

Article

A Lightweight Model-Based Intelligent Recognition Approach for Multi-Category Tunnel Lining Defects Using GPR Data

Yuhao Liu , Hang Zhang *  and Yijun Wang

School of Automation, Central South University, Changsha 410083, China; 8207241709@csu.edu.cn (Y.L.); 703066@csu.edu.cn (Y.W.)

* Correspondence: zhang22@csu.edu.cn

Abstract

Tunnel lining defects pose significant threats to structural integrity and operational safety. Traditional image processing and machine learning methods often suffer from limited accuracy and poor generalization under complex backgrounds. To address these limitations, this study proposes a lightweight intelligent recognition method based on You Only Look Once version 11 nano (YOLOv11n) for Ground Penetrating Radar (GPR) images of tunnel linings. The backbone is replaced with Mobile Network Version 3 (MobileNetV3) to reduce parameters and Floating Point Operations (FLOPs), while depthwise separable convolution and a streamlined Compressed 2-Stage Fused-Lite (C2f-Lite) structure are integrated into the Neck to further decrease computational overhead. Channel mapping layers are employed to ensure smooth feature transfer, and selective use of Squeeze-and-Excitation (SE) attention and Hard-Swish (H-swish) activation balances detection accuracy with efficiency. Evaluated on a low-power mobile workstation acting as an edge-precursor proxy platform, experimental results demonstrate that the improved YOLOv11n_MobileNetV3 model achieves high accuracy with a mean Average Precision (mAP) at 0.5 of 94.4% and mAP@0.5:0.95 of 62.4%, low computational cost of 4.7 Giga Floating Point Operations (GFLOPs), and fast inference speed of 45 Frames Per Second (FPS). Comparative analysis further confirms its superior balance of detection performance and efficiency over YOLO version 5 (YOLOv5) and YOLO version 8 (YOLOv8) baselines. The proposed approach provides a highly optimized, edge-oriented engineering solution for real-time tunnel lining defect inspection, establishing strong structural and theoretical feasibility for future deployment in embedded systems.

Keywords: tunnel lining defects; GPR; lightweight model; object detection; deep learning



Academic Editor: Bingxiang Yuan

Received: 24 June 2026

Revised: 20 July 2026

Accepted: 23 July 2026

Published: 25 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

Tunnel linings, as one of the primary structural components of tunnels, serve to withstand surrounding rock pressure, mitigate geotechnical weathering, prevent structural collapse, and facilitate tunnel waterproofing and drainage. Nevertheless, driven by compounding factors including design discrepancies, construction variability, deferred maintenance, and prolonged service life, tunnel linings are afflicted with multi-scale defects of varying degrees [1], which manifest as structural pathologies during service. These structural anomalies not only present significant operational disruptions to normal tunnel serviceability but also compromise the safety of transport corridors, threatening human lives and properties [2].

GPR serves as the primary data acquisition methodology and foundational framework for subsurface object detection in modern structural health monitoring. As a non-destructive geophysical technology, GPR utilizes high-frequency electromagnetic radar pulses to image subsurface profiles and interfaces [3]. By extracting subsurface structural geometries and dielectric material discontinuities from the propagation, attenuation, and reflection characteristics of electromagnetic waves, GPR offers an array of engineering advantages characterized by rapid scanning speeds, high resolution, and flexible field deployment [4]. Consequently, it has been widely accepted as the benchmark technology for the non-destructive inspection of tunnel linings.

In the nascent stages of automated tunnel lining defect identification, traditional digital image processing pipelines predominated. These techniques heavily relied on deterministic mathematical operators to isolate and parse lining anomalies from empirical radargrams. For example, edge detection algorithms [5,6], morphological filters, and localized feature extraction were extensively deployed to identify surface-level fractures and subsurface fluid seepage within tunnel boundaries [7]. Although computationally straightforward and intuitive to implement, these heuristic approaches exhibit fragile recognition stability when handling low-contrast or low-signal-to-noise-ratio (SNR) radargrams, struggling to separate authentic defects from highly convoluted background clutter. Beyond raw image processing, standard machine learning frameworks were introduced to achieve multi-class classification. For instance: Liu et al. employed the Support Vector Machine (SVM) algorithm to classify cracks in tunnel lining images [8]; Liu et al. adopted the Random Forest (RF) algorithm for the evaluation and prediction of tunnel lining defects (including cracks, water seepage, and spalling) [9]; Yu et al. generated panoramic images of tunnel linings through image processing and model recognition, achieving crack detection via image enhancement, edge detection, interference point removal, crack connection, and feature statistics [10]. While these classic machine learning paradigms yielded improved recognition accuracy over baseline image processing, they fundamentally depend on heavily handcrafted feature engineering and possess restricted generalization boundaries when applied to diverse engineering sites.

In recent years, the paradigm shift toward deep learning has rapidly advanced the domain of automated tunnel lining defect identification, effectively circumventing the limitations of handcrafted feature engineering. Li et al. designed an original-dimension-preserving YOLO architecture integrated with depthwise separable network layers to minimize feature map information loss [11]; meanwhile, Li et al. optimized the Faster Region-based Convolutional Neural Network (R-CNN) feature extraction network based on transfer learning with VGG16, achieving high-precision automated bridge crack recognition [12]. For automated tunnel crack parsing, Zhou et al. introduced a YOLOv4-ED algorithm for automated tunnel crack detection, though its detection accuracy requires further improvement [13], and Chen et al. fused the CBAM attention module with a MobileNetV3-Large backbone network, significantly enhancing crack recognition accuracy [14]; Liu et al. proposed an automatic detection and localization method based on deep learning and transfer learning to determine the precise locations of rebars using object detection algorithms [15]; Kaur et al. developed an algorithm for automatic rebar detection that integrates image gradient features and hyperbolic characteristics of rebars, achieving detection through curve fitting [16]; Han et al. addressed the class imbalance problem caused by small-scale cracks in images by proposing a ternary classifier-based method for tunnel lining crack detection, which significantly reduced the false positive rate [17]; Yan et al. introduced a rapid detection and processing method for tunnel lining cracks based on deep learning, employing an AlexNet convolutional neural network within

the Caffe framework for crack recognition; the effectiveness of the proposed model was validated through experiments [18].

Although significant progress has been made in intelligent recognition technologies for tunnel lining defects, several limitations and shortcomings remain, primarily manifested in the following aspects: (1) Imbalance between Lightweight Design and Accuracy: Existing lightweight methods, such as pruning and quantization, can reduce the number of model parameters and computational load but often at the cost of precision, particularly leading to decreased accuracy in detecting small targets (e.g., fine cracks) [19]. Moreover, there is a lack of lightweight designs specifically tailored for tunnel environments, failing to fully account for the characteristics of tunnel lining defects and falling short of meeting real-time inspection requirements. (2) Multi-Defects Joint Detection Problem: Most existing models are designed for single-defect detection, lacking the capability for joint detection of multiple defects; multi-task learning is prone to task conflicts due to differences in defect characteristics. (3) Dataset Imbalance and Long-Tailed Physical Constraints: Due to field safety restrictions, empirical GPR datasets exhibit severely unbalanced, long-tailed distributions, where standard structural components (e.g., rebars) heavily outnumber rare, high-risk anomalies (e.g., cavities). This feature sparsity closely mirrors the heterogeneous damage evolution and micro-crack propagation modeled in advanced rock mechanics, such as the statistical damage mechanisms in gas-bearing coal [20] or thermo-mechanical degradation and coupled damage mechanics governing micro-crack network evolution in granite subjected to extreme thermal shocks [21,22]. Neglecting such highly skewed representation spaces inevitably induces severe optimization bias in deep models, leading to degraded average precision (AP) for few-shot categories and elevated false alarms amid background clutter.

To address the aforementioned challenges, this paper utilizes GPR images of tunnel linings as the dataset, conducts network training and testing based on the YOLOv11n model, and accomplishes the automatic detection and identification of tunnel lining defects. The specific research contents are as follows: (1) A labeled training dataset was constructed based on multi-source tunnel lining defect data, incorporating GPR-acquired images and data augmentation techniques. (2) For the purpose of enhancing defect detection accuracy, inference speed, and lightweight efficiency, the YOLOv11n model was improved by integrating MobileNetV3 and depthwise separable convolutions, thereby proposing an optimized lightweight model. (3) By conducting ablation and comparative experiments with augmented datasets and benchmark models, the effectiveness of the improved YOLOv11n model was validated in terms of accuracy, speed, and generalization. The flowchart of this paper is shown in Figure 1.

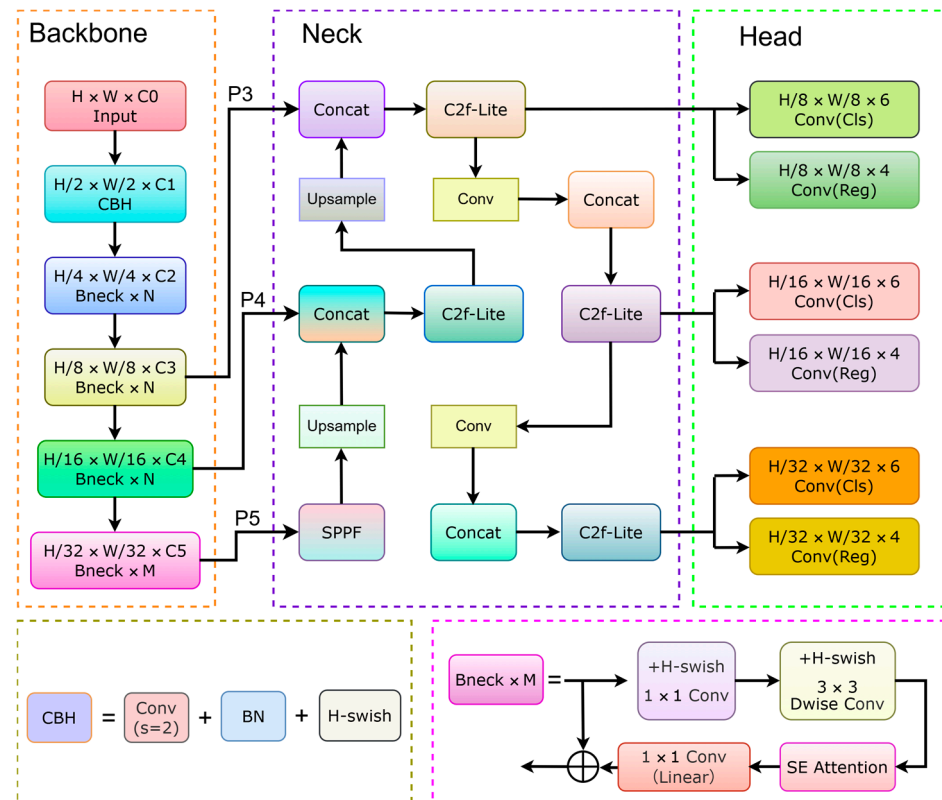


Figure 1. Flowchart of lightweight YOLOv11-based intelligent recognition of tunnel lining defects.

2. Intelligent Recognition of Tunnel Lining Defects Based on Lightweight YOLOv11

2.1. Design of Lightweight Scheme

Deploying deep learning models for tunnel lining defect detection on resource-constrained edge and embedded devices faces significant challenges due to stringent limitations in memory, computational capacity, and power efficiency. Although YOLOv11n achieves high detection accuracy, its large parameter size and computational cost hinder real-time applications. To overcome these limitations, MobileNetV3 is introduced as the backbone, leveraging its depthwise separable convolutions and efficient attention mechanisms to achieve a more compact yet expressive feature extractor. We propose a lightweight design that integrates MobileNetV3 into the YOLOv11n framework, thereby reducing model complexity while maintaining detection performance. This approach enhances the feasibility of real-time deployment in engineering inspection scenarios.

