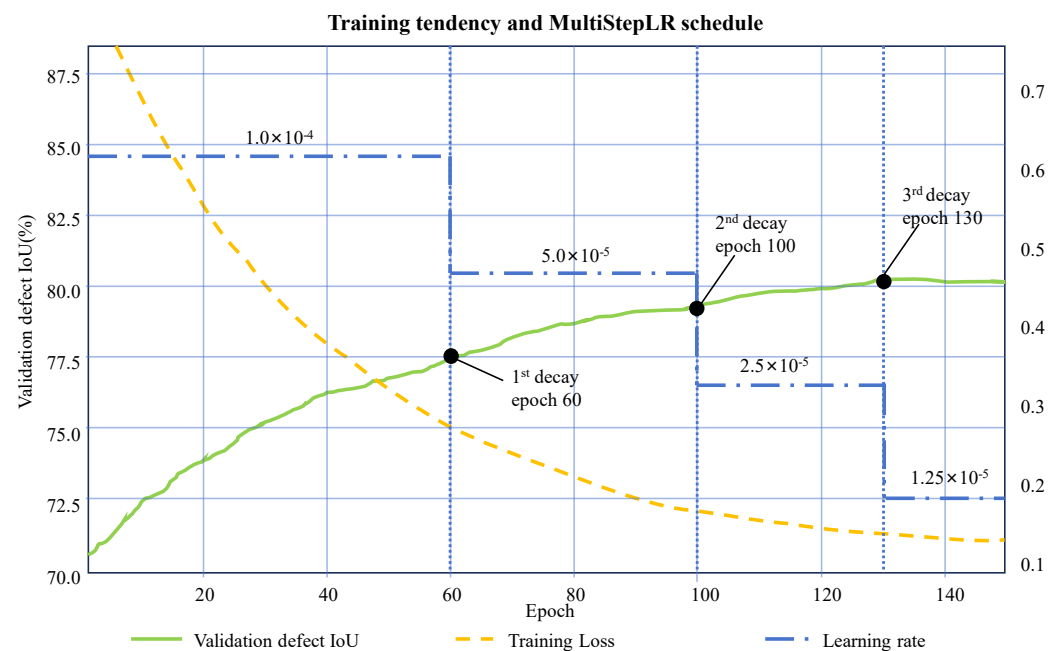


Figure 4. Training tendency and MultiStepLR schedule.

3.3.2. Joint Training with Defective and Defect-Free Images

Defective and defect-free images are jointly optimized in a single LPAU-Net rather than in separate models. Their network processing is identical; only the ground-truth mask differs.

For defective training images, the ground-truth mask consists of defect regions and background regions:

$$Y(x,y)=\begin{cases} 1, & (x,y) \in \Omega_{\text{def}} \\ 0, & (x,y) \in \Omega_{\text{bg}} \end{cases} \quad (70)$$

where Ω_{def} denotes the visible surface defect regions, such as cracks, pitting, and honeycombing, and Ω_{bg} denotes the non-defective background regions. Defective images therefore supervise anomaly morphology, boundaries, and spatial distribution.

For defect-free training images, the ground-truth mask is entirely background:

$$Y(x,y)=0, \forall (x,y) \quad (71)$$

Defect-free images provide explicit supervision for normal tower surfaces, including textures, joints, shadows, reflections, and background regions. A nonzero predicted probability in these images contributes false-positive loss, whereas an output close to zero reduces the loss. Their purpose is therefore to constrain spurious responses rather than to provide defect-shape information.

Given an input image, LPAU-Net outputs a continuous defect probability map:

$$\hat{Y}=f_{\theta}(X), \quad \hat{Y} \in [0,1]^{1 \times H \times W} \quad (72)$$

During training, the continuous probability map and the ground-truth mask are used directly in the loss calculation. Binarization is applied only during inference and evaluation.

Taking BCE Loss as an example,

$$L_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1-y_i) \log(1-\hat{y}_i)] \quad (73)$$

For defect-free images,

$$y_i = 0 \quad (74)$$

Therefore, BCE Loss can be written as

$$L_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N \log(1-\hat{y}_i) \quad (75)$$

For defect-free images, a high predicted probability in a normal region increases the loss and encourages the model to suppress that response. The effect on defect sensitivity is not assumed to be uniformly beneficial; Section 4.5 evaluates the trade-off between defective-image Recall and defect-free-image false alarms.

3.3.3. Evaluation Metrics and Protocol

Evaluation is separated by ground-truth image type. Defective images contain positive pixels and are used to calculate Precision, Recall, F1-score, and Defect IoU. Defect-free images contain empty masks, for which per-image F1-score and IoU can be undefined or misleading; they are therefore evaluated using predicted false-positive area and image-level FAR. The definitions follow standard Precision–Recall and F-measure evaluation conventions [48].

The calculation formulas are as follows:

$$\text{Precision} = \frac{TP}{TP+FP+\epsilon} \quad (76)$$

$$\text{Recall} = \frac{TP}{TP+FN+\epsilon} \quad (77)$$

$$F1 = \frac{2\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall} + \epsilon} \quad (78)$$

$$\text{Dice} = \frac{2TP}{2TP+FP+FN+\epsilon} \quad (79)$$

$$\text{IoU}_{\text{defect}} = \frac{TP}{TP+FP+FN+\epsilon} \quad (80)$$

TP, FP, and FN are accumulated over the defective-image subset. The reported Precision, Recall, F1-score, and Defect IoU therefore describe segmentation on images containing visible anomalies rather than performance on the combined test set.

For defect-free images, the ground-truth mask satisfies

$$Y(x,y)=0, \quad \forall(x,y) \quad (81)$$

Defect-free images are excluded from F1-score and Defect IoU. Any predicted defect pixel in this subset is a false positive, and the average false-positive area ratio is calculated as follows:

$$A_{\text{FP}}^{\text{normal}} = \frac{1}{K} \sum_{j=1}^K \frac{\sum_{x=1}^H \sum_{y=1}^W M_j(x,y)}{H \times W} \times 100\% \quad (82)$$

where K denotes the number of defect-free test images and $M_j(x,y)$ denotes the binarized prediction at pixel (x,y) of defect-free image k ($1 = \text{defect}$ and $0 = \text{background}$). Lower values indicate less predicted defect area on normal tower surfaces.

The evaluation protocol therefore reports anomaly segmentation and false-alarm control separately. Unless explicitly stated as a representative seed-1 analysis, the values in quantitative tables are reported as mean $\pm$ sample standard deviation over the three runs. These repeated runs describe between-run variability but do not constitute a formal statistical significance test.

