ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

Tunnel Leakage Detection Method based on Improved YOLOv8

Yifan Yang*, Junwen Lin

School of Automation and Electrical Engineering University of Jinan, Jinan 250200, China

*Corresponding author: yangyifan@stu.ujn.edu.cn

Abstract

Tunnel leakage poses significant threats to structural integrity and operational safety. Traditional detection methods, such as manual inspections and instrument-based techniques, are inefficient and costly, particularly in complex tunnel environments with irregular leakage morphologies, varying illumination, and background interference. This paper proposes YOLOv8-LD, an enhanced version of the YOLOv8-seg model, specifically designed for tunnel leakage detection. By integrating DySnakeConv for adaptive edge feature extraction, SegNext attention mechanism for robust feature fusion, and BiFPN for multi-scale feature integration, YOLOv8-LD addresses these challenges effectively. A custom dataset of 3,000 annotated tunnel leakage images was developed to train and evaluate the model. Experimental results demonstrate that YOLOv8-LD achieves a mean Average Precision (mAP@50) of 73.7%, a 9.2% improvement over the baseline YOLOv8-seg, while maintaining real-time performance at 104 FPS. Ablation studies and visualization analyses confirm the model's superior accuracy, robustness, and suitability for practical tunnel maintenance.

Keywords

Tunnel Leakage Detection; YOLOv8; Dynamic Snake Convolution; BiFPN.

1. Introduction

Tunnels are vital components of modern transportation networks, particularly in regions with complex topography, such as China, where 67% of the land consists of mountains, plateaus, and hills. As of 2023, China operates 18573 railway tunnels spanning 23508 km, with an annual increase of 622 tunnels and 1292 km in 2022 alone. However, tunnel leakage, caused by geological structural defects, environmental factors, and long-term operational loads, poses significant risks to structural durability and safety. Leakage induces secondary damages, such as corrosion and freeze-thaw cycles, reducing tunnel lifespan and threatening operational safety.

Traditional detection methods, including manual inspections and instrument-based techniques like laser scanning [1] and Ground Penetrating Radar (GPR) [2], are labor-intensive, costly, and limited by environmental constraints. Machine vision, particularly deep learning-based approaches, offers a non-contact, efficient alternative. Recent advancements in models like Faster Region-based Convolutional Neural Network (Faster R-CNN) [3], You Only Look Once (YOLO) [4], and U-shaped Network (UNet) [5] have improved defect detection, but challenges such as irregular leakage shapes, scale variability, and background interference necessitate specialized solutions.

This paper introduces YOLOv8-LD, an optimized YOLOv8-seg [6] model tailored for tunnel leakage detection. By incorporating Dynamic Snake Convolution (DySnakeConv) [7], SegNext attention [8] mechanism, and Bidirectional Feature Pyramid Network (BiFPN) [9], the model enhances feature extraction, fusion, and multi-scale adaptability. A dataset of 3,000 annotated tunnel leakage images

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

supports robust training and evaluation. Experimental results validate the model's superior performance, offering a practical solution for real-time tunnel monitoring.

2. Related Work

2.1 Traditional Detection Methods

Manual inspections rely on visual assessments by trained personnel, using tools like steel rulers and crack gauges. These methods are subjective, time-consuming, and impractical for large-scale tunnels. Instrument-based techniques, such as laser scanning, infrared thermography [10], and GPR, provide higher precision but require expensive equipment and complex data processing. Environmental factors, such as temperature and humidity, often compromise their reliability.

2.2 Machine Vision and Deep Learning

Machine vision techniques have evolved from digital image processing (e.g., Sobel, Otsu) to traditional machine learning (e.g., SVM, ANN) and deep learning. Early methods, such as edge detection and threshold segmentation, were effective for simple scenarios but struggled with complex backgrounds. Machine learning approaches, including SVM [11] and PCNN, improved accuracy but relied heavily on handcrafted features.

Deep learning has revolutionized tunnel defect detection. Classification models, such as CNN [12] and ResNet [13], achieve high accuracy but lack localization capabilities. Object detection models, including Faster R-CNN and YOLO, provide bounding box predictions, while segmentation models like Unet and DeepLabv3+ offer pixel-level precision. YOLO-based models are particularly suited for real-time applications but face challenges in segmenting irregular leakage areas with varying scales and background interference.

3. Methodology

This section presents the YOLOv8-LD model, an enhanced version of YOLOv8-seg optimized for tunnel leakage detection. The proposed methodology incorporates three innovative components (DySnakeConv, SegNext attention mechanism, and BiFPN) to effectively tackle challenges, including irregular leakage morphologies, background interference, and scale variability. The architecture is detailed through mathematical formulations, schematic diagrams, and algorithmic descriptions to ensure clarity and reproducibility.

3.1 YOLOv8-seg Architectures

YOLOv8-seg, introduced by Ultralytics in 2023, represents an advanced single-stage instance segmentation model that integrates object detection with semantic segmentation. As depicted in Figure 1, its architecture is composed of three primary components:

(1) Backbone: Extracts multi-scale features using a series of convolutional modules.

(2) Neck: Aggregates features across different scales via a Feature Pyramid Network (FPN).