To meet the deployment requirements of intelligent tunnel lining defect recognition on embedded and real-time platforms, this study proposes a structured lightweight design scheme while preserving the multi-scale detection capability and original detection head of YOLOv11n. The modified model is designed to seamlessly interface with the original FPN/PAN and Detect Head structures, and is readily compatible with subsequent lightweight techniques such as distillation, quantization, and pruning. The details of the base architecture and lightweight modules are provided in Sections 2.2 and 2.3, respectively. The proposed modifications constitute a cohesive architectural strategy, emphasizing functional interdependence among the integrated modules. For instance, the 1×1 convolutional channel mapping layers are essential for reconciling the dimensional discrepancies between the MobileNetV3 backbone and the PAN/FPN neck. Consequently, these components were integrated and evaluated as a unified architectural update to ensure the synergistic optimization of accuracy, parameters, and inference speed.

The overall strategy is as follows: The original Backbone of YOLOv11n (including the initial Stem, sequential C2f/C3, and SPPF structures) is replaced with a MobileNetV3 series network as the new feature extractor. The original Neck (a multi-scale fusion structure of FPN/PAN style) and Detect Head (multi-scale output with regression/classification) of YOLOv11n are retained to maintain the original detection output format and inference pipeline. To further reduce computational overhead, standard convolutions in the Neck are replaced with depthwise separable convolutions, and the C2f module is substituted with its lightweight variant (C2f-Lite).

For feature scale and channel alignment, the new Backbone outputs three-scale features compatible with the YOLO framework (corresponding to strides of 8, 16, and 32, respectively), ensuring consistent input resolution for the downstream FPN/PAN. Since the native output channel dimensions of MobileNetV3 typically differ from those expected by the original Neck, channel mapping layers (1×1 convolutions) are added between the Backbone and Neck to unify the channel dimensions. Specifically, the three feature pathways are mapped to a unified set of lightweight channels (e.g., (96, 192, 384) or equivalent channels adjusted according to width_multiple) to facilitate subsequent feature fusion and network convergence. Additionally, the feature maps input to the Detect Head are designated in the order $P3 (1/8) \rightarrow P4 (1/16) \rightarrow P5 (1/32)$ to avoid scale ambiguity.

To accelerate convergence and enhance performance on limited datasets, the model is fine-tuned using pre-trained MobileNetV3 weights. Partial freezing strategies for lower-level Batch Normalization (BN) layers may also improve training stability. During initial integration, the spatial dimensions and channel numbers of the three feature pathways should be printed and verified to ensure consistency with the expectations of the Neck/Head before proceeding to large-scale training. The pseudo-code of the lightweight design scheme is presented in Algorithm 1.

Algorithm 1: YOLOv11n Lightweight Design with MobileNetV3 Backbone

Input: Input $I \in \mathbb{R}^{H \times W \times 3}$

Output: Detection results $\{class, bbox\}$

Step 1: Backbone replacement;

$F_{backbone} \leftarrow \text{MobileNetV3}(I)$

$C_3 \leftarrow$ feature map at stride = 8 (≈ 40 channels);

$C_4 \leftarrow$ feature map at stride = 16 (≈ 112 channels);

$C_5 \leftarrow$ feature map at stride = 32 (≈ 160 channels);

Step 2: Channel alignment;

$P3_{in} \leftarrow \text{Conv}_{1 \times 1}(C_3, 96)$;

$P4_{in} \leftarrow \text{Conv}_{1 \times 1}(C_4, 192)$;

$P5_{in} \leftarrow \text{Conv}_{1 \times 1}(C_5, 384)$;

Step 3: Lightweight Neck (PAN/FPN with depthwise separable convs);

$P4 \leftarrow \text{Fuse}(P4_{in}, \text{upsample}(P5_{in}))$;

$P3 \leftarrow \text{Fuse}(P3_{in}, \text{upsample}(P4))$;

$P4 \leftarrow \text{Fuse}(P4_{in}, \text{downsample}(P3))$;

$P5 \leftarrow \text{Fuse}(P5_{in}, \text{downsample}(P4))$;

Step 4: Detect Head;

Results $\leftarrow \text{DetectHead}([P3, P4, P5])$;

return Results;

2.2. Base Architecture

YOLOv11 is developed as an architectural approach that prioritizes both speed and accuracy in object detection [23]. Compared to all previous versions, YOLOv11 incorporates

several advanced techniques, such as the C3K2 module, SPFF module, C2PSA module, as well as enhancements like CSPA (Cross Spatial Pyramid Aggregation), QCSP (Quad Circular Shift Pooling), P3 (Pyramid Pooling with Three Levels), ESPP (Enhanced Spatial Pyramid Pooling), and DFP (Distribution Focal Pooling). These improvements enhance the model's ability to process spatial information while increasing inference speed. Within the YOLOv11 series, YOLOv11n is designed as a lightweight variant [24], exhibiting a significant reduction in parameter count and computational cost compared to the standard YOLOv11. Despite its compact structure, YOLOv11n maintains superior detection performance, making it particularly suitable for applications with limited computational resources or high real-time requirements. To achieve real-time defect detection for edge devices, Section 2.1 proposes replacing the Backbone with MobileNetV3 and implementing lightweight modifications to the Neck modules.

First, the Backbone network serves as the core feature extractor of YOLOv11n, transforming raw image data into multi-scale feature maps via a convolutional neural network. It primarily consists of four key components: (1) Convolutional Block (Conv): Each Conv block comprises a 2D convolutional layer, a BN layer, and a SiLU (Sigmoid-weighted Linear Unit) activation function. (2) Bottleneck Layer: Implemented as a sequence of Conv blocks in series, it supports an optional residual connection mechanism. Enabling residual connections enhances gradient flow and mitigates the vanishing gradient problem, while disabling them reduces computational overhead by omitting skip connections. (3) C2f Module: Designed based on the Cross-Stage Partial (CSP) strategy, it splits feature channels into two paths—one processed through a series of bottleneck layers, and the other directly bypassed. Feature concatenation at the end enhances diversity while reducing parameters and computational cost. (4) C3K2 Module: Compared to the C2f structure, this module employs multiple lightweight 3×3 convolutional kernels and a more compact architecture. It is flanked by Conv blocks at both ends and integrates a series of bottleneck layers in between, achieving efficient feature extraction with fewer parameters and lower complexity. A comparison of the C2f and C3K2 blocks is illustrated in Figure 2. This paper replaces the original backbone with the MobileNetV3 series as the new feature extractor, leveraging its depthwise separable convolutions, Bneck structure, and lightweight attention module to significantly reduce both parameters and computational complexity. The direct objective of this replacement is to preserve the semantic and spatial resolution of the three-scale feature representations (P3/P4/P5 with strides = 8/16/32), while mapping the output channels to a lightweight channel count compatible with the original Neck (e.g., 96/192/384 or adjusted according to width_multiple).

Second, the Neck network acts as a bridge for multi-scale feature fusion and attention enhancement, integrating two core modules: SPFF (Spatial Pyramid Fast Pooling): Aggregates contextual information through multi-scale max-pooling operations, improving small object detection capability while maintaining real-time efficiency. C2PSA: A position-sensitive attention module employing a dual-branch parallel structure, enabling adaptive feature weighting in the spatial dimension. The C2PSA module is depicted in Figure 3.

Finally, the Detect Head is responsible for generating the final predictions for object localization and classification. It integrates features from both the Backbone and Neck networks, producing detection outputs at three scales—low, medium, and high. Fine-grained features are dedicated to small object detection, while coarse-grained features capture larger objects. This multi-scale prediction strategy ensures accurate detection of objects across various sizes. To ensure the output format of the Detect Head remains consistent with the original framework, this paper retains the original FPN/PAN-style Neck/Head design. However, within the Neck, several standard convolutions are replaced with depthwise separable convolutions, and the C2f module is replaced with C2f-Lite to reduce computational

overhead. This modification aims to maintain multi-scale feature fusion capabilities while mitigating the computational bottlenecks associated with intermediate feature fusion.

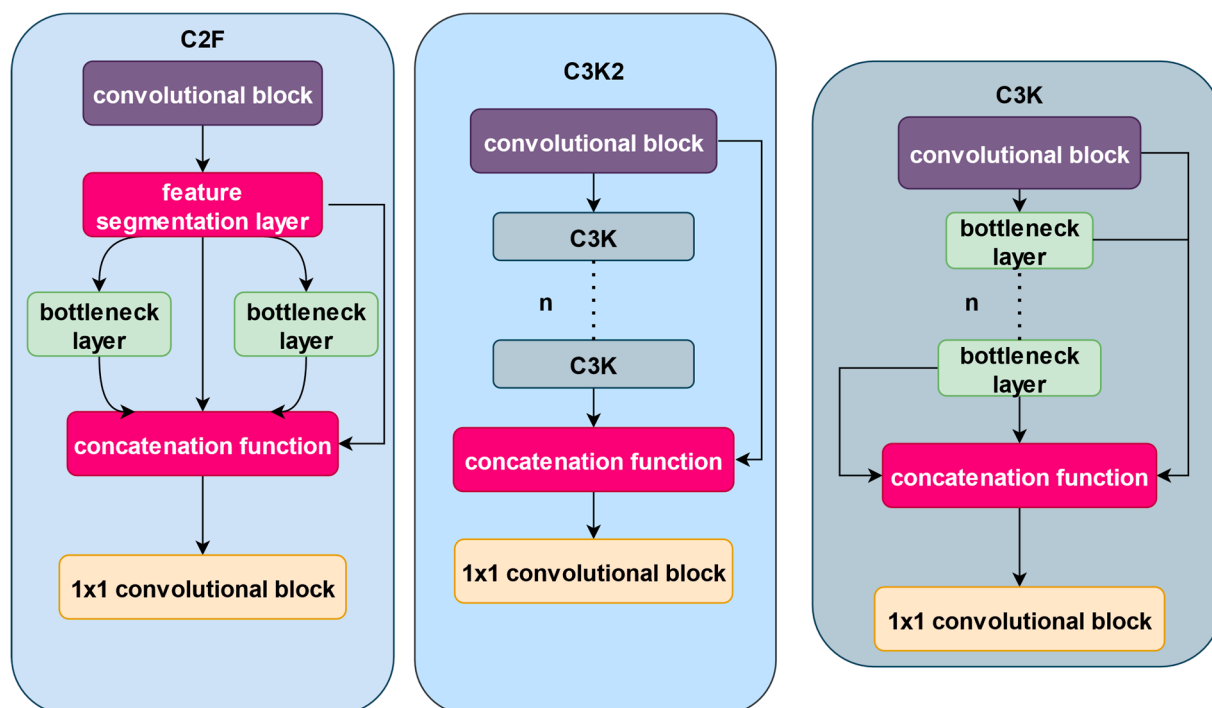


Figure 2. Comparison of the C2f and C3K2 blocks.