3.3.4. Validation-Based Model Selection and Threshold Determination

Campaign 4 was used for checkpoint selection and threshold determination without contributing to parameter updates. The pixel threshold was selected on the 122 defective validation images by maximizing Defect IoU. With that threshold fixed, the predicted defect-area ratio was calculated for all 152 validation images, and the image-level area threshold was selected by maximizing the Youden index. Both thresholds were fixed before Campaign 5 evaluation.

After the pixel-level threshold was fixed at 0.5, the predicted defect-area ratio was calculated for all 152 images in the Campaign 4 validation set, including both defective and defect-free images. For the j -th validation image, the predicted defect-area ratio was defined as

$$R_j = \frac{\sum_{x=1}^H \sum_{y=1}^W M_j(x,y)}{H \times W} \quad (83)$$

where $M_j(x,y) \in \{0,1\}$ denotes the binary prediction obtained using the pixel-level threshold of 0.5.

The image-level prediction was defined as

$$\hat{Z}_j = \begin{cases} 1, & R_j \geq \tau_a \\ 0, & R_j < \tau_a \end{cases} \quad (84)$$

where $\hat{Z}_j = 1$ indicates that the image was classified as containing visible defects and $\hat{Z}_j = 0$ indicates that no visible defect was detected.

Five predefined image-level area thresholds (0.30%, 0.50%, 0.67%, 0.80%, and 1.00%) were evaluated on all 152 Campaign 4 images. The threshold with the maximum Youden index was retained; ties were resolved in favor of the smaller threshold to reduce missed-detection risk. This procedure selected 0.67%, equivalent to approximately 1756 predicted pixels in a 512×512 image. The pixel threshold of 0.5 and area threshold of 0.67% were then applied unchanged to Campaign 5.

$$J = \text{Sensitivity} + \text{Specificity} - 1 \quad (85)$$

The selected thresholds define the validation-based operating point used throughout the final evaluation. They should not be interpreted as universal thresholds for other sites, devices, defect scales, or illumination distributions. The image-level area-threshold sensitivity results are presented in Table 5.

Table 5. Image-level area-threshold sensitivity on Campaign 4.

Area Threshold	Sensitivity	Specificity	FAR	Youden Index
0.30%	98.36% (120/122)	80.00% (24/30)	20.00% (6/30)	0.7836
0.50%	96.72% (118/122)	86.67% (26/30)	13.33% (4/30)	0.8339
0.67%	95.90% (117/122)	90.00% (27/30)	10.00% (3/30)	0.8590
0.80%	90.98% (111/122)	93.33% (28/30)	6.67% (2/30)	0.8432
1.00%	87.70% (107/122)	96.67% (29/30)	3.33% (1/30)	0.8437

Sensitivity was calculated from the 122 defective Campaign 4 images, whereas specificity and FAR were calculated from the 30 defect-free Campaign 4 images. This threshold-sensitivity analysis was conducted using the representative seed-1 validation predictions rather than the three-run mean. FAR equals 1 minus specificity, and the 0.67% operating point was selected by the maximum Youden index.

Lower area thresholds increase sensitivity by accepting smaller predicted regions but also increase false alarms. Higher thresholds reduce false alarms while increasing the risk of rejecting small true anomalies. The 0.67% operating point provided the highest Youden index on Campaign 4 and was therefore fixed before Campaign 5 was evaluated.

The main architecture, original ablation plan, comparison models, training settings, and evaluation rules were specified using Campaigns 1–4 before Campaign 5 was evaluated. Revision-stage sensitivity analyses were restricted to Campaign 4 and were not used to optimize results on Campaign 5.

4. Results

The final evaluation uses Campaign 5, a later time-separated acquisition batch that did not participate in parameter updating, checkpoint selection, or threshold determination. Campaign 5 contains the same nine physical towers and the same wind farm as the earlier campaigns; it is therefore not an unseen-tower or cross-wind-farm test. Defective images are evaluated using Precision, Recall, F1-score, and Defect IoU, whereas defect-free images are evaluated using false-positive indicators.

Table 6 summarizes the complete model over three seeds on the held-out Campaign 5 subset. The segmentation metrics are calculated from defective images only, whereas the false-positive results for defect-free images and the effects of the defect-free training strategy are discussed separately in later sections.

Table 6. Final test performance of LPAU-Net over three random seeds.

Model	Test Precision	Test Recall	Test F1-Score	Test Defect IoU	Params/M	FLOPs/G	FPS
LPAU-Net	89.84 ± 0.08%	89.24 ± 0.08%	89.54 ± 0.08%	81.07 ± 0.13%	4.34	14.8	32.81

Precision, Recall, F1-score, and Defect IoU refer to defective Campaign 5 images and are reported as mean ± sample standard deviation over three random seeds. FPS refers to the common desktop-GPU benchmark described in Section 3.3.1 and is not evidence of embedded-device performance.

Figure 5 presents validation Precision, Recall, and F1-score from the representative seed-1 run to illustrate convergence behavior. It is not an additional independent experiment and is not a three-run average.

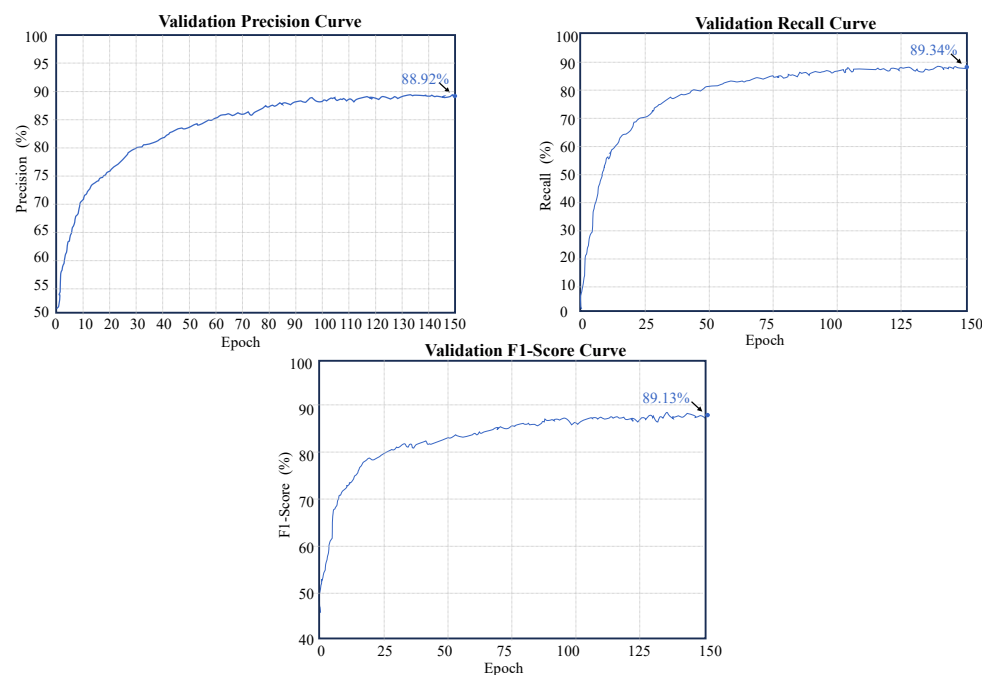


Figure 5. Convergence of validation Precision, Recall, and F1-score for the representative seed-1 run.