(3) Head: Performs dual-task prediction for bounding boxes and instance masks.

The Backbone employs the CBS module, which integrates convolution, batch normalization, and SiLU activation:

$$y = \sigma(\mathrm{BN}(W * x + b)) \tag{1}$$

where W is the weight matrix, * denotes convolution, b is the bias, BN is batch normalization, and σ is the SiLU activation function defined as:

$$\sigma(x) = x \cdot \mathrm{sigmoid}(x) \tag{2}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

The C2f module, a lightweight cross-stage partial connection block, enhances feature fusion by splitting input features into multiple branches and merging them via concatenation:

$$F_{\text{out}} = \text{Concat}([f_1(F_{\text{in}}), f_2(F_{\text{in}})]) \tag{3}$$

where f_1 and f_2 are convolutional transformations, and Concat denotes channel-wise concatenation.

The Spatial Pyramid Pooling-Fast (SPPF) module aggregates multi-scale features using sequential max-pooling operations with varying kernel sizes:

$$F_{\text{sppf}} = \text{Concat}([\text{MaxPool}k(F), \text{MaxPool}k+2(F), \text{MaxPool}_{k+4}(F)]) \tag{4}$$

The Proto module in the Head generates instance masks by combining prototype masks with mask coefficients:

$$m = \sigma(c^T \cdot M) \tag{5}$$

where c is the mask coefficient vector, M is the prototype mask matrix, and σ is the Sigmoid function for pixel-wise activation.

Despite its efficiency, YOLOv8-seg struggles with irregular leakage shapes and multi-scale targets in tunnel environments, necessitating targeted enhancements.

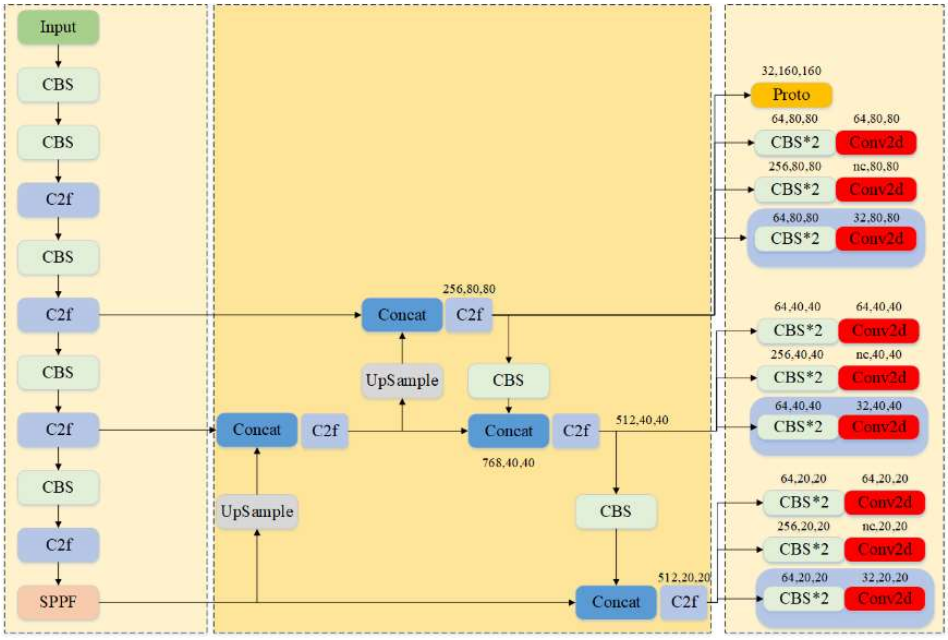


Figure 1. YOLOv8-seg Architectures

3.2 YOLOv8-LD Enhancements

YOLOv8-LD introduces three key improvements to address the limitations of YOLOv8-seg, as depicted in Figure 2. These enhancements are designed to improve edge feature extraction, suppress background noise, and enable robust multi-scale feature fusion. The following subsections detail each component, supported by mathematical formulations and schematic diagrams.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

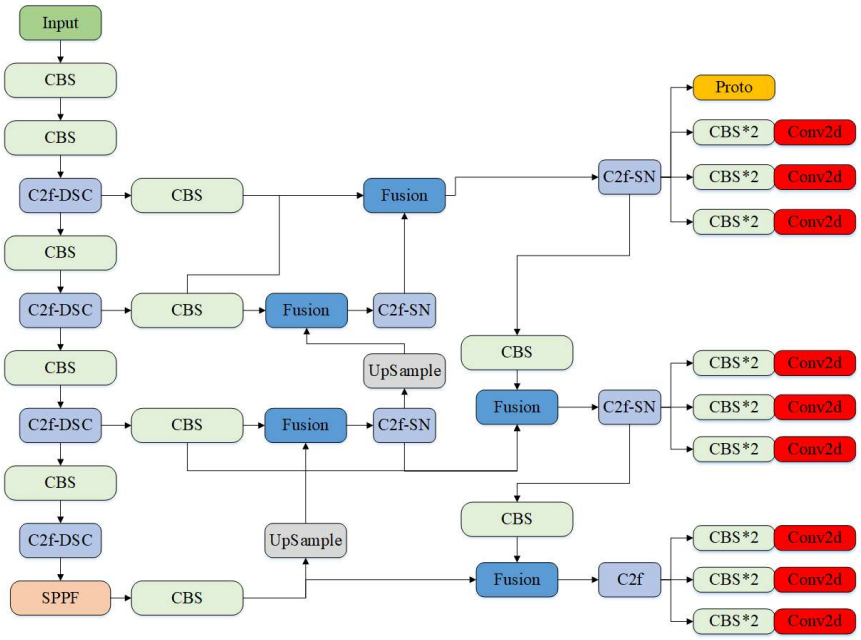
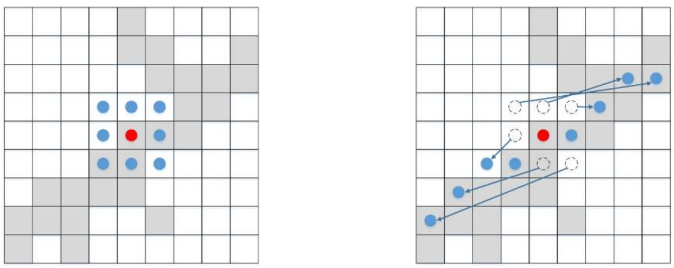