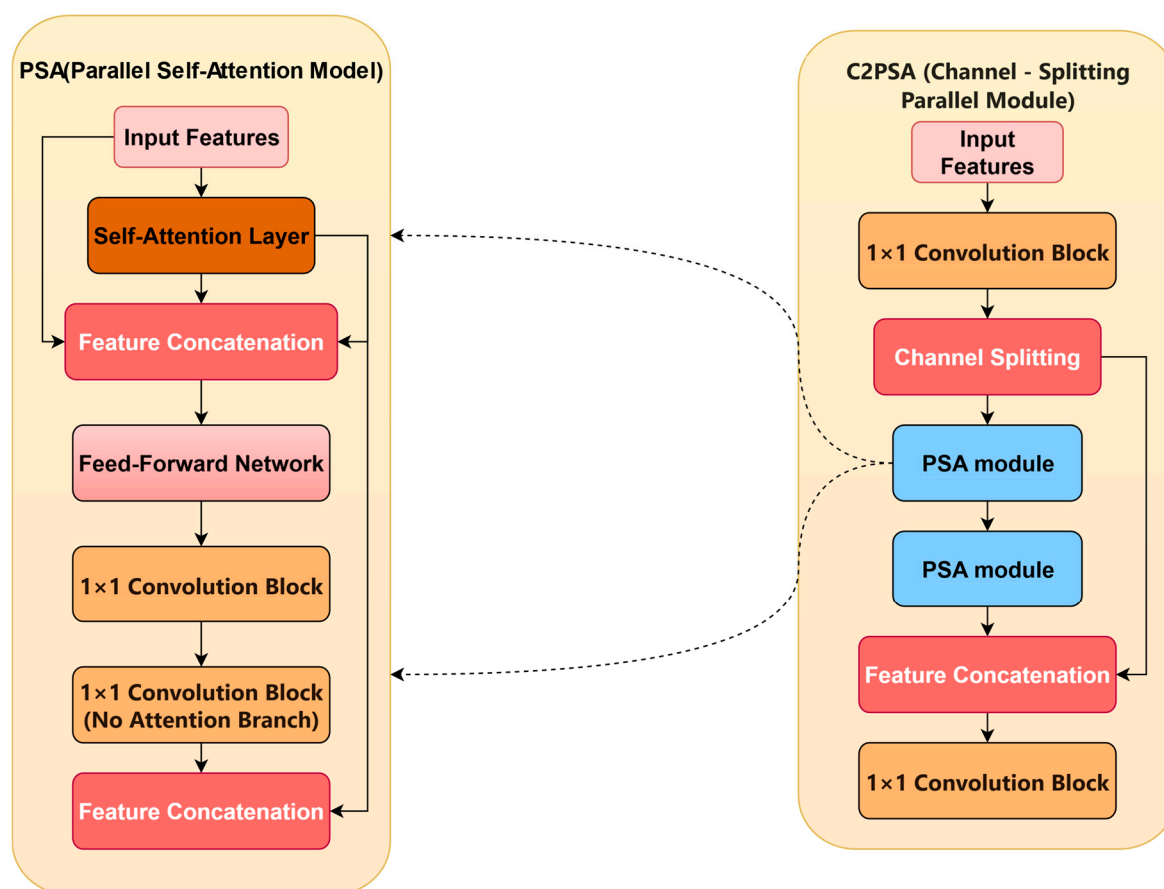


Figure 3. C2PSA module.

2.3. Improved Lightweight Modules

While several lightweight backbones exist (e.g., ShuffleNet and GhostNet), MobileNetV3 was specifically selected for this GPR-based task due to its unique integration of the SE attention mechanism within the inverted residual blocks. In the context of GPR imaging, where defect signatures are often faint and buried under complex structural clutter, the channel-wise reweighting provided by the SE module offers a distinct advantage in feature discrimination that purely convolution-shuffling architectures may lack. MobileNetV3 is developed through Neural Architecture Search (NAS) technology, automatically exploring network structures under constraints such as mobile device latency and computational resources. Specifically, it combines the large-scale framework search of MnasNet with the layer-wise channel adjustment technique of NetAdapt, thereby eliminating redundant operations and optimizing filter allocation. This hybrid design significantly reduces both parameter count and computational cost while maintaining competitive accuracy, making the network highly suitable for deployment in resource-constrained environments.

MobileNetV3 primarily utilizes the Bneck structure, depthwise separable convolutions, inverted residual structures, and an improved linear bottleneck [25]. The Bneck structure integrates the depthwise separable convolution from MobileNetV1 [26] with the inverted residual structure and linear bottleneck from MobileNetV2 [27]. Depthwise separable convolutions are particularly important for lightweight networks as they reduce computational load and parameters while maintaining equivalent feature extraction capabilities. The inverted residual structure is designed to mitigate information loss caused by computations in low-dimensional spaces, enhancing the network's ability to learn better features by first expanding and then reducing dimensions. Improvements to the linear bottleneck further enhance this network structure, reducing information loss while improving information transmission efficiency and strengthening feature representation. We will retain the critical Bneck layers of MobileNetV3 and perform channel alignment with the multi-scale outputs of the original YOLOv11n by incorporating 1×1 convolutional mapping layers.

In terms of network architecture, MobileNetV3 introduces several key improvements:

First, the average pooling layer is moved forward, and the final convolutional layer is removed. Specifically, features are first projected to a higher dimension via a 1×1 convolution, then summarized using a Global Average Pooling (GAP) layer, and finally output as K classes through a 1×1 convolution. The feature extraction is primarily performed by the 1×1 convolutional layer at a resolution of 7×7 . This modification reduces latency by approximately 11% with negligible impact on accuracy.

Additionally, MobileNetV3 introduces the H-swish activation function. Traditional activation functions like ReLU, widely adopted in various deep learning models due to their simplicity and few parameters, do not perform optimally in certain scenarios. While the Swish function can improve network accuracy, its computational cost is high, making it less suitable for deployment on resource-constrained devices. To address this, the H-swish activation function is proposed, defined as follows: This type of nonlinear activation function offers high accuracy, ease of implementation in various software and hardware systems, no precision loss during quantization, and fast execution. Although introducing nonlinearity increases overall network latency by approximately 15%, this overhead is partially offset by the acceleration benefits from the architectural changes. Replacing the previous multiplicative activation with H-Swish can eliminate the remaining overhead. With a minimal increase in computational cost, especially when introducing H-Swish in the deeper layers of the model, significant improvements in model performance can be achieved, leveraging its advantages effectively. HardSwish(x) is shown in Equation (1):

$$\text{HardSwish}(x) = x \cdot \frac{\text{ReLU6}(x + 3)}{6} \quad (1)$$

VRAM (2048 MiB), the MX450 serves as a representative proxy for evaluating the relative efficiency and computational feasibility of lightweight models before deployment on even more resource-constrained edge platforms. Crucially, due to the stringent computational constraints imposed by this 2 GB memory ceiling, standard mini-batch training configurations (e.g., Batch Size = 8 or 16) are structurally unfeasible, as the massive spatial feature maps and dense tensor backpropagation consistently precipitate catastrophic Out-of-Memory (OOM) fatal exceptions. Consequently, bounding the empirical hardware batch size to 1 (batch = 1) was an absolute operational necessity dictated by real-world physical deployment constraints to guarantee framework execution stability.

Table 2. Hyperparameter Settings for the Experimental Model.

Parameter	Value
Learning Rate Scheduler	Default Linear Scheduler
Training Epochs	100
Optimizer	Stochastic Gradient Descent (SGD)
Mosaic Augmentation	Enabled (except final stage)
Data Loading	Single-threaded
Input Resolution	640 × 640 pixels
Class Settings	Supports multi-class detection
Batch Size	batch = 1
Loss function	YOLOv8 default (Cls + Reg + Obj)

The selection of 100 training epochs was determined based on preliminary tests to ensure an optimal balance between the convergence of the lightweight architecture and computational efficiency, thereby avoiding excessive carbon footprint and time costs during the development phase. To systematically counteract the extremely high gradient variance and potential optimization instability theoretically inherent to single-sample updates (*Batch Size* = 1), a programmatic gradient accumulation strategy was strictly enforced within the deep learning pipeline. Gradients were accumulated over iterative consecutive forward steps before executing a single backward parameter adjustment. This protocol mathematically emulates the smoothing and stabilizing dynamics of a stable, large-batch optimizer update while adhering strictly to the physical 2 GB VRAM ceiling. Furthermore, the highly regularized and smooth convergence profiles visualized in the loss trajectories reflect the epoch-aggregated loss calculation methodology. By tracking and plotting the collective mean loss across complete epoch boundaries rather than erratic step-wise instantaneous fluctuations, localized stochastic step-level noise is naturally filtered out, thereby accurately capturing the macro-scale asymptotic convergence behavior of the lightweight YOLOv11n_MobileNetV3 network.

Evaluation metrics including detection accuracy, speed, and loss functions were employed to assess the performance of the tunnel lining defect detection. Detection accuracy primarily encompasses mAP@0.5, mAP@0.5:0.95, Precision, and Recall. mAP@0.5 represents the mean Average Precision calculated at an Intersection over Union (IoU) threshold of 0.5 across all classes. mAP@0.5:0.95 denotes the average mAP computed over various IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. Precision, also known as positive predictive value, is defined as the proportion of true positive predictions among all positive predictions made by the model. Recall, or sensitivity, refers to the proportion of actual positives that were correctly identified by the model.

Speed metrics primarily include inference speed and computational complexity. Inference speed is measured in FPS, while computational complexity is quantified by the number of FLOPs. Crucially, to ensure complete algorithmic reproducibility, all computational metrics were profile-tested under a standardized protocol: an input resolution

fixed at 640×640 pixels, a hardware batch size of 1 (batch = 1) to replicate real-time single-frame streaming, and executed via the built-in benchmarking utilities of the deep learning framework. To eliminate stochastic processing jitters and initial hardware allocation lag, the reported FPS values represent the statistical mean calculated across the 715 testing frames, excluding the first 20 inference cycles, which were reserved strictly as a hardware warm-up phase to stabilize GPU clock frequencies. The loss function comprises three components: bounding box loss (box_loss), classification loss (cls_loss), and distribution focal loss (dfl_loss). The box_loss decreases during training, with smaller values indicating smaller discrepancies between the predicted bounding boxes and the ground truth. The cls_loss similarly decreases during training; lower values signify better prediction of target categories. The dfl_loss reformulates the bounding box regression problem into the prediction of distributions for the four boundaries; a smaller dfl_loss indicates superior prediction performance.

3.2. Dataset Construction and Augmentation

3.2.1. Source Data and Target Categories

The target taxonomy encompasses six distinct categories: three defect types (uncompacting defects, cavities, and voids) and three structural components (steel arches, pipelines, and rebars). The characteristics of different defect and structure types and their corresponding GPR image features were analyzed. Acquiring high-quality GPR images in active tunnel environments is inherently challenging due to strict safety regulations, limited access windows, and complex site conditions. Consequently, the 382 original images collected for this study represent a significant and meticulously curated sample set of real-world tunnel lining defects. The data were acquired from a domestic tunnel site, encompassing defects like uncompacting defects, cavities, and voids, alongside tunnel structures such as rebars, steel arches, and pipelines. Note that the utilized dataset strictly preserves the native scanning resolution and natural long-tailed anomaly distribution from active operational tunnel infrastructures under standard confidentiality protocols, with no manual signal filtering applied to maintain deployment realism.

3.2.2. Rationale for Utilizing Unprocessed GPR Radargrams and Noise Implications

To fulfill the stringent real-time operational constraints of embedded tunnel scanning platforms, this study intentionally adopts an end-to-end object detection approach utilizing raw, unprocessed B-scan radargrams. Traditional signal-level geophysical preprocessing pipelines—such as time-zero correction, background subtraction via spatial filtering, and deterministic gain compensation (e.g., AGC or SEC)—typically require extensive manual parameter tuning or heavy cross-dependencies on specialized software packages. The execution of such iterative, user-dependent matrix operations introduces prohibitive computational latencies that conflict with the low-latency scanning requirements of edge hardware. By bypassing traditional filtering and presenting raw structural clutter directly to the network, the lightweight deep learning framework is compelled to autonomously extract noise-robust and clutter-invariant semantic representations.