On defective Campaign 5 images, the reported mean Precision and Recall differ by 0.60 percentage points, indicating a relatively balanced pixel-level operating point under the selected threshold. The model has 4.34 M parameters, 14.8 G reported operations, and 32.81 FPS under the common desktop-GPU benchmark. These values support computational feasibility within the tested environment but do not establish embedded-device performance. The pixel-level threshold sensitivity on Campaign 4 is reported in Table 7.

For the representative seed-1 model, lower pixel thresholds increased Recall and false-positive pixels, whereas higher thresholds increased Precision and missed more defect pixels. A threshold of 0.5 produced the highest Campaign 4 Defect IoU among the evaluated candidates and was fixed before Campaign 5 evaluation. The final model performance reported in Table 6 is based on the mean $\pm$ sample standard deviation across three runs rather than on the threshold-sensitivity run alone.

Table 7. Pixel-level threshold sensitivity on the defective Campaign 4 validation subset.

Pixel Threshold	Precision	Recall	F1-Score	Defect IoU
0.30	84.63%	92.90%	88.57%	79.48%
0.40	87.41%	90.19%	88.78%	79.82%
0.50	88.92%	89.34%	89.13%	80.39%
0.60	91.72%	85.37%	88.43%	79.26%
0.70	94.36%	81.03%	87.19%	77.29%

All values were calculated on the 122 defective Campaign 4 images using the representative seed-1 validation predictions. They explain threshold selection and are not three-run averages. Campaign 5 was not used for threshold selection.

4.1. Validation-Based Threshold Selection

Tables 5 and 7 summarize the image-level and pixel-level threshold analyses. The selected pixel threshold of 0.5 and area threshold of 0.67% were determined exclusively from Campaign 4 and then applied unchanged to Campaign 5.

4.2. Final Test Performance Under Cross-Time Campaign Separation

Campaigns 1–3, Campaign 4, and Campaign 5 were used for training, validation, and final evaluation, respectively. Campaign 5 was collected at a later time but contains the same nine physical towers from the same wind farm. This section therefore evaluates variation across time-separated acquisition campaigns within one site; it does not evaluate generalization to unseen tower bodies, a different wind farm, or an independent external dataset. Representative-run performance across the training, validation, and Campaign 5 subsets is compared in Table 8.

Table 8. Representative-run performance across the training, validation, and fifth-campaign test subsets.

Dataset	Precision	Recall	F1-Score	Defect IoU	Description
Training set	90.35%	91.07%	90.71%	83.00%	Campaigns 1–3; used for parameter optimization
Validation set	88.92%	89.34%	89.13%	80.39%	Campaign 4; used for model and threshold selection
Test set	89.78%	89.17%	89.47%	80.95%	Campaign 5; later time-separated acquisition batch from the same T1–T9 towers and wind farm

Training metrics were computed in inference mode on the 458 original, non-augmented training images. Validation and Campaign 5 metrics were computed on their corresponding original defective images. All values are from the representative seed-1 run and should not be interpreted as three-run averages.

For the representative seed-1 run, Campaign 5 F1-score and Defect IoU were 1.17 and 1.93 percentage points below the original training subset, while validation and Campaign 5 values were numerically close. This single-run observation is used only to illustrate the behavior of one selected checkpoint; the final reported performance is the three-seed mean $\pm$ sample standard deviation. The observation should not be interpreted as proof of external generalization.

The same nine tower bodies appear in all campaigns, and occasional overlap in photographed surface locations cannot be completely excluded. The results should therefore be interpreted only as within-site cross-time-campaign performance. Evidence for broader generalization requires tower-level hold-out testing and independently acquired data from other wind farms, coatings, environmental conditions, or imaging devices. The representative seed-1 training and validation curves are shown in Figure 6. The false-positive analysis on defect-free Campaign 5 images is reported in Table 9.

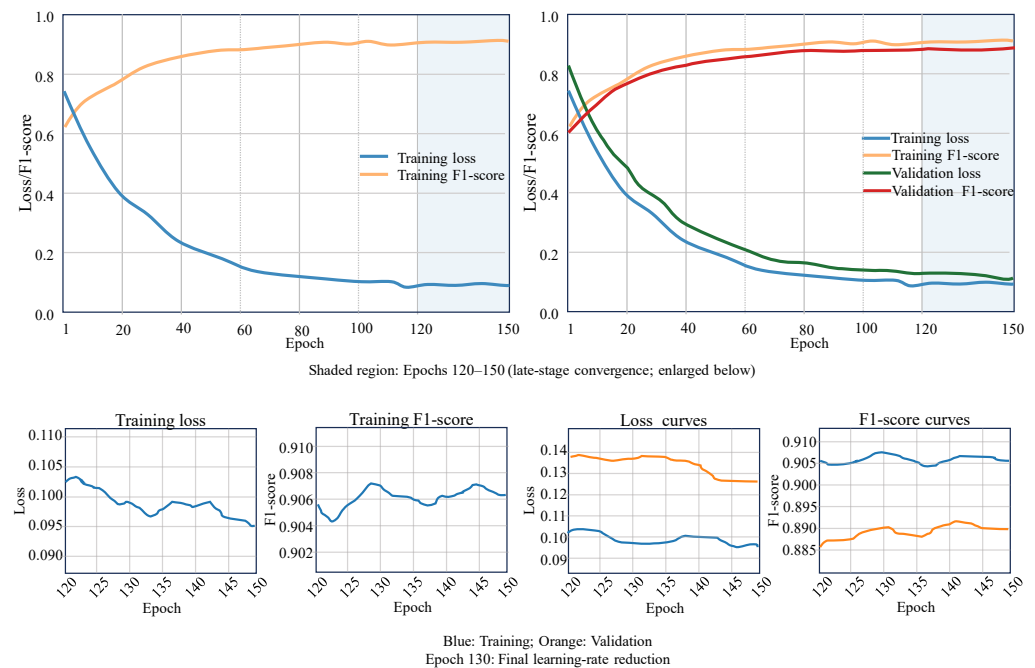


Figure 6. Training and validation curves for the representative seed-1 run.