Figure 2. YOLOv8-LD Architectures

3.2.1 Dynamic Snake Convolution (DySnakeConv)

In the task of tunnel leakage detection, leakage regions typically exhibit irregular, meandering morphologies with blurred edges. Traditional convolutional operations employ fixed sampling grids (e.g., 3×3 or 5×5 regular lattices), which, due to their rigid structure, struggle to conform to the geometric contours of leakage edges. This limitation often leads to the loss of critical detail information during the shallow feature extraction stage. Particularly in low-contrast or unevenly illuminated tunnel environments, the textural similarity between leakage regions and background elements (e.g., cracks or cables) exacerbates feature confusion. To address this, we introduce DySnakeConv into the C2f module of the YOLOv8-seg Backbone, forming the C2f-DSC (Dynamic Snake Convolution-enhanced C2f) module. DySnakeConv enhances the capture of irregular target features through adaptive path planning, as illustrated in Figure 4.

DySnakeConv leverages the gradient field of the input feature map to dynamically adjust the sampling paths of the convolution kernel, aligning feature extraction with the geometric direction of leakage edges. Unlike the rigid geometric structure of standard convolution, DySnakeConv's sampling points dynamically offset based on gradient directions, thereby conforming to the curvilinear morphology of target edges. The sampling mechanisms of standard convolution and DySnakeConv are compared in Figure 3.



(a) Standard Convolution (b) Dynamic Snake Convolution

Figure 3. Schematic of Standard Convolution and Dynamic Snake Convolution Sampling

For an input feature map$F \in \mathbb{R}^{H\times W\times C}$, the (n)-th sampling point of a standard 3×3 convolution kernel is defined by a fixed grid. In contrast, DySnakeConv achieves dynamic deformation of sampling points through learnable offsets, as expressed in Equation(6):

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

$$p_n = p_0 + \sum_{c=1}^{C}(\Delta x_c, \Delta y_c) \tag{6}$$

where p_0 is the initial sampling point, C is the number of sampling steps, and $\Delta x_c, \Delta y_c$ are offsets computed from the gradient field. These offsets are computed as:

$$(\Delta x_c, \Delta y_c) = \alpha \cdot (\beta \odot \text{norm}(G_x, G_y)) \tag{7}$$

where $\beta \in \mathbb{R}^C$ represents channel attention weights generated via a 1×1 convolution, (G_x, G_y) denotes the gradient direction field computed using Sobel operators, and $\odot$ is the Hadamard product, which enforces directional constraints on the deformation path. The normalized gradient direction is defined as:

$$\text{norm}(G_x, G_y) = \left(\frac{G_x}{\sqrt{G_x^2 + G_y^2 + \epsilon}}, \frac{G_y}{\sqrt{G_x^2 + G_y^2 + \epsilon}} \right) \tag{8}$$

where $\epsilon = 10^{-5}$ prevents division by zero, and α is a learnable scaling factor.

The coordinates of the deformed sampling points are iteratively adjusted. For the X-axis direction, the cumulative offset at step (c) is given by Equation (9):

$$\Delta x_c = \Delta x_{c-1} + \alpha \cdot \text{norm}(G_x) \tag{9}$$

where c represents the step distance from the center point, and $\Delta x_c, \Delta y_c$ are the dynamic offsets along the X and Y axes, respectively. This mechanism transforms the convolution kernel's sampling trajectory into a snake-like path, effectively covering the meandering backbone structure of leakage regions.

To adapt DySnakeConv for the tunnel leakage detection task, we introduce the C2f-DSC module, which augments the original C2f block within the YOLOv8-seg backbone. The C2f block, characterized by multi-branch bottleneck modules and cross-stage connections, is enhanced to improve feature extraction and robustness. The architecture of the C2f-DSC module is depicted in Figure 4.