However, omitting raw signal filtering introduces distinct trade-offs regarding model generalization and category-specific error patterns. The absence of traditional background removal allows periodic and stochastic background noise—such as continuous stratigraphic boundaries, structural layer interfaces, or water-induced environmental reflections—to propagate into the network's spatial feature maps. These artifacts frequently display geometric profiles and phase configurations that mimic the hyperbolic diffraction curves characteristic of authentic structural elements. Consequently, this unmitigated noise poses a substantial generalization challenge, directly driving a higher rate of false positives

where background clutter is misclassified as densely arranged rebar. To systematically counteract this negative impact without inducing run-time latency, specialized noise-based augmentations are utilized during training to visually disrupt these static noise textures, as detailed in Section 3.2.4.

3.2.3. Data Partitioning, Limitations, and Transfer Learning Strategy

To prevent data leakage and ensure the reliability of model evaluation, a group-wise split strategy was implemented strictly at the original-image level. Specifically, the 382 original images were first randomly partitioned into training, validation, and testing sets according to a 7:1:2 ratio. Data augmentation was then applied independently within each subset. This approach ensures that any original image and its augmented variants remain confined within the same set, thereby strictly avoiding the inclusion of training-related features in the test set. The dataset partition yielded the following allocation: the training set, utilized for model parameter learning, constitutes 70% of the baseline corpus (yielding 2500 images after expansion); the validation set, used for hyperparameter tuning and early stopping, accounts for 10% (358 images); and the test set, representing 20% of the total dataset (715 images), was reserved exclusively for the final performance evaluation. The final expanded dataset comprises a total of 3573 images.

It must be explicitly acknowledged that a baseline corpus of 382 original physical radar-grams presents an inherent sample scarcity challenge for a comprehensive 6-class object detection task, averaging approximately 64 original instances per category. Within empirical underground engineering, severe structural defects (e.g., cavities, voids) are intrinsically sparse and chronically underrepresented compared to standard structural components (e.g., rebar layers), creating a severe long-tailed data distribution. Furthermore, while pixel-level geometric and photometric transformations expand the training volume, these operations act purely as mathematical regularization mechanisms to suppress overfitting; they do not introduce novel electromagnetic boundary physics or wave-propagation signatures.

To systematically counteract this raw data limitation and stabilize the optimization convergence of the few-shot categories, a transfer learning strategy was strictly enforced. Rather than training the lightweight framework from scratch, the network weights were initialized using generalized visual feature representations pre-trained on the large-scale MS COCO benchmark. This enables the MobileNetV3 backbone to leverage pre-established feature extractors (e.g., universal edge detection, structural spatial hierarchies) prior to fine-tuning on the domain-specific GPR structural clutter, thereby substantially mitigating the optimization failure risks associated with sparse-class sample constraints.

3.2.4. Data Augmentation and Noise Regularization Protocol

To broaden the model's generalization capacity and enable the network to fit complex decision boundaries under specialized engineering data constraints, a multi-tier data augmentation protocol was executed. Spatial augmentations included horizontal and vertical flipping, center cropping with synchronized bounding box adjustment, and subsequent image resizing to expose the model to variable object perspectives and structural orientations commonly encountered in real-world scanning coordinates. Photometric augmentations involved random brightness variation (± 120 pixel delta, range 0.5–1.5), normalized contrast adjustment (± 0.47), and controlled saturation modification within [0.0, 1.0]. Crucially, to counter the negative impacts of the un-preprocessed background noise detailed in Section 3.2.2, specialized noise-based perturbations were injected as a deliberate regularization mechanism: Gaussian noise (mean = 0, std = 0.1) was used to emulate low-light tunnel environments, while salt-and-pepper noise replaced 10–30% of pixels. This explicit noise injection visually fractures the continuous, deceptive clutter textures inherent in raw

radargrams, thereby compelling the lightweight YOLOv11n_MobileNetV3 framework to suppress background variance and rely predominantly on robust, high-amplitude target boundaries for object-localized regression.

3.3. Engineering Rationale for the Proposed Structural Combination

To ensure successful automated health monitoring along highly active transport corridors, object detection frameworks must be optimized not merely for theoretical algorithmic complexity, but for practical field deployability. Consequently, this study intentionally positions the developed YOLOv11n_MobileNetV3 framework as an edge-oriented engineering lightweight deployment solution tailored to withstand the distinctive operational constraints of underground GPR inspections. The architectural symbiosis of MobileNetV3, depthwise separable convolutions, and the streamlined C2f-Lite modules is mathematically and physically justified through two primary engineering dimensions:

First, raw, un-preprocessed B-scan radargrams are characteristically contaminated by high-frequency system noise and continuous stratigraphic reflections that frequently mask faint structural defect signatures. Standard convolutional networks typically succumb to severe overfitting when trained on such noisy, small-sample matrices. To counteract this, the selected MobileNetV3 backbone selectively retains the SE attention mechanism within its deep layer feature maps. This configuration allows the network to execute dynamic channel-wise recalibration, effectively dampening static background clutter while selectively amplifying low-contrast, high-amplitude target boundaries characteristic of internal lining voids and cavities.

Second, practical on-site tunnel scanning demands high-velocity, real-time multi-task parsing synchronized with the continuous movement of inspection vehicles, enforcing a non-negotiable computational latency cap. The integration of depthwise separable convolutions and C2f-Lite topological stages directly compresses the cross-channel feature extraction pathways. Compared to the baseline architecture, this combined re-engineering achieves a parameter contraction down to 2.26 M and a 25.4% reduction in floating-point operations (dropping to 4.7 GFLOPs). By maximizing structural parameter efficiency, this framework establishes robust structural and computational feasibility for low-latency synchronization on power-constrained embedded systems, offering a highly viable, real-world trade-off between absolute detection sensitivity and edge execution speed.

Furthermore, the deliberate selection of MobileNetV3 as the foundational backbone—over alternative lightweight paradigms such as GhostNet, ShuffleNet, or EfficientNet-Lite—is governed by stringent hardware execution realities and domain-specific GPR feature requirements. Specifically, while GhostNet minimizes FLOPs via cheap linear operations, its frequent cross-channel splits and concatenations introduce high Memory Access Costs (MAC), yielding execution bottlenecks on low-power edge processors. Similarly, the ‘channel shuffle’ operations in the ShuffleNet series demand intensive tensor reshaping and memory permutations, which impose severe runtime latency on microcontrollers lacking specialized memory-indexing hardware. EfficientNet-Lite, despite its high accuracy, preserves deep topological scaling that leads to elevated peak visual memory footprints, risking OOM faults within the hard 2 GB VRAM boundary of the edge proxy platform. Conversely, MobileNetV3, synthesized via hardware-aware NAS, optimizes actual physical latency. Crucially, its integrated lightweight SE attention mechanism dynamically calibrates channel dependencies to isolate faint hyperbolic diffraction boundaries out of hostile sub-surface clutter, while its H-swish activation eliminates expensive exponential calculations, delivering a highly viable, edge-oriented engineering solution.

3.4. Experimental Design

The experiments are organized to validate the proposed lightweight architecture in terms of detection accuracy, computational efficiency, module-wise contribution, and robustness. (1) Qualitative detection visualization: We will show example detections on the test set (bounding boxes + confidences) to demonstrate real-case performance and failure modes. (2) Overall quantitative performance comparison: Compared four detectors—YOLOv5 v6.0, YOLOv8n, YOLOv11n and the proposed YOLOv11n_MobileNetV3—using mAP@0.5, mAP@0.5:0.95, FLOPs, parameter count, and FPS. Results reported in tables to quantify the accuracy–efficiency trade-offs. (3) Category-wise analysis: Per-class PR curves and AP values will be plotted and discussed (rebars, uncompacting defects, cavities, steel arch, pipelines, voids). A normalized confusion matrix and F1-score curves will be used to analyze which classes are confused and how detection quality varies with confidence thresholds. (4) Ablation study: A focused comparison between the baseline YOLOv11n and the improved YOLOv11n_MobileNetV3 will be conducted. This isolates the impact of the MobileNetV3 & lightweight changes on each defect type. (5) Parameter analysis: Measure and compare parameter counts, FLOPs and FPSs across models to support claims about deployability on resource-limited hardware.

4. Performance Discussion and Analysis

4.1. Detection Results and Performance Evaluation

(1) Tunnel Lining Defect Detection Results

The detection results of the YOLOv11n model are shown in Figure 5. Each bounding box indicates a detected defect or structure along with its corresponding confidence score. As observed in Figure 5, the model effectively identifies and localizes multiple targets in these representative cases. However, as further indicated by the quantitative metrics, detection challenges remain in highly cluttered regions. In Figure 5, the YOLOv11n model successfully detected and differentiated between a “cavity” (confidence 0.78) and multiple “steel arch” instances (confidence mostly ≥ 0.71), demonstrating its capability to recognize multiple structural targets. Figure 5 shows that the YOLOv11n model can identify multiple “rebar” instances (confidences 0.73, 0.69, 0.68), indicating strong discriminative ability for densely distributed targets of the same category. A comprehensive observation of Figure 5 reveals that despite challenges such as blurring and overlapping objects, YOLOv11n achieved accurate identification and localization, showcasing its robust performance in complex scenarios.

As observed in Figure 5, the model effectively identifies and localizes representative defect targets. However, detection challenges remain in highly cluttered regions. Representative failure cases observed during inference primarily manifest in two visual ways: (1) false positive bounding boxes generated in dense, noisy background patches misidentified as structural targets, and (2) false negative omissions where faint, deep-seated defects fail to trigger detection bounding boxes. The quantitative breakdown and physical mechanisms behind these specific failure modes are analyzed in detail in the subsequent confusion matrix discussion.

(2) Normalized Confusion Matrix

The normalized confusion matrix of the YOLOv11n model is presented in Figure 6. Analysis of the matrix indicates that the correct detection rate for rebars is 93%, with 7% misclassified as background. The correct detection rate for uncompacting defects is 84%, with a small portion misclassified as cavities (8%). Cavities are correctly detected at a rate of 92%, with a minor misclassification as uncompacting defects (2%) and steel arches (2%). Steel arches are detected with an accuracy of 95%, with some misclassified as background (approximately 5%). Pipelines are detected with 90% accuracy but are prone

to misclassification as cavities (7%). Voids achieve a detection accuracy of 96%, with a small fraction (4%) mistaken for background. The background class itself has a relatively low correct detection rate, with a significant portion (approximately 74%) misclassified as rebars.