Figure 6 Training and validation curves for the representative seed-1 run. In the full-training plots, the blue and orange curves represent the training loss and training F1-score, respectively, whereas the green and red curves represent the validation loss and validation F1-score, respectively. The light-blue shaded region indicates epochs 120–150, which are enlarged in the lower panels. This interval represents the late stage of training and covers the period immediately before and after the final learning-rate reduction at epoch 130, thereby allowing the convergence stability and the differences between the training and validation metrics to be examined. In the enlarged panels, the blue and orange curves represent the training and validation metrics, respectively.

Table 9. False-positive analysis of the final LPAU-Net on defect-free Campaign 5 images.

Number of Defect-Free Test Images	Average False-Positive Area Ratio	Median False-Positive Area Ratio	Maximum False-Positive Area Ratio	Image-Level FAR	Typical False-Positive Source
30	$0.42 \pm 0.03\%$	$0.28 \pm 0.02\%$	$3.26 \pm 0.18\%$	$10.00 \pm 3.33\%$	Shadows, tower joints, and local reflections

Image-level FAR uses the validation-selected pixel threshold of 0.5 and area threshold of 0.67%. Values are summarized over the three independently trained models. Across the three runs, 2, 4, and 3 of the 30 defect-free Campaign 5 images exceeded the area threshold, giving a mean FAR of $10.00 \pm 3.33\%$.

The mean, median, and maximum false-positive area ratios quantify different aspects of residual response on defect-free images. The relatively low mean and median coexist with a larger maximum, indicating that a small number of images can still produce extensive false-positive regions.

The remaining errors were associated mainly with strong shadow boundaries, structural joints, and local reflections. Because the defect-free test subset contains only 30 images, changing the outcome of one or two images materially changes FAR. The reported value is therefore an initial estimate under the present acquisition conditions rather than a stable population-level false-alarm rate.

At the selected operating point, approximately one in ten defect-free images would require unnecessary manual review. A higher area threshold could reduce this workload but would increase the risk of rejecting small true anomalies. The threshold should therefore be adapted and revalidated before use in a different inspection workflow.

4.3. Ablation Study

Seven ablation configurations were used to separate architecture-level and training-stage effects. Group A is the standard U-Net baseline. Group B replaces standard convolution with DSConv. Groups C and D add PAM or MSD to Group B, respectively. Group E combines PAM and MSD, Group F adds data augmentation, and Group G adds the FTL term to form the complete LPAU-Net. The ablation design is summarized in Table 10, and the corresponding three-seed results are reported in Table 11.

Table 10. Ablation experiment design.

Group	DSConv	PAM	MSD	Data Augmentation	FTL Term	Experimental Purpose
A	×	×	×	×	×	Original baseline model
B	√	×	×	×	×	Verify lightweight effect
C	√	√	×	×	×	Verify the effect of PAM on defect enhancement
D	√	×	√	×	×	Verify boundary recovery by multi-scale decoding
E	√	√	√	×	×	Verify combined effect of PAM and MSD
F	√	√	√	√	×	Verify the effect of data augmentation strategy
G	√	√	√	√	√	Complete LPAU-Net

A check mark indicates inclusion and a cross indicates removal. DSConv, PAM, and MSD alter the inference architecture; data augmentation and FTL affect training only.

Table 11. Ablation experiment results over three random seeds.

Group	Precision	Recall	F1-Score	Defect IoU	Params/M	FLOPs/G	FPS
A	83.69 ± 0.09%	84.13 ± 0.09%	83.91 ± 0.09%	72.28 ± 0.13%	7.70	25.4	25.00
B	82.51 ± 0.10%	83.19 ± 0.09%	82.85 ± 0.09%	70.72 ± 0.14%	3.80	12.7	36.00
C	85.31 ± 0.09%	87.50 ± 0.08%	86.39 ± 0.08%	76.04 ± 0.13%	3.90	13.9	34.84
D	84.59 ± 0.09%	85.89 ± 0.08%	85.24 ± 0.08%	74.27 ± 0.13%	4.00	14.3	33.20
E	86.67 ± 0.09%	89.63 ± 0.09%	88.13 ± 0.09%	78.77 ± 0.15%	4.34	14.8	32.81
F	88.63 ± 0.09%	88.77 ± 0.09%	88.70 ± 0.09%	79.70 ± 0.14%	4.34	14.8	32.81
G	89.84 ± 0.08%	89.24 ± 0.08%	89.54 ± 0.08%	81.07 ± 0.13%	4.34	14.8	32.81

Segmentation metrics are calculated on defective Campaign 5 images and reported as mean ± sample standard deviation over three random seeds. Parameter count, FLOPs, and FPS describe the inference architecture, so training-only changes do not alter these quantities.

Replacing standard convolution with DSConv reduced parameters from 7.70 M to 3.80 M and increased FPS from 25.00 to 36.00, while F1-score and Defect IoU decreased modestly. This result establishes the main efficiency–accuracy trade-off of the lightweight encoder rather than an unconditional improvement.

Adding PAM to the lightweight backbone increased mean F1-score and Defect IoU by 3.54 and 5.32 percentage points. The numerical improvement is consistent with better feature selection under background interference, although the aggregate metrics alone do not show where attention was spatially concentrated.

Adding MSD to the same backbone increased mean F1-score and Defect IoU by 2.39 and 3.55 percentage points. This result is consistent with improved reuse of multi-scale spatial detail, but boundary-specific and defect-size-stratified metrics were not calculated.

Groups C and D provide separate numerical evidence for PAM and MSD. PAM produced the larger aggregate gain in the present dataset, while MSD also improved overlap. Their intended roles remain mechanistic interpretations supported by the architecture and ablation pattern, not direct causal proof from feature visualization.

Combining PAM and MSD in Group E outperformed either module alone. Data augmentation in Group F shifted the operating point toward higher Precision and slightly lower Recall while modestly increasing F1-score and Defect IoU. These results suggest complementary effects, but they remain specific to the present split and training protocol.

Adding FTL in Group G further increased the reported aggregate metrics relative to Group F. Because no defect-size-stratified analysis or formal paired significance test was performed, the result supports a numerical improvement in overall overlap but does not establish that the gain is concentrated on small defects or statistically significant at the image level.