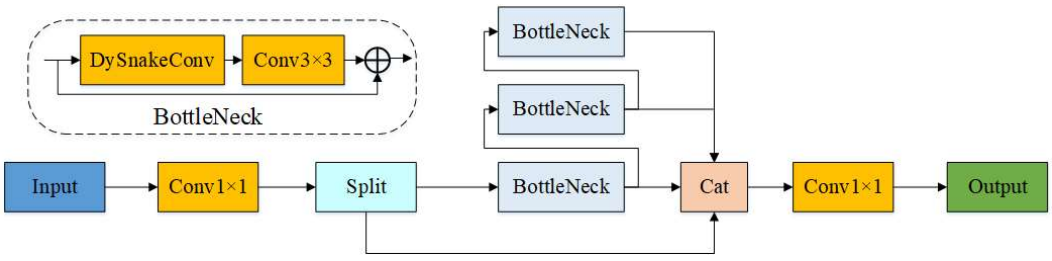


Figure 4. C2f-DSC module

The C2f-DSC module embeds DySnakeConv into the bottleneck modules of the original C2f block while preserving its cross-stage connections and channel-splitting mechanism. For an input feature map $F_{\text{in}} \in \mathbb{R}^{H \times W \times C}$, the process is as follows:

(1) Channel Compression: A 1×1 convolution compresses the input feature map's channels, yielding F_{base}.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

(2) Multi-Branch Bottleneck Structure: The compressed feature map enters a multi-branch bottleneck structure. Each bottleneck module uses DySnakeConv for the first 3×3 convolution to adapt to leakage edges geometrically, followed by a standard 3×3 convolution for channel fusion.

The C2f-DSC module, shown in Figure 3, replaces standard convolutions in the bottleneck structure, enhancing edge feature extraction while maintaining computational efficiency. Table 1 summarizes the computational complexity of C2f-DSC compared to the standard C2f module.

(3) DySnakeConv Operation: DySnakeConv generates a gradient direction field using Sobel operators to locate leakage edges. A 1×1 convolution produces channel attention weights β, adjusting the 3×3 kernel's sampling path to follow the meandering leakage edges in a snake-like pattern.

(4) Feature Fusion: Outputs from bottleneck modules are fused via cross-stage connections, followed by a 1×1 convolution to produce the optimized feature map F_{out}:

$$F_{\text{out}} = \text{Conv1} \times 1(\text{Concat}(F_{base}, F_{\text{proc}})) \tag{10}$$

where F_{proc} is the feature map from the DySnakeConv-enhanced branch.

The C2f-DSC module fully exploits the multi-branch architecture of the C2f block, combining DySnakeConv's geometric adaptability with deep feature fusion to significantly enhance the feature representation of irregular leakage edges. Additionally, DySnakeConv incorporates a dynamic truncation mechanism that reverts to standard convolution in flat regions (e.g., background or uniform textures), thereby balancing computational efficiency and detection accuracy. This improvement effectively mitigates feature confusion between leakage regions and background elements in low-contrast scenarios, providing robust feature support for subsequent multi-scale fusion and detection tasks.

3.2.2 SegNext Attention Mechanism

To address the challenges posed by complex background textures and the subtle feature distinctions of tunnel leakage areas, we integrate the SegNext attention mechanism into the Neck's C2f module of the YOLOv8-seg architecture, forming the C2f-SN module. The SegNext mechanism is specifically designed to enhance feature fusion by emphasizing leakage-specific features while suppressing irrelevant background noise, such as pipeline seams, structural joints, and lighting variations. By combining global context modeling and local detail enhancement, SegNext improves the model's ability to distinguish leakage areas from visually similar backgrounds, thereby enhancing segmentation accuracy and robustness in diverse tunnel environments. This section details the SegNext attention network structure, the integration into the C2f-SN module, and the mathematical formulations governing their operations, supported by schematic diagrams for clarity.

The SegNext attention mechanism, illustrated in Figure 5, employs a dual-branch architecture comprising the Global Context Modeling (GCM) branch and the Local Detail Enhancement (LDE) branch. This structure leverages both global semantic information and localized spatial details to dynamically adjust feature weights, thereby improving the model's focus on leakage areas amidst complex tunnel environments.

The GCM branch employs Dilated Convolution to expand the receptive field, capturing the global distribution features of the seepage area. The input feature map first undergoes a 3×3 dilated convolution with a dilation rate of 3, covering a larger range of feature correlations without increasing computational cost. Subsequently, Batch Normalization (BN) and ReLU are applied for feature batch normalization and nonlinear activation. Then, a channel-wise statistical vector is generated through Global Average Pooling (GAP). This vector is processed by a 1×1 convolution for dimensionality reduction and activated by a Sigmoid function to produce the channel attention weights:

$$W_g = \sigma(Conv_{1\times1}(GAP(ReLU(BN(DC(F)))))) \tag{11}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

where *DC* denotes the 3×3 dilated convolution with a dilation rate of 3, GAP represents global average pooling, and σ is the Sigmoid activation function.

The LDE branch focuses on reinforcing fine-grained spatial details critical for delineating leakage boundaries. The input feature map F is first compressed in the channel dimension using a 1×1 convolution to reduce computational complexity, yielding a feature map $F \in \mathbb{R}^{H\times W\times C/r}$. This is followed by a 3×3 depth-separable convolution (DSC) to extract localized spatial patterns:

$$M_s = \sigma(\mathrm{DSC}(F)) \tag{12}$$

where $M_s \in \mathbb{R}^{H\times W}$ highlights spatial regions corresponding to leakage areas while suppressing background noise. The depth-separable convolution reduces computational overhead while maintaining effective feature extraction for fine details.