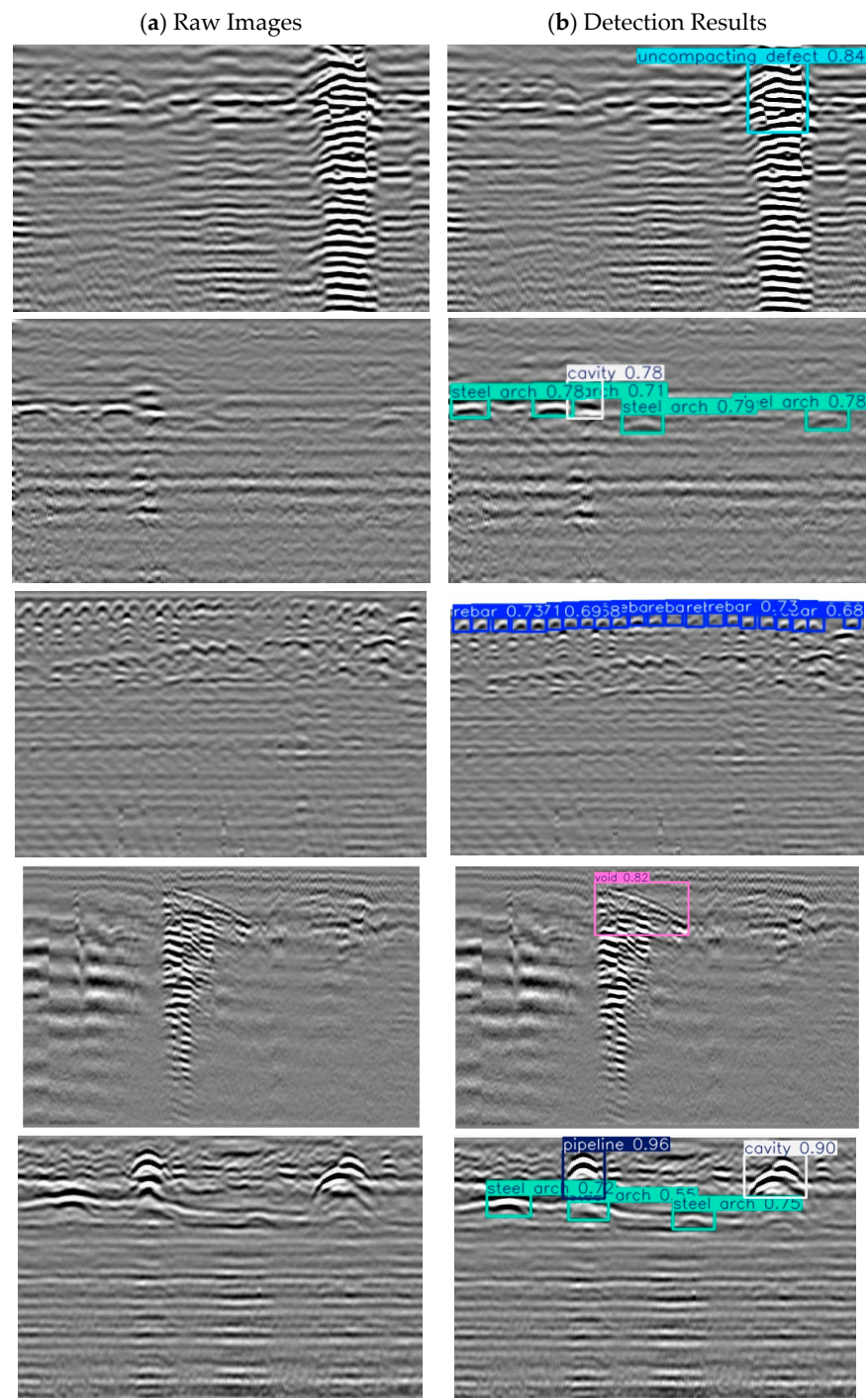


Figure 5. YOLOv11n detection results on the test set.

The YOLOv11n model demonstrates high accuracy in identifying “void” (96%) and “steel arch” (95%) categories. However, the recognition accuracy for “uncompacting defect” and the “background” category is relatively low. Notably, a substantial number of background instances are misclassified into other categories, indicating a need for improved discrimination of the background class. This could be addressed by incorporating more

background samples or further optimizing the YOLOv11n model. Additionally, reducing the misclassification of other categories as background should also be prioritized.

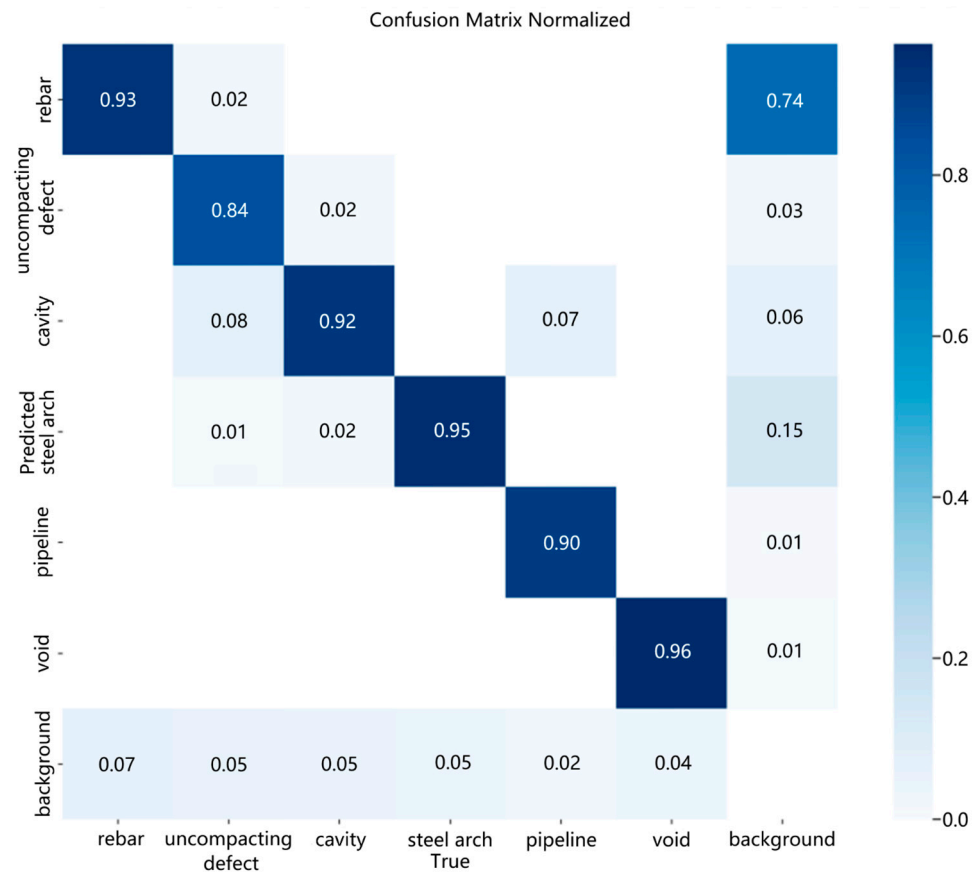


Figure 6. Normalized confusion matrix of YOLOv11n.

A detailed FP/FN breakdown reveals specific error patterns. The FP rate for “rebar” is notably high because certain background noise—such as structural layer reflections or water-induced clutter—exhibits periodic patterns similar to the hyperbolic echoes of rebar. The FN rate for “uncompacting defects” (16%) is primarily attributed to their subtle radar signatures, which can be easily obscured by background clutter or confused with “cavities” due to overlapping signal intensity ranges. Furthermore, 7% of pipelines are misclassified as cavities because both present as high-amplitude arc reflections in GPR B-scan images.

(3) F1-Score Curve Analysis

The F1-score curves of the YOLOv11n model are shown in Figure 7. Analysis of the curves reveals that for rebars, a high F1-score is achieved even at lower confidence thresholds. The F1-score initially increases and then decreases as the confidence threshold rises, forming a characteristic curve. The F1-curve for uncompacting defects is unstable, with low F1-values at lower confidence levels, suggesting poorer YOLOv11n model robustness for this category, likely due to significant intra-class variation and differences from other categories. For cavities, the F1-curve remains relatively stable, indicating consistent performance for this class, with high F1-values at medium confidence levels.

For steel arches, the F1-curve peaks at a medium confidence level, indicating optimal prediction accuracy at this threshold. For pipelines, the F1-curve begins to rise at lower confidence levels, suggesting that instances of this category can be effectively distinguished from others even with lower confidence predictions. For voids, the F1-score remains high at medium confidence levels, reflecting strong predictive accuracy within this confidence range. The composite F1-score for all classes reaches 0.91 at a confidence threshold of

0.321 (represented by the thick blue line), indicating superior overall model performance at this operating point.

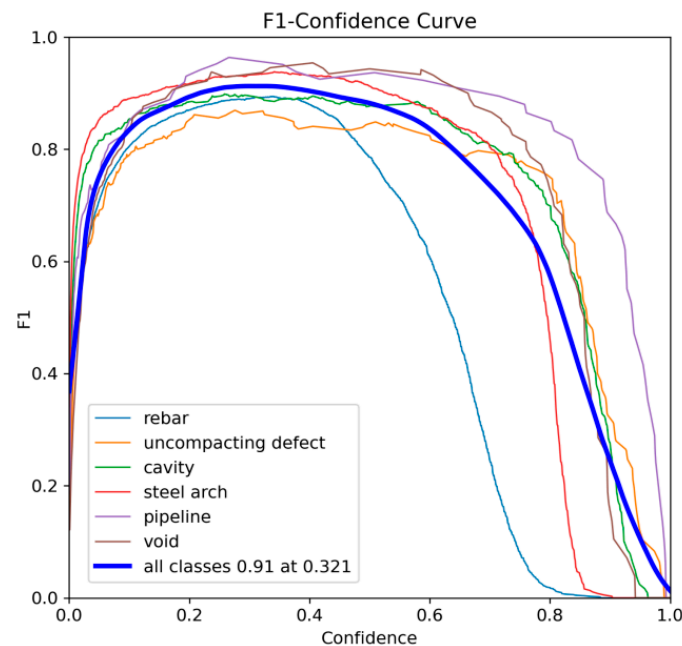


Figure 7. F1-curves of YOLOv11n.

The optimal working point for the YOLOv11n model is around a confidence threshold of 0.321, where the composite F1-score for all classes is maximized (0.91). Pipelines and voids maintain high F1-scores across most confidence intervals, whereas the F1-score for uncompact defects is relatively lower and more sensitive to confidence threshold variations. The F1-scores for “steel arch” and “cavity” categories reach satisfactory levels at medium confidence, indicating reliable discrimination for these classes.

(4) PR Curve Analysis

The Precision–Recall (PR) curves of the YOLOv11n model are shown in Figure 8. The curves indicate that the AP for rebars is 0.940. The curve shows that precision remains high across most recall levels, demonstrating good performance. The AP for uncompact defects is 0.912. While precision is high at low recall rates, it decreases rapidly as recall increases, indicating weaker model performance for this category. The AP for cavities is 0.957. The curve is generally stable, with precision maintained at a high level across a wide range of recall values, denoting excellent performance.

The AP for steel arches is 0.973. The curve shows very high precision across all recall rates, indicating a strong ability to identify this category. The AP for pipelines is 0.974. Precision remains high at medium recall levels but decreases slightly at higher recall rates. The AP for voids is 0.977. Precision is relatively high at low to medium recall rates but decreases as recall increases. The mean Average Precision for all classes at IoU = 0.5 (mAP@0.5) is 0.955 (represented by the blue line), indicating strong overall network performance across all categories.

The results show that the YOLOv11n model exhibits strong recognition capabilities for voids (AP = 0.977) and pipelines (AP = 0.974). However, the detection accuracy for uncompact defects is relatively lower (AP = 0.912), suggesting significant room for optimization for this category. The steel arch category maintains high precision even at high recall rates, demonstrating good model robustness for this class. The mean Average Precision (mAP@0.5) of 0.955 across different categories indicates effective detection performance.

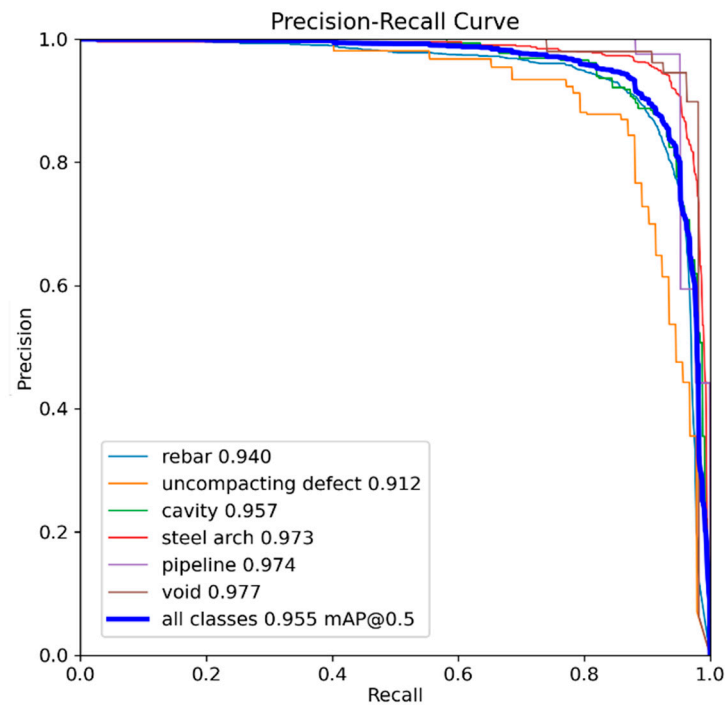


Figure 8. PR curves of YOLOv11n.