DSCnv provides the largest reduction in parameter count and reported operations. PAM and MSD add limited architecture-level cost, while augmentation and FTL leave the inference graph unchanged. All efficiency comparisons are restricted to the common desktop-GPU benchmark.

Overall, the ablation pattern is consistent with the intended problem-driven roles of the components: DSCnv controls complexity, PAM improves aggregate discrimination under complex backgrounds, MSD improves multi-scale overlap, and FTL changes the training objective. These interpretations are bounded by the absence of condition-stratified robustness tests and direct feature-map visualization.

To describe the learned PAM behavior across repeated training, the final scalar fusion coefficients were extracted from the three seed-specific checkpoints selected on Campaign 4 and summarized in Table 12.

Table 12. Learned PAM fusion coefficients and normalized branch contributions over three random seeds.

PAM Module	Channel-Attention Fusion Coefficient	Spatial-Attention Fusion Coefficient	Normalized Channel Contribution Weight	Normalized Spatial Contribution Weight
Low-level PAM	0.764 ± 0.022	1.013 ± 0.015	0.430 ± 0.011	0.570 ± 0.011
High-level PAM	1.086 ± 0.025	0.695 ± 0.012	0.610 ± 0.010	0.390 ± 0.010

The reported channel- and spatial-attention fusion coefficients are the absolute learned values (λ_c and λ_s). The normalized channel and spatial contribution weights are auxiliary ratios obtained by dividing each coefficient by their sum. All values are mean $\pm$ sample standard deviation over three random seeds and do not localize attention in image space.

Across the three runs, the low-level PAM assigned the larger normalized weight to the spatial branch, whereas the high-level PAM assigned the larger weight to the channel branch. This pattern is compatible with the intended distinction between shallow spatial detail and deeper semantic discrimination, but it is descriptive and does not replace feature-map or attention-response visualization. The validation comparison between fixed and learnable PAM fusion is reported in Table 13.

Table 13. Validation comparison between fixed and learnable PAM fusion on Campaign 4.

PAM Fusion	Precision	Recall	F1-Score	Defect IoU	Params/M
Fixed fusion	88.68 ± 0.07%	88.95 ± 0.07%	88.81 ± 0.07%	79.88 ± 0.11%	4.340000
Learnable fusion	88.96 ± 0.07%	89.31 ± 0.07%	89.13 ± 0.07%	80.40 ± 0.11%	4.340004

Fixed and learnable PAM fusion were trained with the same three seeds and compared on the 122 defective Campaign 4 images. Campaign 5 was not used to choose the fusion strategy.

Learnable fusion produced modest mean gains in validation F1-score and Defect IoU while adding only four scalar parameters. With three runs and no paired image-level significance test, this result is reported as a repeatable numerical advantage rather than formal statistical evidence.

The low-level spatial emphasis and high-level channel emphasis were consistent across the three runs. The coefficients remain dataset- and model-specific and should not be interpreted as universally optimal or as evidence of causal attention behavior. The validation effect of adding FTL to the BCE+Dice loss is reported in Table 14.

Table 14. Validation effect of adding FTL to the BCE+Dice loss.

Loss Configuration	Validation Precision	Validation Recall	Validation F1-Score	Validation Defect IoU
BCE+Dice	88.59 ± 0.07%	88.82 ± 0.09%	88.70 ± 0.08%	79.70 ± 0.13%
BCE+Dice+FTL	88.96 ± 0.07%	89.31 ± 0.07%	89.13 ± 0.07%	80.40 ± 0.11%

Values are reported as mean ± sample standard deviation over three random seeds on the 122 defective Campaign 4 images. FTL denotes Focal Tversky Loss; Campaign 5 was not used to select the loss configuration.

Adding FTL to BCE+Dice produced small numerical increases in all reported Campaign 4 validation metrics. These results support its retention in the tested configuration but do not establish statistical significance. The sensitivity of the joint-loss weights is reported in Table 15.

Table 15. Sensitivity to the joint loss weights on Campaign 4

BCE Weight	Dice Weight	FTL Weight	Validation Precision	Validation Recall	Validation F1	Validation Defect IoU
0.4	0.3	0.3	88.52 ± 0.06%	88.85 ± 0.08%	88.68 ± 0.07%	79.66 ± 0.11%
0.3	0.4	0.3	88.96 ± 0.07%	89.31 ± 0.07%	89.13 ± 0.07%	80.40 ± 0.11%
0.3	0.3	0.4	88.72 ± 0.07%	89.00 ± 0.07%	88.86 ± 0.07%	79.96 ± 0.11%

The three loss weights sum to 1. Values are reported as mean ± sample standard deviation over three random seeds on the 122 defective Campaign 4 images; the analysis represents a local sensitivity check around the adopted weight combination.

Among the three nearby weight combinations, 0.3/0.4/0.3 produced the highest mean Campaign 4 validation F1-score and Defect IoU and was retained for final evaluation. This is a limited local sensitivity analysis rather than a global optimization of loss weights.

4.4. Comparison with Representative Segmentation Models

U-Net++, Attention U-Net, DeepLabV3+, MobileNetV3-U-Net, and Fast-SCNN were selected to cover dense skip connections, attention-enhanced decoding, deep semantic segmentation, lightweight backbones, and mobile-oriented segmentation. Each model used the same campaign split, 512 × 512 input, metric definitions, and three random seeds. MobileNetV3-U-Net and Fast-SCNN were introduced primarily as lightweight efficiency references in the quantitative comparison; therefore, their role is evaluated in Table 16

through mean metrics, parameters, FLOPs, and FPS, rather than being required in the qualitative visual panel. Representative seed-1 predictions from the selected models are compared qualitatively in Figure 7.

Table 16. Comparative experiment results over three random seeds.