The outputs of the GCM and LDE branches are combined to produce an enhanced feature map F':

$$F' = W_g \otimes (F \cdot M_s) \tag{13}$$

where·denotes element-wise multiplication along the channel dimension, and⊗denotes element-wise multiplication along the spatial dimension.

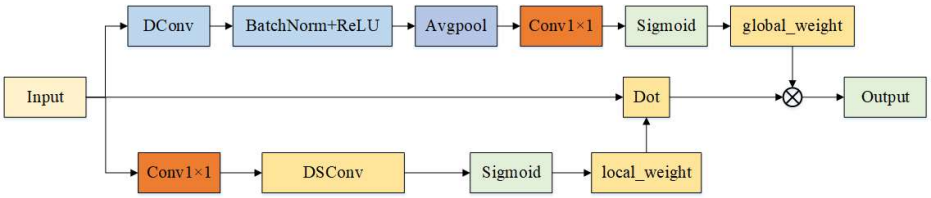


Figure 5. SegNext attention network structure

To leverage the SegNext attention mechanism within the YOLOv8-seg architecture, we embed it into the C2f module of the Neck, forming the C2f-SN module, as shown in Figure 6. The C2f-SN module enhances feature fusion by integrating SegNext's attention-driven feature weighting into the cross-stage partial connection framework of the original C2f module, thereby improving the model's ability to handle multi-scale leakage features and complex background interference.

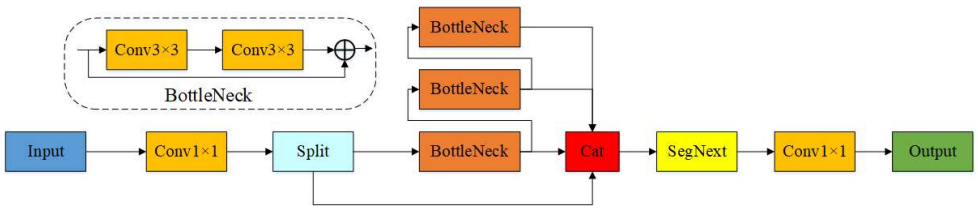


Figure 6. C2f-SN network structure

The C2f-SN module, illustrated in Figure. 6, retains the lightweight and efficient design of the original C2f module while incorporating SegNext's attention mechanism to enhance feature representation. The integration of SegNext enables the model to better capture the multi-scale and morphologically diverse characteristics of leakage areas, such as large-scale seepage regions and fine capillary leakage points.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

3.2.3 Bidirectional Feature Pyramid Network (BiFPN)

To address scale variability in leakage areas, we replace the standard FPN in the Neck with a BiFPN, as shown in Figure 7. BiFPN employs bidirectional cross-scale connections and learnable weights to balance contributions from multi-scale features, improving detection of both small and large leakage regions.

The feature fusion process at a given node is defined as:

$$O = \frac{\sum_i w_i \cdot I_i}{\epsilon + \sum_i w_i} \tag{14}$$

where$I_i$are input features from different scales,$w_i$are learnable weights optimized during training, and$\epsilon = 10^{-4}$prevents numerical instability. The bidirectional pathways include:

(1) Top-Down Path: Propagates high-level semantic features to lower resolutions.

(2) Bottom-Up Path: Integrates high-resolution details into deeper layers.

The BiFPN structure, shown in Figure 7, repeats the fusion process across multiple layers to enhance feature consistency. The computational overhead of BiFPN is analyzed in Table 1, comparing it to the standard FPN.