(5) Training Result Analysis

The curves of various performance metrics during training are shown in Figure 9. Analysis of the curves indicates that both training and validation losses exhibit a decreasing trend, demonstrating that the YOLOv11n model effectively learns the characteristics of the training samples and generalizes well on the validation set. Precision, recall, and average precision (mAP50 and mAP50-95) all improve with increasing training iterations, indicating progressively enhanced performance on the detection task. Furthermore, the synchronization between the training loss and validation loss curves provides strong empirical evidence that the model has not suffered from severe overfitting. Instead of memorizing specific pixel patterns, the model successfully learned generalized GPR signatures. The smoothness of the loss curves suggests a stable training process without severe overfitting or underfitting. The continuous improvement in model performance metrics indicates an increasingly better fit to the data and effective performance on the detection task. It is worth noting that while the performance curves exhibit a marginal upward trend toward the 100th epoch, the incremental gains in mAP and the reduction in loss values have stabilized significantly. This suggests that the model has reached a state of practical convergence, where the current performance is representative and sufficient for the requirements of real-time tunnel defect identification.

4.2. Comparative Analysis of Experimental Results

4.2.1. Performance Comparison of Lightweight Models

(1) Curve Analysis

Figure 10 presents the performance metric curves of different baseline models. All mAP50-95, mAP50, Precision, and Recall curves show upward trends, indicating that detection performance improves as training progresses. Among them, the YOLOv11n model consistently achieves the highest values across all metrics, reflecting superior detection accuracy, precision, and recall at various IoU thresholds. This demonstrates that YOLOv11n maintains better overall detection effectiveness and generalization throughout training.

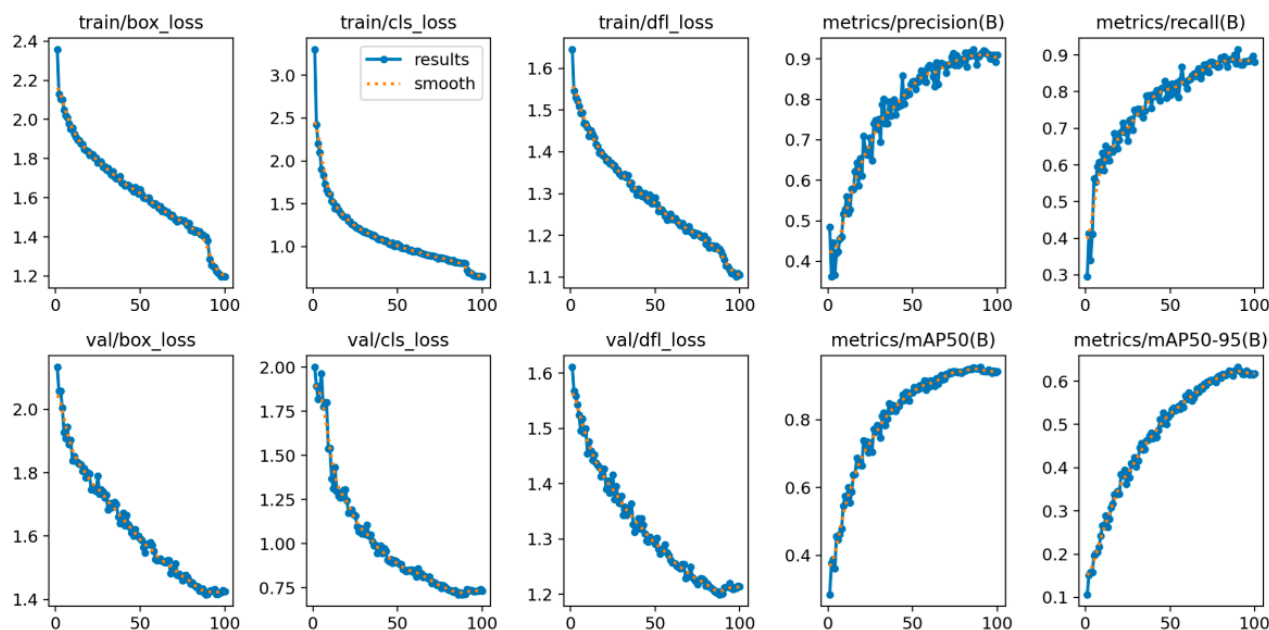


Figure 9. Curves of performance metrics during the training process.

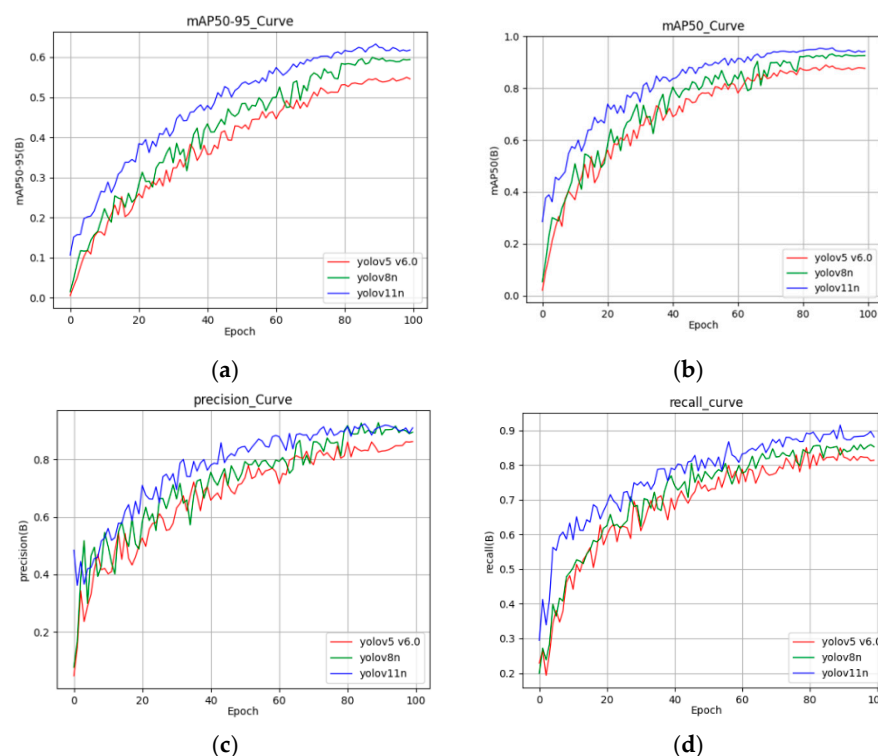


Figure 10. Performance metric curves under different baseline models. (a) mAP50-95_Curve, (b) mAP50_Curve, (c) precision_Curve, (d) recall_Curve.

(2) Performance Metrics Comparison

(i) Overall Performance:

As shown in Table 3, the performance of different object detection models is compared. The YOLOv5 v6.0 model achieves an mAP@0.5 of 0.878 and an mAP@0.5:0.95 of 0.55. YOLOv8n achieves an mAP@0.5 of 0.929 and an mAP@0.5:0.95 of 0.60. YOLOv11n achieves an mAP@0.5 of 0.955 and an mAP@0.5:0.95 of 0.634. YOLOv11n surpasses YOLOv8n by 2.6% in mAP@0.5, with lower FLOPs, indicating higher computational efficiency and superior performance. Compared to YOLOv5 v6.0, YOLOv11n improves mAP@0.5 by

7.7% and mAP@0.5:0.95 by 8.4%, while also achieving a higher FPS, indicating faster inference speed alongside maintained higher accuracy.

Table 3. Performance of different object detection models.

Model	mAP@0.5	mAP@0.5:0.95	FLOPs (G)	FPS
YOLOv5 v6.0	87.8%	55.0%	5.8	49
YOLOv8n	92.9%	60.0%	6.8	45
YOLOv11n	95.5%	63.4%	6.3	43

(ii) Category-wise Performance:

Figure 11 shows the PR curves of the models under different baseline models. Analysis of the curves reveals that for rebars, the mAP@0.5 is 0.919 under YOLOv5 v6.0, 0.946 under YOLOv8n, and 0.940 under YOLOv11n. For uncompacting defects, the mAP@0.5 under YOLOv11n (0.912) is significantly higher than that under YOLOv8n (0.798) and YOLOv5 v6.0 (0.729). YOLOv11n achieves an mAP@0.5 of 0.957 for the cavity category and 0.973 for the steel arch category, higher than the other two models.

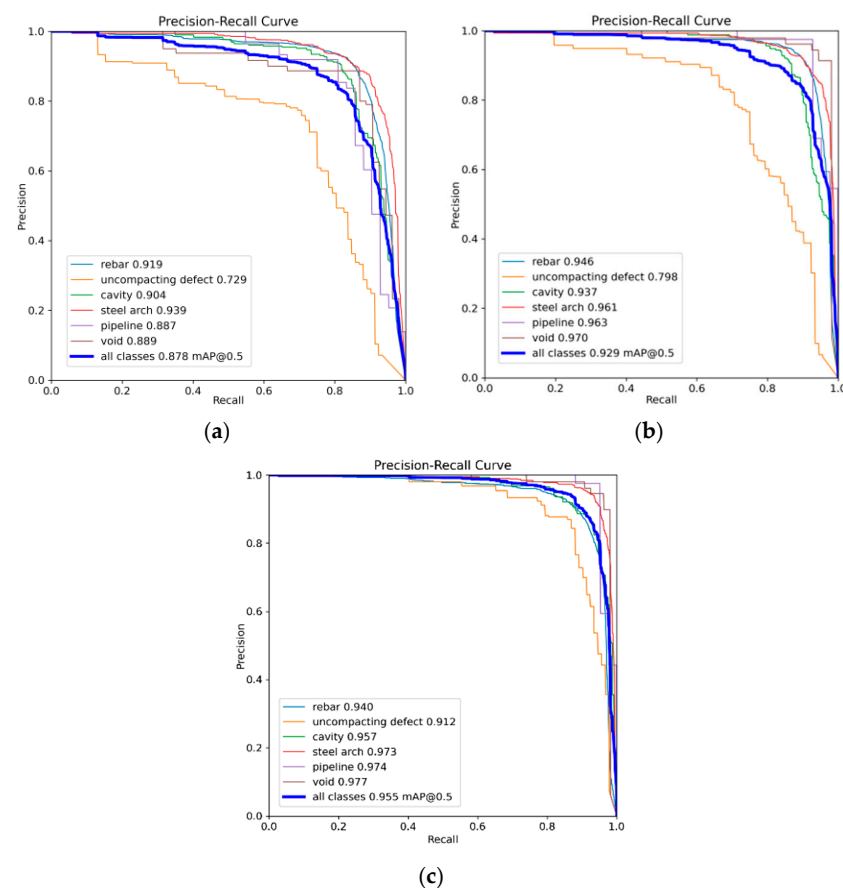


Figure 11. PR curves of the models under different baseline models. (a) YOLOv5 v6.0, (b) YOLOv8n, (c) YOLOv11n.