Model	Precision	Recall	F1-Score	Defect IoU	Params/M	FLOPs/G	FPS
U-Net++	$84.19 \pm 0.12\%$	$77.75 \pm 0.13\%$	$80.84 \pm 0.13\%$	$67.84 \pm 0.18\%$	26.90	48.3	30.60
Attention U-Net	$85.53 \pm 0.11\%$	$87.42 \pm 0.10\%$	$86.46 \pm 0.10\%$	$76.15 \pm 0.16\%$	34.00	81.5	17.00
DeepLabV3+	$77.27 \pm 0.12\%$	$83.56 \pm 0.15\%$	$80.29 \pm 0.13\%$	$67.07 \pm 0.18\%$	59.00	82.9	30.20
MobileNetV3-U-Net	$87.11 \pm 0.09\%$	$86.40 \pm 0.09\%$	$86.75 \pm 0.09\%$	$76.61 \pm 0.14\%$	3.18	2.46	49.62
Fast-SCNN	$83.85 \pm 0.11\%$	$84.73 \pm 0.11\%$	$84.29 \pm 0.11\%$	$72.84 \pm 0.16\%$	1.11	1.36	63.85
LPAU-Net	$89.84 \pm 0.08\%$	$89.24 \pm 0.08\%$	$89.54 \pm 0.08\%$	$81.07 \pm 0.13\%$	4.34	14.8	32.81

All comparison models used the same dataset split, input size, evaluation rules, and desktop-GPU environment. Segmentation metrics are reported as mean $\pm$ sample standard deviation over three seeds. Parameter count and FLOPs are architecture-dependent, and FPS is intended only for relative comparison under the common benchmark. MobileNetV3-U-Net and Fast-SCNN are included to assess lightweight efficiency; their omission from Figure 7 does not affect the quantitative comparison.

LPAU-Net achieved the highest mean segmentation metrics among the selected reference models. This comparison establishes competitiveness within the evaluated set, while the controlled ablations provide more direct evidence for the contribution of individual LPAU-Net components. Figure 7 should be read as a representative visual illustration from the seed-1 predictions only, whereas Table 16 provides the three-run quantitative basis for comparison.

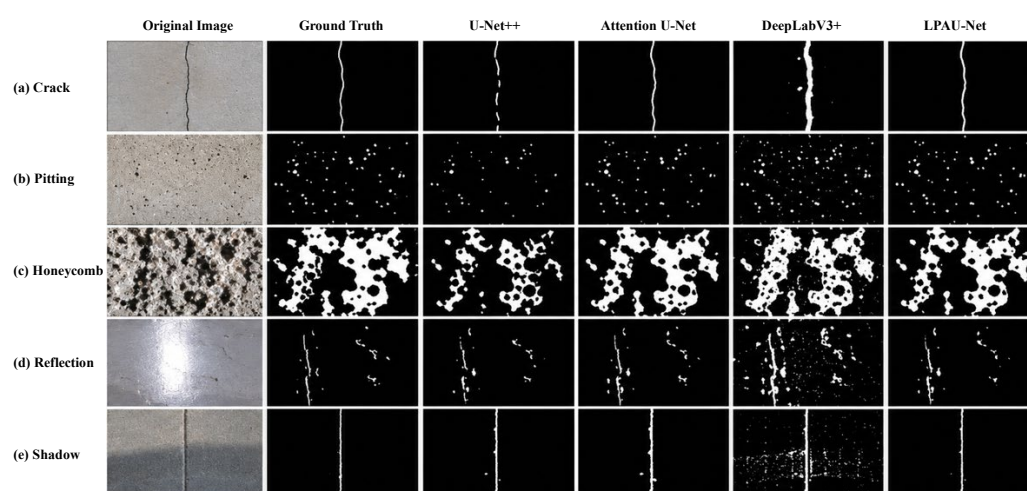


Figure 7. Visual comparison of representative seed-1 predictions from selected models. The figure is qualitative and is not a three-run average; the complete quantitative comparison, including MobileNetV3-U-Net and Fast-SCNN, is reported in Table 16.

LPAU-Net is substantially smaller than U-Net++, Attention U-Net, and DeepLabV3+, but it is larger and computationally heavier than MobileNetV3-U-Net and Fast-SCNN. Its FPS is also below the two dedicated lightweight baselines. The proposed model should therefore be interpreted as a higher-accuracy lightweight operating point, not the minimum-cost or universally fastest architecture.

The combined results indicate a balanced accuracy–complexity trade-off under the reported RTX 3060 environment. Because no embedded processor, UAV platform, or in-

dustrial edge device was evaluated, the study demonstrates desktop-GPU computational efficiency rather than verified field-deployment readiness.

4.5. Effect of Defect-Free Training Images on Segmentation Sensitivity and False-Positive Suppression

Two otherwise identical three-seed configurations were trained to isolate the effect of defect-free images. The only change was whether the 92 defect-free training images and their augmented derivatives were included. Both configurations used the same fixed pixel and area thresholds and were evaluated separately on 122 defective and 30 defect-free Campaign 5 images. The resulting segmentation and false-positive metrics are compared in Table 17.

Table 17. Effect of defect-free training data on segmentation and false-positive suppression over three random seeds.

Model Training Data	Precision	Recall	F1-Score	Defect IoU	Average False-Positive Area Ratio	Median False-Positive Area Ratio	Maximum False-Positive Area Ratio	FAR
Without defect-free data	$87.23 \pm 0.12\%$	$89.56 \pm 0.09\%$	$88.38 \pm 0.11\%$	$79.18 \pm 0.17\%$	$0.85 \pm 0.03\%$	$0.59 \pm 0.03\%$	$5.61 \pm 0.18\%$	$16.67 \pm 3.34\%$
With defect-free data	$89.84 \pm 0.08\%$	$89.24 \pm 0.08\%$	$89.54 \pm 0.08\%$	$81.07 \pm 0.13\%$	$0.42 \pm 0.03\%$	$0.28 \pm 0.02\%$	$3.26 \pm 0.18\%$	$10.00 \pm 3.33\%$

Segmentation metrics in Table 17 are calculated on defective images, whereas false-positive area ratios and FAR are calculated on defect-free images. Values are mean $\pm$ sample standard deviation over three runs and should be distinguished from the representative qualitative examples in Figure 8.

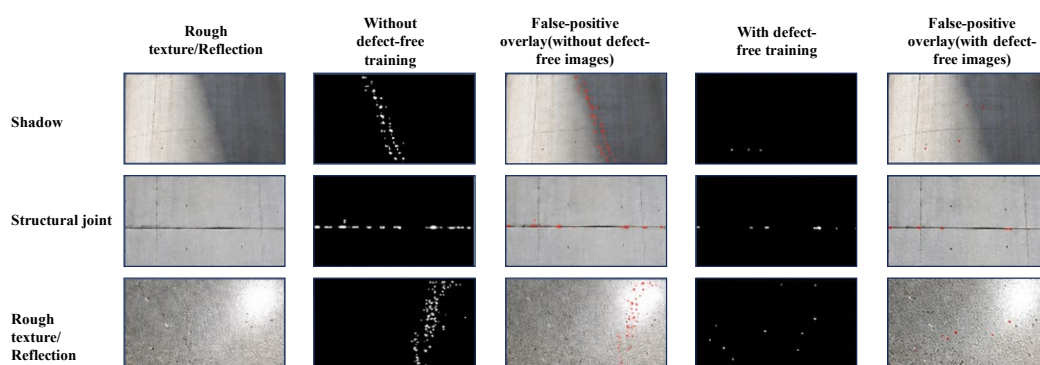


Figure 8. Representative seed-1 qualitative examples showing the effect of defect-free samples on false-positive suppression. The quantitative effect over three random seeds is reported in Table 17.