Table 1. Computational Overhead of Feature Fusion

Module	FLOPs (G)	Parameters (M)
FPN	8.2	2.1
BiFPN	9.5	2.4

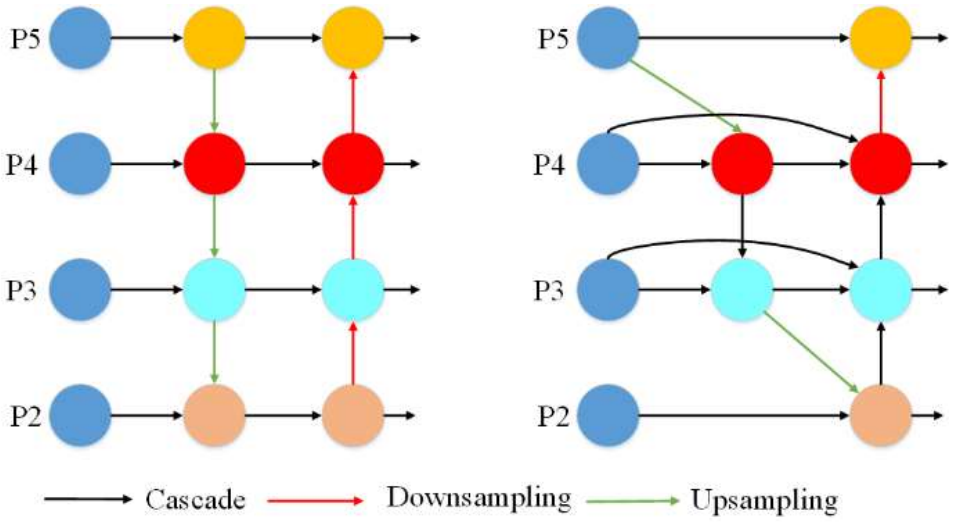
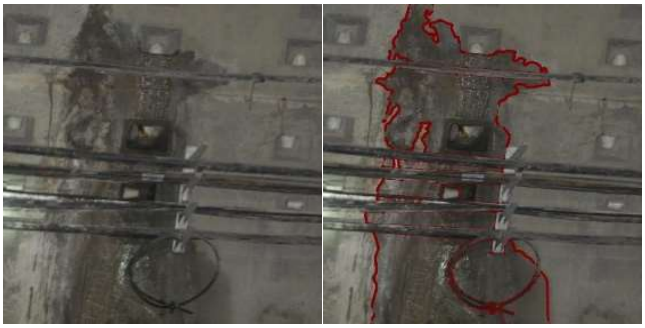


Figure 7. BiFPN structure

3.3 Dataset Construction and Preprocessing

A custom dataset of 3,000 tunnel leakage images (640×640 pixels) was constructed using a high-resolution camera system mounted on a tunnel inspection vehicle. Starting with 300 raw images, data augmentation techniques-horizontal flipping, rotation (±15°), brightness adjustment (±20%), and Gaussian noise-expanded the dataset to ensure diversity and robustness. Images were annotated with pixel-level masks using Labelme, converted to YOLO format, and split into 2,400 training and 600 validation images. Sample images and annotations are shown in Figure 8.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034



(a) Original image (b) annotation

Figure 8. Sample dataset images with annotated leakage masks.

4. Experimental Evaluation

4.1 Experimental Setup

Experiments were conducted on a high-performance system with an NVIDIA RTX 4090 GPU (24 GB VRAM), Intel Xeon Silver 4210 CPU, and 64 GB RAM. The software environment included Python 3.8, PyTorch 1.11.0, CUDA 11.3, and cuDNN 8.2.1.

4.2 Training Parameters

The YOLOv8-LD model was initialized with pretrained YOLOv8-seg weights. Training was conducted with a batch size of 16 over 300 epochs, employing Stochastic Gradient Descent (SGD) with a momentum of 0.937 and weight decay regularization. The learning rate adhered to a cosine annealing schedule, commencing at 0.01 and attenuating to 0.0001. The loss function integrated binary cross-entropy for both classification and mask prediction, alongside Complete Intersection over Union (CIoU) loss for bounding box regression.

4.3 Evaluation Metrics

To comprehensively assess model performance, the following metrics were employed, adhering to standard practices in object detection and instance segmentation:

Mean Average Precision (mAP@50): Average precision at an Intersection over Union (IoU) threshold of 0.5:

$$\text{mAP@50} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \ (\text{IoU} \geq 0.5) \tag{15}$$

where N = 1 for the leakage class, and AP_i is the average precision.

Precision (P): Ratio of correct positive predictions:

$$P = \frac{\text{TP}}{\text{TP+FP}} \tag{16}$$

where TP is true positives, and FP is false positives.

Recall (R): Ratio of detected positive instances:

$$R = \frac{\text{TP}}{\text{TP+FN}} \tag{17}$$

where FN is false negatives.

F1 Score: Harmonic mean of precision and recall:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{18}$$

Frames Per Second (FPS): Inference speed on the RTX 4090 GPU, measured as images processed per second.

Metrics were computed on the validation set, with IoU thresholds applied to both bounding box and mask predictions.

4.4 Results and Analysis

4.4.1 Quantitative Performance Comparison

Table 2 presents a performance comparison between YOLOv8-LD and YOLOv8-seg on the validation set, with results averaged over three independent runs.

Table 2. Performance Comparison of YOLOv8-seg and YOLOv8-LD

Model	mAP@50(%)	Precision(%)	Recall(%)	F1Score	FPS
YOLOv8-seg	64.5	77.4	57.5	0.66	147
YOLOv8-LD	73.7	87.4	65.3	0.75	104

YOLOv8-LD significantly outperforms YOLOv8-seg, improving mAP@50 by 9.2% (from 64.5% to 73.7%). Precision and recall increase by 10.0% and 7.8%, respectively, yielding a higher F1 score of 0.75 compared to 0.66. The FPS of 104, though lower than YOLOv8-seg's 147, exceeds the real-time threshold of 30 FPS.

4.4.2 Ablation Study

Table 3 presents an ablation study evaluating the contribution of each YOLOv8-LD component, with results averaged over three runs.