In summary, YOLOv11n outperforms YOLOv5 v6.0 across all performance metrics, with a particularly noticeable advantage in mAP@0.5:0.95, indicating better overall detection performance and effectiveness across various IoU thresholds. While YOLOv8n performs better than YOLOv5 v6.0, it does not achieve the same level of performance as YOLOv11n in certain categories. Therefore, model selection should consider the trade-off between performance and efficiency. If higher accuracy is required, YOLOv11n is preferable; if faster inference speed is prioritized, YOLOv8n or YOLOv5 v6.0 could be chosen.

4.2.2. Ablation Study

(1) Detection Results of Different Defects Using the Improved Model

Figure 12 presents the detection results for uncompacting defects, cavities, steel arches, rebars, voids, and pipelines using the baseline YOLOv11n model and the improved YOLOv11n_MobileNetV3 model.

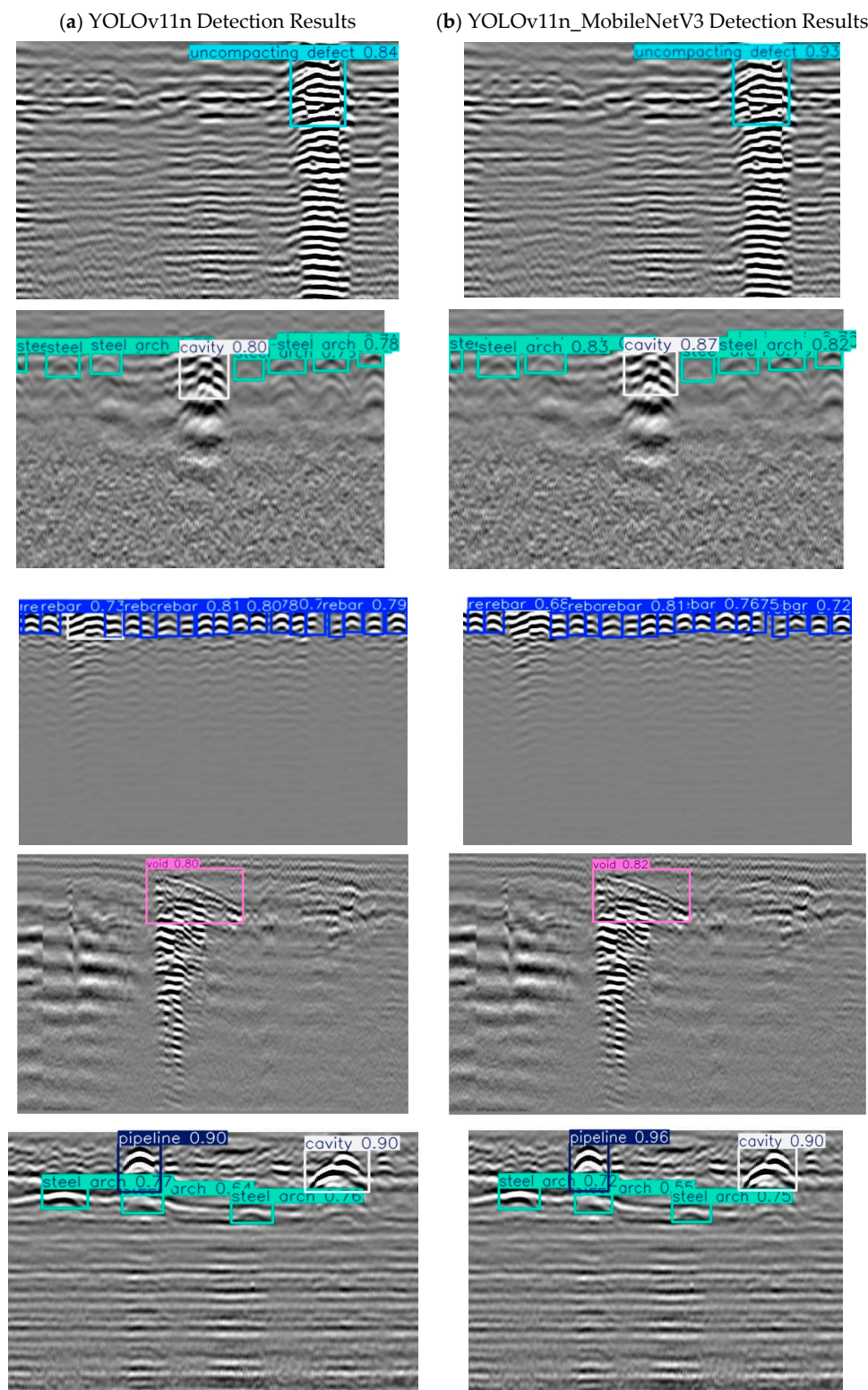


Figure 12. Detection results under different models.

(2) Performance Comparison

Table 4 compares the training results of YOLOv11n and YOLOv11n_MobileNetV3, where each pair of values in Precision, Recall, AP@0.5, and AP@0.5:0.95 represents the two models. Overall, YOLOv11n_MobileNetV3 achieves higher precision (0.923 vs. 0.891). For uncompact defects, its precision (0.867) and recall (0.870) are slightly lower. For cavities, it shows higher precision (0.939 vs. 0.887) and similar recall. For steel arches, precision improves (0.963 vs. 0.929), though recall decreases (0.880 vs. 0.938). For voids, precision also increases (0.926 vs. 0.910), with a slight drop in recall (0.922 vs. 0.963). Overall, YOLOv11n_MobileNetV3 prioritizes higher precision and better suppression of false positives, while the baseline YOLOv11n achieves stronger overall recall and robustness in distinguishing true samples. While individual module metrics are not reported separately, the overall results demonstrate that the gains in computational efficiency (4.7 GFLOPs) are primarily driven by the backbone replacement and lightweight neck modules, while the maintenance of high detection accuracy (94.4% mAP) is attributed to the selective use of attention mechanisms and optimized channel alignment. This joint contribution validates the effectiveness of the proposed integrated lightweight scheme for practical tunnel inspection.

Table 4. Training result data for the baseline YOLOv11n and YOLOv11n_MobileNetV3 models.

Category	Instances	Precision (P)	Recall (R)	AP@0.5	AP@0.5:0.95
Rebar	2085	0.891/0.923	0.894/0.879	0.940/0.941	0.443/0.422
uncompact defect	92	0.875/0.867	0.859/0.870	0.912/0.897	0.626/0.658
Cavity	244	0.887/0.939	0.899/0.889	0.957/0.960	0.686/0.677
Steel arch	747	0.929/0.963	0.938/0.880	0.973/0.972	0.631/0.646
Pipeline	42	0.975/0.890	0.940/0.905	0.974/0.947	0.694/0.673
Void	54	0.910/0.926	0.963/0.922	0.977/0.948	0.722/0.671

Figure 13 shows the PR curves for the baseline and improved models. While the baseline YOLOv11n achieves a slightly larger overall area under the curve (mAP@0.5 of 0.955 compared to 0.944), the proposed YOLOv11n_MobileNetV3 demonstrates highly competitive and robust performance. It can be observed that the PR curve of YOLOv11n_MobileNetV3 remains relatively stable in the high-recall region, indicating its ability to maintain acceptable precision even at higher recall thresholds. More importantly, it achieves this highly comparable PR curve profile while significantly reducing computational complexity, confirming a successful accuracy–efficiency trade-off for resource-constrained deployments.

4.2.3. Parameter Count and Computational Efficiency Evaluation

As shown in Table 5, the performance comparison of different object detection models is as follows. Regarding parameter count and computational efficiency, the model based on the YOLOv5 v6.0 baseline framework has only 2.19 M parameters and relatively low computational complexity (5.8 GFLOPs), making it suitable for resource-constrained scenarios while achieving a high inference speed (49 FPS). YOLOv8n increases the parameter count (2.69 M) and computational complexity (6.8 GFLOPs), raising the hardware requirements for running the network structure. It is, therefore, less suitable for deployment in resource-limited environments, although it offers improved detection accuracy.

YOLOv11n achieves a balance between model size and performance, with 2.58 M parameters and 6.3 GFLOPs, positioned between YOLOv5 v6.0 and YOLOv8n. By integrating the MobileNetV3 module, YOLOv11n_MobileNetV3 reduces parameters to 2.26 M and FLOPs to 4.7, improving computational efficiency and achieving 45 FPS while maintaining

strong performance. In the field of real-time engineering inspection, FPS and GFLOPs are widely recognized as primary proxies for on-device latency and computational intensity. Given the targeted deployment on the NVIDIA GeForce MX450 platform, the achieved 45 FPS confirms the model's suitability for continuous tunnel scanning, effectively balancing the constraints of hardware memory and inference speed without the need for excessive power consumption. In detection accuracy, YOLOv5 v6.0 performs the worst ($\text{mAP}@0.5 = 87.8\%$, $\text{mAP}@0.5:0.95 = 55.0\%$), YOLOv8n performs better (92.9% , 60.0%), and YOLOv11n achieves the highest accuracy (95.5% , 63.4%) with moderate complexity. YOLOv11n_MobileNetV3 maintains high accuracy (94.4% , 62.4%) and greater stability across IoU thresholds while minimizing computational cost. In inference speed, YOLOv5 v6.0 leads with 49 FPS due to its simpler design, while YOLOv8n and YOLOv11n deliver 45 and 43 FPS, respectively. YOLOv11n_MobileNetV3 combines 45 FPS speed with high precision and low complexity, offering the most balanced trade-off. Overall, YOLOv11n ensures top accuracy, whereas YOLOv11n_MobileNetV3 provides optimal balance between detection performance, efficiency, and speed, making it the preferred model for defect detection in this study. It is explicitly acknowledged that YOLOv11n_MobileNetV3 trades a minor decrease in $\text{mAP}@0.5$ (1.1%) for a substantial 25.4% reduction in FLOPs. In resource-constrained applications, this constitutes a highly successful optimization.

Table 5. Performance comparison of different object detection models.

Model	Parameters (M)	FLOPs (G)	$\text{mAP}@0.5$	$\text{mAP}@0.5:0.95$	FPS
YOLOv5 v6.0	2.19	5.8	87.8%	55.0%	49
YOLOv8n	2.69	6.8	92.9%	60.0%	45
YOLOv11n	2.58	6.3	95.5%	63.4%	43
YOLOv11n_MobileNetV3	2.26	4.7	94.4%	62.4%	45

Note: All models listed in Table 5 were evaluated under identical measurement conditions to guarantee fair comparability: input size 640×640 pixels, batch = 1, hosted on the NVIDIA GeForce MX450 (2 GB VRAM) edge-precursor proxy platform. FLOPs represent the hardware-agnostic computational complexity, and FPS denotes the macro-scale average inference speed after filtering out initial framework warm-up latencies.