Including defect-free training images increased Precision, F1-score, and Defect IoU and reduced the average false-positive area ratio and image-level FAR, while Recall decreased slightly. The comparison therefore shows a shift in the operating balance rather than an improvement in every metric.

The observed trade-off is important for safety-oriented use: suppressing nuisance responses reduces unnecessary manual verification, but it may increase missed-defect risk. The preferred operating point should therefore be selected according to application-specific costs and validated on a larger independent defect-free set.

4.6. Qualitative Analysis of Segmentation and False-Positive Suppression

Figure 8 provides representative seed-1 qualitative examples of defective and defect-free images containing background textures, shadows, structural joints, and local reflections. The figure supplements the aggregate metrics, but it is not a three-run average and is not a dedicated condition-stratified robustness experiment.

The examples show that many predicted regions follow the annotated anomaly morphology and that defect-free training reduces some nuisance responses. Residual false positives remain near sharp shadow boundaries, structural joints, and local reflections, whereas very small, low-contrast, or weak-boundary regions may be missed or incompletely segmented. Although the dataset contains strong-light, low-light, shadow, reflection, and joint interference, Campaign 5 was not prospectively stratified into sufficiently sized independent subsets for these conditions. The aggregate results therefore represent performance under the mixed conditions present in the later Campaign 5 batch. They cannot quantify a one-to-one robustness improvement for strong light, shadows, reflections, or any other single interference factor.

The qualitative outputs were prepared for representative comparison rather than through a prospectively defined failure-case sampling protocol. A dedicated panel separating true-positive, false-positive, and false-negative pixels was not constructed. Consequently, the present observations identify plausible failure modes but do not quantify their frequency, severity, or probability under a specific interference condition.

Cracks, pitting, honeycombing, and related visible anomalies were merged into one binary defect class. Consequently, the network output indicates anomalous versus background pixels only and cannot classify the detected anomaly type. Category-specific Precision, Recall, F1-score, and IoU are therefore unavailable, and the aggregate metrics may conceal differences among defect morphologies and boundary contrasts. Future datasets should retain category labels and sufficient samples for multi-class evaluation.

The masks provide anomaly location, shape, and relative image-space area. Without geometric scale calibration and tower-coordinate registration, they cannot be converted directly into physical length, width, area, orientation, tower location, severity, or maintenance grade. They are therefore inspection cues for review rather than direct structural-condition measurements.

Acquisition distance and viewpoint were not standardized, and original image resolution was not analyzed as an independent factor. Uniform resizing to 512×512 does not eliminate variation in defect pixel scale or possible loss of fine detail. Controlled acquisition experiments are needed to quantify sensitivity to distance, resolution, blur, occlusion, and defect scale.

4.7. Limitations and Engineering Applicability

The current evidence is bounded by one wind farm, the same nine recurring physical towers, 762 original images, possible overlap of photographed defect locations, and the exclusion of severely blurred or occluded samples. Campaign-level separation prevents direct mixing of acquisition times, but Campaign 5 remains a later batch from the same towers and site and therefore does not provide unseen-tower, cross-wind-farm, or external validation. The 30-image defect-free test subset also yields an imprecise FAR estimate. Consequently, all conclusions are restricted to within-site cross-time-campaign performance under the evaluated acquisition conditions.

Several limitations remain in the present evaluation. First, strong light, low light, shadows, structural joints, and reflections were not independently stratified, so the aggregate metrics do not establish condition-specific robustness and cannot quantify how much robustness changed for any single interference condition. Second, PAM interpretation is limited to learned scalar coefficients and controlled ablation comparisons; no feature-map

or attention-response visualization was retained. Third, mean and sample standard deviation across three random seeds characterize between-run variability but do not constitute formal statistical significance testing. Fourth, the qualitative analysis did not use a prospectively defined failure-case sampling procedure or complete TP/FP/FN error maps. Finally, the binary labels merge all anomaly morphologies, so the model cannot classify defect types. These limitations require independently labeled condition subsets, intermediate-feature visualization, image-level paired statistical inference, systematic failure analysis, and category-specific annotations in future work.

5. Conclusions

This study presents LPAU-Net as a task-oriented lightweight framework for first-stage localization of visible surface anomalies on wind turbine towers. The individual convolution, attention, decoding, and loss operations are established; the methodological contribution lies in their grouped configuration, learnable low-/high-level fusion, and joint use of defective and defect-free samples for the present task.

Across the reported three-seed Campaign 5 evaluation, LPAU-Net achieved $89.84 \pm 0.08\%$ Precision, $89.24 \pm 0.08\%$ Recall, $89.54 \pm 0.08\%$ F1-score, and $81.07 \pm 0.13\%$ Defect IoU on defective images, with 4.34 M parameters, 14.8 G FLOPs, and 32.81 FPS under the desktop-GPU benchmark. Defect-free training reduced false-positive response and image-level FAR while slightly decreasing Recall, indicating a sensitivity–false-alarm trade-off rather than an across-the-board improvement.

Campaign 5 is a later time-separated acquisition batch from the same nine tower bodies and the same wind farm; the results therefore do not establish unseen-tower, cross-wind-farm, or external-site generalization. Strong light, shadows, and reflections were not evaluated as independent condition subsets, so no one-to-one robustness improvement can be reported for these factors. Because all visible anomalies were merged into one binary defect class, LPAU-Net localizes anomalous regions but cannot classify crack, pitting, honeycombing, or other defect types. Formal image-level significance testing, spatial attention visualization, geometric calibration, systematic TP/FP/FN failure analysis, and embedded-device testing were also not completed. Future work should address these limitations using larger independent datasets, category-specific labels, condition-stratified protocols, and true tower-level and cross-site validation.

Author Contributions: Conceptualization, R.W.; Methodology, R.W., Y.Z. and Q.G.; Validation, F.Z. and Q.G.; Formal analysis, W.P.; Investigation, F.Z.; Resources, F.Z., S.Y. and Q.G.; Data curation, Q.G.; Writing—original draft, R.W. and Y.M.; Writing—review and editing, R.W., Y.Z., W.P., S.Y. and Q.G.; Project administration, F.Z. and Q.G.; Funding acquisition, F.Z. and Q.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We thank the staff who offered assistance during the experimental tests. No generative artificial intelligence tools were used in the preparation of this manuscript.