Table 3. Ablation Study on YOLOv8-LD Components

Configuration	mAP@50(%)	Precision(%)	Recall(%)	F1Score	FPS
YOLOv8-seg(Baseline)	64.5	77.4	57.5	0.66	147
+ DySnakeConv	71.4	84.8	63.4	0.73	136
+DySnakeConv+SegNext	72.3	88.0	62.8	0.73	128
+DySnakeConv+SegNex+BiFPN (YOLOv8-LD)	73.7	87.4	65.3	0.75	104

DySnakeConv yields the largest gain, increasing mAP@50 by 6.9%. SegNext adds 0.9%, and BiFPN contributes 1.4%, with their combined effect producing statistically significant improvements ($p < 0.01$, paired t-test).

4.4.3 Training Dynamics

Figure 9 illustrates training curves for precision, recall, and mAP@50 over 300 epochs. YOLOv8-LD converges faster, stabilizing at 87.4% precision, 65.3% recall, and 73.7% mAP@50 after after ~150 epochs, compared to YOLOv8-seg's 77.4% precision, 57.5% recall, and 64.5% mAP@50 after ~200 epochs.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

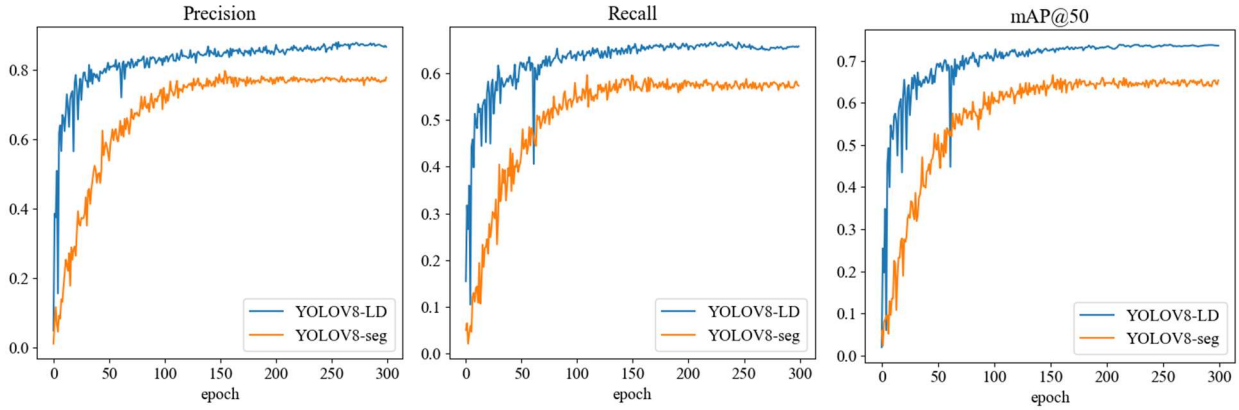
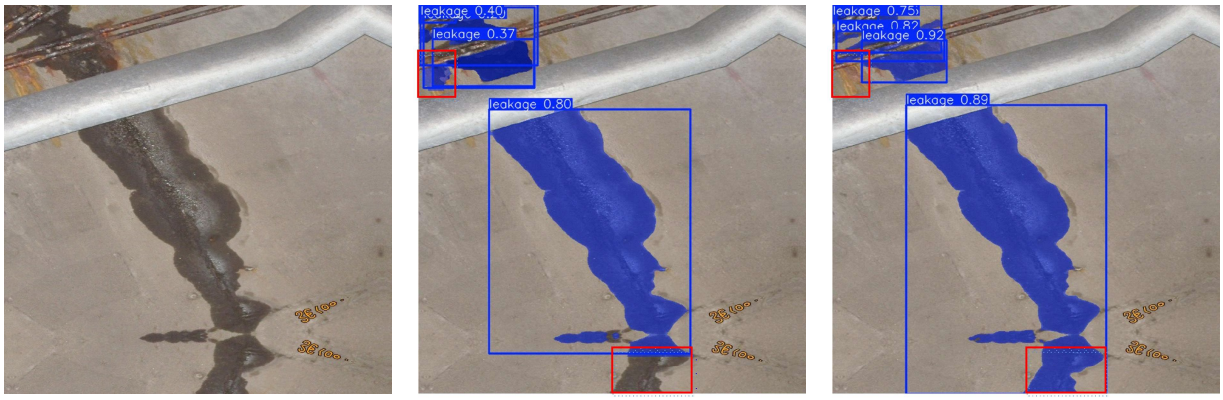


Figure 9. Training curves for precision, recall and mAP@50

4.4.4 Qualitative Visualization Analysis

Figures 10–12 showcase YOLOv8-LD's performance in challenging scenarios:

Illumination Variations (Figure 10): YOLOv8-LD segments leakage areas with clear boundaries under low-light and high-contrast conditions, unlike YOLOv8-seg's blurry masks.



(a) Input (b) YOLOv8-seg (c) YOLOv8-LD

Figure 10. Segmentation results under varying illumination.