Conflicts of Interest: Authors Fanqiang Zeng, Yinan Ma, Yu Zhang, Wanpeng Ping and Songbin Yang were employed by the company Heilongjiang Tietou Prefabricated Components Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Global Wind Energy Council. *Global Wind Report 2024*; Global Wind Energy Council: Brussels, Belgium, 2024.
2. IEC 61400-1:2019; Wind Energy Generation Systems-Part 1: Design Requirements; International Electrotechnical Commission: Geneva, Switzerland, 2019.
3. Tchakoua, P.; Wamkeue, R.; Ouhrouche, M.; Slaoui-Hasnaoui, F.; Tameghe, T.A.; Ekemb, G. Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges. *Energies* **2014**, *7*, 2595–2630.
4. Shihavuddin, A.; Chen, X.; Fedorov, V.; Nymark Christensen, A.; Andre Brogaard Riis, N.; Branner, K.; Bjorholm Dahl, A.; Reinhold Paulsen, R. Wind turbine surface damage detection by deep learning aided drone inspection analysis. *Energies* **2019**, *12*, 676.
5. Spencer, B.F., Jr.; Hoskere, V.; Narazaki, Y. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering* **2019**, *5*, 199–222.
6. Koch, C.; Georgieva, K.; Kasireddy, V.; Akinci, B.; Fieguth, P. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Adv. Eng. Inform.* **2015**, *29*, 196–210.
7. Cha, Y.J.; Choi, W.; Büyüköztürk, O. Deep learning-based crack damage detection using convolutional neural networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378.
8. Zhang, W.; Zhang, Z.; Qi, D.; Liu, Y. Automatic crack detection and classification method for subway tunnel safety monitoring. *Sensors* **2014**, *14*, 19307–19328.
9. Dung, C.V. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58.
10. Zou, Q.; Zhang, Z.; Li, Q.; Qi, X.; Wang, Q.; Wang, S. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Trans. Image Process.* **2018**, *28*, 1498–1512.
11. Ali, L.; Alnajjar, F.; Al Jassmi, H.; Gocho, M.; Khan, W.; Serhani, M.A. Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. *Sensors* **2021**, *21*, 1688.
12. Ali, L.; Alnajjar, F.; Khan, W.; Serhani, M.A.; Al Jassmi, H. Bibliometric analysis and review of deep learning-based crack detection literature published between 2010 and 2022. *Buildings* **2022**, *12*, 432.
13. Sohaib, M.; Arif, M.; Kim, J.M. Evaluating YOLO models for efficient crack detection in concrete structures using transfer learning. *Buildings* **2024**, *14*, 3928.
14. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
15. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
16. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder Architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495.
17. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
18. Zhou, Z.; Rahman Siddiquee, M.M.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *International Workshop on Deep Learning in Medical Image Analysis*; Springer International Publishing: Cham, Switzerland, 2018; pp. 3–11.
19. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B.; et al. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999.
20. Wang, J.; Chen, Z.; Dai, J.G. A framework for multi-class surface damage segmentation, quantification, and severity assessment of reinforced concrete structures with GAN data augmentation. *Comput.-Aided Civ. Infrastruct. Eng.* **2026**, *49*, 100102. <https://doi.org/10.1016/j.cacaie.2026.100102>.
21. Chen, Z.; Wang, J.; Dai, J.G. Attention-enhanced dual-stage deep learning for automated tunnel crack detection and quantification. *Comput.-Aided Civ. Infrastruct. Eng.* **2026**, *41*, 100020. <https://doi.org/10.1016/j.cacaie.2026.100020>.
22. Wang, J.; Yang, Y.; Wang, Z.; Lv, L.; Ueda, T. A scalable UAV-based structural health monitoring framework using augmented deep learning for multilevel damage classification. *J. Civ. Struct. Health Monit.* **2026**, *16*, 14. <https://doi.org/10.1007/s13349-025-01050-5>.
23. Wang, J.; Wang, Z.; Wang, Y.; Li, Z. Automated multi-type damage detection framework in reinforced concrete structures via data augmentation and deep segmentation networks. *J. Civ. Struct. Health Monit.* **2025**, *15*, 3861–3884. <https://doi.org/10.1007/s13349-025-01020-x>.
24. Chen, Y.; Zhu, Z.; Lin, Z.; Zhou, Y. Building surface crack detection using deep learning technology. *Buildings* **2023**, *13*, 1814.

25. Liu, Z.; Cao, Y.; Wang, Y.; Wang, W. Computer vision-based concrete crack detection using U-net fully convolutional networks. *Autom. Constr.* **2019**, *104*, 129–139.
26. Zhang, C.; Peng, N.; Yan, J.; Wang, L.; Chen, Y.; Zhou, Z.; Zhu, Y. A novel YOLOv10-DECA model for real-time detection of concrete cracks. *Buildings* **2024**, *14*, 3230.
27. Pu, R.; Ren, G.; Li, H.; Jiang, W.; Zhang, J.; Qin, H. Autonomous concrete crack semantic segmentation using deep fully convolutional encoder–decoder network in concrete structures inspection. *Buildings* **2022**, *12*, 2019.
28. Bergmann, P.; Batzner, K.; Fauser, M.; Sattlegger, D.; Steger, C. The MVTec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. *Int. J. Comput. Vis.* **2021**, *129*, 1038–1059.
29. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
30. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
31. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
32. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
33. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
34. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
36. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 23 July 2017; pp. 2980–2988.
39. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Jorge Cardoso, M. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *International Workshop on Deep Learning in Medical Image Analysis*; Springer International Publishing: Cham, Switzerland, 2017; pp. 240–248.
40. Jadon, S. A survey of loss functions for semantic segmentation. In *Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Via del Mar, Chile, 27–29 October 2020*; IEEE: New York, NY, USA; pp. 1–7.
41. Abraham, N.; Khan, N.M. A novel focal tversky loss function with improved attention u-net for lesion segmentation. In *Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019*; IEEE: New York, NY, USA; pp. 683–687.
42. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60.
43. Choi, S.M.; Cha, H.S.; Jiang, S. Hybrid data augmentation for enhanced crack detection in building construction. *Buildings* **2024**, *14*, 1929.
44. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and flexible image augmentations. *Information* **2020**, *11*, 125.
45. Perez, L.; Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv* **2017**, arXiv:1712.04621.
46. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
47. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
48. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.