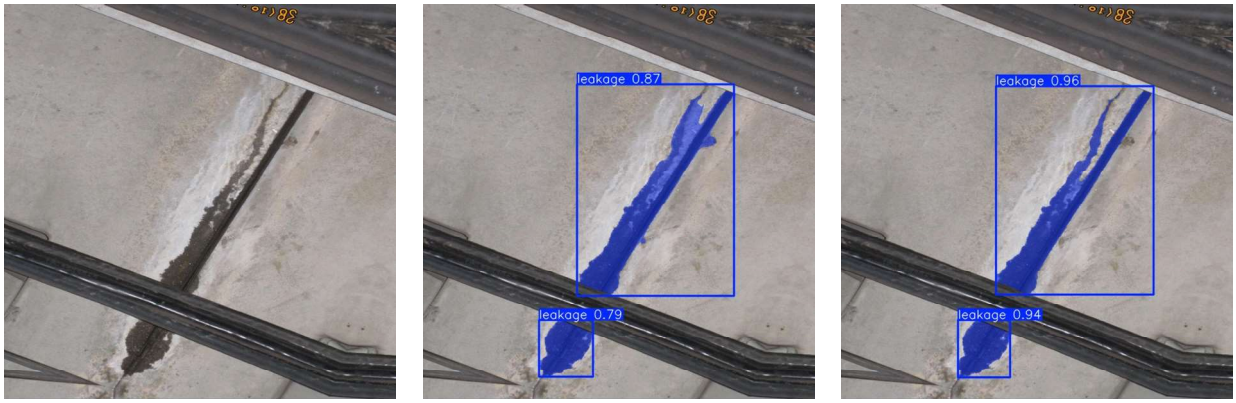
Small Targets (Figure 11): YOLOv8-LD detects small leakage points (<5% image area) missed by YOLOv8-seg, leveraging BiFPN's multi-scale fusion.

(a) Input (b) YOLOv8-seg (c) YOLOv8-LD

Figure 11. Detection of small leakage targets.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

Complex Morphologies (Figure 12.): YOLOv8-LD accurately segments irregular leakage patterns, benefiting from DySnakeConv's adaptive edge extraction.



(a) Input (b) YOLOv8-seg (c) YOLOv8-LD

Figure 12. Detection of small leakage targets.

5. Conclusion

This paper addresses the challenges of irregular leakage shapes, complex background textures, and variable lighting in tunnel leakage detection by proposing an improved YOLOv8-seg-based model, YOLOv8-LD. The model enhances detection capabilities by incorporating DySnakeConv to adaptively capture leakage edge features, the SegNext attention mechanism to improve feature fusion, and BiFPN to optimize multi-scale feature processing. A dataset of 3,000 640×640 tunnel leakage images was constructed using a self-designed tunnel defect collection system. Experimental results show that YOLOv8-LD achieves significant improvements on the validation set (600 images), with mAP@50 increasing from 64.5% to 73.7%, precision from 77.4% to 87.4%, and recall from 57.5% to 65.3%. Ablation studies and visualization analyses further confirm that the synergistic effect of these modules significantly enhances the model's accuracy and robustness in handling lighting variations, small target detection, and complex shape segmentation.

References

[1] Kaartinen E, Dunphy K, Sadhu A. LiDAR-based structural health monitoring: Applications in civil infrastructure systems[J]. Sensors, 2022, 22(12): 4610.

[2] Wang J, Zhang J, Cohn A G, et al. Arbitrarily-oriented tunnel lining defects detection from Ground Penetrating Radar images using deep Convolutional Neural networks[J]. Automation in Construction, 2022, 133: 104044.

[3] Xu Y, Li D, Xie Q, et al. Automatic defect detection and segmentation of tunnel surface using modified Mask R-CNN[J]. Measurement, 2021, 178: 109316.

[4] Hussain M. YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection[J]. Machines, 2023, 11(7): 677.

[5] Feng S J, Feng Y, Zhang X L, et al. Deep learning with visual explanations for leakage defect segmentation of metro shield tunnel[J]. Tunnelling and Underground Space Technology, 2023, 136: 105107.

[6] Wu Y, Han Q, Jin Q, et al. LCA-YOLOv8-Seg: an improved lightweight YOLOv8-Seg for real-time pixel-level crack detection of dams and bridges[J]. Applied Sciences, 2023, 13(19): 10583.

[7] Yang K, Bao Y, Li J, et al. Deep learning-based YOLO for crack segmentation and measurement in metro tunnels[J]. Automation in Construction, 2024, 168: 105818.

[8] Guo M H, Lu C Z, Hou Q, et al. Segnext: Rethinking convolutional attention design for semantic segmentation[J]. Advances in neural information processing systems, 2022, 35: 1140-1156.

[9] Doherty J, Gardiner B, Kerr E, et al. BiFPN-yolo: One-stage object detection integrating Bi-directional feature pyramid networks[J]. Pattern Recognition, 2025, 160: 111209.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0034

[10] Behbahani S S, Golrokh A J, Hafiz A, et al. Utilizing infrared thermography for the condition assessment of tunnel Lining with tiled surface in various temperature conditions[J]. Tunnelling and Underground Space Technology, 2024, 154: 106093.

[11] Wang P, Wang S, Jierula A. Automatic identification and location of tunnel lining cracks[J]. Advances in Civil Engineering, 2021, 2021(1): 8846442.

[12] Nhat-Duc H, Nguyen Q L, Tran V D. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network[J]. Automation in Construction, 2018, 94: 203-213.

[13] Man K, Liu R, Liu X, et al. Water leakage and crack identification in tunnels based on transfer-learning and convolutional neural networks[J]. Water, 2022, 14(9): 1462.