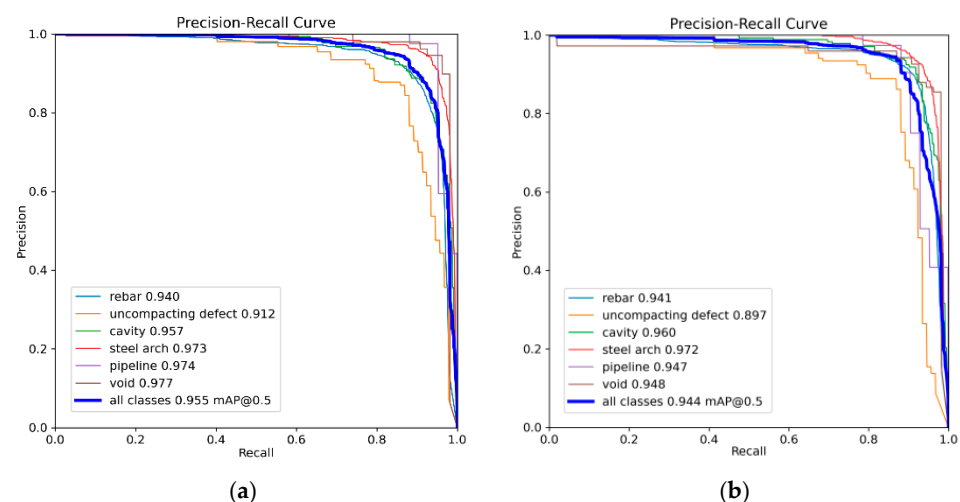


Figure 13. PR curves for the baseline and improved models. (a) YOLOv11n, (b) YOLOv11n_MobileNetV3.

4.3. Empirical Discussion

To bridge the gap between theoretical lightweight algorithmic efficiency and empirical structural health monitoring, it is vital to contextualize the developed framework within the real-world operational lifecycle of underground civil infrastructure. The case study executed along the active transport corridor highlights a clear, validated trajectory for

transitioning from traditional expert-dependent diagnostics toward automated structural edge intelligence.

Engineering Significance: Modern tunnel asset management faces an urgent operational requirement for rapid, high-throughput sub-surface parsing. Traditional manual GPR interpretation acts as a critical operational bottleneck due to inherent human subjectivity, high variance in expert proficiency, and significant analytical delays. The empirical significance of this case study lies in validating that an end-to-end, multi-class intelligent detector can simultaneously isolate highly sparse, safety-critical internal anomalies (e.g., cavities, lining voids, and uncompacting concrete matrices) and pervasive structural reinforcement components (e.g., steel arches, dense rebar layers, and utility pipelines) out of highly heterogeneous sub-surface media. By converting erratic raw reflection waveforms into standardized, real-time spatial bounding coordinates, the framework establishes a reliable, objective foundation for large-scale digitized infrastructure auditing.

Field Implementation Challenges: Bypassing controlled laboratory conditions and deploying deep learning networks onto on-site inspection vehicles introduces severe physical and computational friction. First, to ensure instantaneous processing velocity, traditional signal-level geophysical preprocessing pipelines—such as time-zero corrections, deterministic gain controls, and interactive spatial filtering—are intentionally omitted. Consequently, the feature extraction network is forced to withstand hostile stratigraphic clutter, continuous layer interface reflections, and water-induced phase shifts that closely mimic authentic structural hyperbolic diffractions, severely threatening classification stability. Second, real-world field scanning operates under rigid hardware redlines. When high-resolution radar tensors are backpropagated through standard heavy networks on power-constrained mobile edge platforms, the restricted video memory allocation routinely precipitates catastrophic OOM fatal errors or prohibitive inference latencies. Bounding the structural complexity to 4.7 GFLOPs via the customized MobileNetV3 backbone and depthwise separable convolutions represents a necessary, hardware-aligned trade-off to guarantee macro-scale runtime stability under these uncompromising field constraints.

Practical Implications for Inspection and Maintenance: The operational superiorities of the optimized YOLOv11n_MobileNetV3 framework translate directly into tangible structural maintenance advantages. Operating at an inference velocity of 45 FPS with a contracted footprint of 2.26 M parameters enables the model to be successfully flashed onto compact vehicle-mounted edge microcontrollers for real-time streaming. This capability drastically reduces the track-possession windows required for active corridor inspections, minimizing public traffic disruptions along vital civic transit arteries and mitigating hazardous exposure for field personnel. Crucially, the standardized multi-class telemetry allows asset managers to transition from traditional, reactive structural repairs into an advanced preventative maintenance paradigm. By compiling spatial defect distributions over successive inspection cycles, engineering teams can precisely monitor the structural degradation trajectory of discrete lining cavities and voids, optimizing budgetary and resource allocations before catastrophic structural failures manifest.

Edge-Computing Platform Compatibility and Optimization Pathways: To further evaluate the practical viability of the proposed YOLOv11n_MobileNetV3 framework on representative edge-computing hardware, its architectural compatibility with platforms such as NVIDIA Jetson Nano, Jetson Xavier, and Raspberry Pi is thoroughly analyzed. Although physical hardware-in-the-loop flashing is slated for the next phase of our engineering field testing, the structural optimizations introduced—namely the MobileNetV3 backbone and the streamlined C2f-Lite Neck—mathematically reduce the FLOPs by 25.4%, providing a solid cross-platform mathematical guarantee for edge migration. For the NVIDIA Jetson series (e.g., Nano and Xavier), the framework is highly compatible with

the NVIDIA TensorRT optimization pipeline via ONNX compilation, which can leverage half-precision (FP16) or INT8 quantization to fully exploit Tensor Cores and maximize hardware throughput. For CPU-bound, ultra-low-power platforms like the Raspberry Pi, deploying the model through lightweight inference engines such as NCNN or TensorFlow Lite (TFLite) ensures minimal memory footprints and efficient sequential single-frame processing. This structural adaptivity guarantees that the theoretical efficiency gains documented in this study can directly translate into operational acceleration across diverse edge computing architectures.

5. Conclusions and Future Work

This study proposes YOLOv11n_MobileNetV3, a lightweight intelligent recognition model tailored for multi-category tunnel lining defects using GPR data. By integrating a MobileNetV3 backbone, depthwise separable convolutions, and a C2f-Lite structure, the model effectively addresses the challenge of deploying high-precision deep learning algorithms on resource-constrained embedded devices. Experimental results confirm that the proposed approach achieves a superior balance between detection accuracy and computational efficiency compared to standard baselines, offering a highly practical solution for real-time engineering inspection.

Despite these strengths, several limitations remain. First, to maximize edge-inference efficiency, the model operates directly on raw B-scan radargrams without geophysical preprocessing. This avoids operational latency but leaves the network susceptible to stratigraphic clutter, causing background false positives that are not yet isolated by customized loss configurations. Second, the small baseline dataset (382 original images) relies on geometric augmentations that lack electromagnetic wave physics, limiting the AP for highly sparse defect categories like cavities and voids. Third, exhaustive empirical evaluations against alternative lightweight backbones (e.g., ShuffleNet and GhostNet) are not fully documented. Fourth, actual physical performance on authentic edge hardware (e.g., Jetson Nano and Raspberry Pi) requires direct validation beyond theoretical FLOPs/FPS proxies. Finally, a more granular ablation study is required to isolate the precise quantitative contributions of each integrated sub-module.

Future work will focus on addressing these gaps through targeted methodology upgrades. First, to suppress background false alarms without sacrificing latency, we will integrate real-time digital signal processing (DSP) filtering into the data ingestion pipeline, complemented by specialized background suppression losses (e.g., hard-negative mining) and cost-sensitive weighting to counteract category imbalance optimization biases. Second, physical deployment and hardware-algorithm co-design will be executed on authentic edge platforms alongside extensive benchmarks against diverse lightweight architectures. Third, to overcome the small-sample bottleneck, electrodynamics-driven data synthesis using finite-difference time-domain (FDTD) simulation (e.g., gprMax) and meta-learning will be employed to enrich the corpus with genuine wave-propagation signatures. Concurrently, a rigorous statistical evaluation framework computing 95% confidence intervals for class-wise AP will be introduced to formally quantify sparse-category stability. Finally, granular ablation studies and hyperparameter tuning will be refined to further optimize model convergence.

Author Contributions: Conceptualization, Y.L. and H.Z.; Methodology, Y.L.; Software, Y.L.; Investigation, Y.L.; Data curation, Y.L. and Y.W.; Writing—original draft, Y.L.; Writing—review & editing, Y.L., H.Z. and Y.W.; Visualization, Y.L. and Y.W.; Supervision, H.Z.; Project administration, H.Z.; Funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by ZQSG-1 Project Department of Weisuxian High-Speed Railway to Qingdao Connecting Line, China Railway 10th Bureau Group Co., Ltd. (Grant Number 43010100).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare that this study received funding from the ZQSG-1 Project Department of Weisuxian High-Speed Railway to Qingdao Connecting Line, China Railway 10th Bureau Group Co., Ltd. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

References

1. Bergeson, W.; Ernst, S.L. *Tunnel Operations Maintenance, Inspection, and Evaluation (TOMIE) Manual*; Federal Highway Administration: Washington, DC, USA, 2015.
2. Yang, T.; Tangqing, J.; Zhang, G.; Long, S.; Guo, F.; Xu, F.; Chen, Z.; Wu, Y. A CNN-based automatic detection method for tunnel lining internal defect using ground penetrating radar. In Proceedings of the 2023 International Annual Conference on Complex Systems and Intelligent Science (CSIS-IAC), Shenzhen, China, 20–22 October 2023; pp. 779–783. [\[CrossRef\]](#)
3. Ruggiero, A.; D’Anna, S. *Modeling and Imaging of Ground Penetrating Radar Data*; The University of Texas at Austin: Austin, TX, USA, 2004.
4. Li, J.; Liu, C.; Zeng, Z.; Chen, L. GPR signal denoising and target extraction with the CEEMD method. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1615–1619. [\[CrossRef\]](#)
5. Wang, G.C.; Tse, P.W.; Yuan, A.D. Automatic internal crack detection from a sequence of infrared images with a triple-threshold Canny edge detector. *Meas. Sci. Technol.* **2018**, *29*, 025403. [\[CrossRef\]](#)
6. Li, C.; Xu, P.; Niu, L.; Chen, Y.; Sheng, L.; Liu, M. Tunnel crack detection using coarse-to-fine region localization and edge detection. *WIREs Data Min. Knowl. Discov.* **2019**, *9*, e1308. [\[CrossRef\]](#)
7. Man, K.; Liu, R.; Liu, X.; Song, Z.; Liu, Z.; Cao, Z.; Wu, L. Water Leakage and Crack Identification in Tunnels Based on Transfer-Learning and Convolutional Neural Networks. *Water* **2022**, *14*, 1462. [\[CrossRef\]](#)
8. Liu, Z.; Suandi, S.A.; Ohashi, T.; Ejima, T. Tunnel crack detection and classification system based on image processing. In *Machine Vision Applications in Industrial Inspection X*; SPIE: Bellingham, WA, USA, 2002; Volume 4664, pp. 145–153. [\[CrossRef\]](#)
9. Liu, Y.; Zhang, L.; Wang, X. Risk prediction and diagnosis of water seepage in operational shield tunnels based on random forest. *J. Civ. Eng. Manag.* **2021**, *27*, 539–552. [\[CrossRef\]](#)
10. Yu, C.; Geng, D.X.; Huang, Z.J.; Zhu, Z.H.; Shi, Y.F. Refinement identification and evaluation of tunnel lining cracks. *J. Highw. Transp. Res. Dev.* **2020**, *14*, 75–83. [\[CrossRef\]](#)
11. Li, Z.Y.; Xu, C.; Yang, Y.Y.; Lai, J.Y.; Xian, Q.; Mai, X.M.; Leiri, Y.; Yoshie, O. CWA-YOLO: Rethinking the YOLOv8 architecture at the level of feature information and applying it to substation equipment detection. In Proceedings of the 2024 7th International Conference on Information Communication and Signal Processing (ICICSP), Zhoushan, China, 21–23 September 2024; pp. 1050–1057. [\[CrossRef\]](#)
12. Wang, S.; Dong, Q.; Chen, X.; Chu, Z.; Li, R.; Hu, J.; Gu, X. Measurement of asphalt pavement crack length using YOLO V5-BiFPN. *J. Infrastruct. Syst.* **2024**, *30*, 04024005. [\[CrossRef\]](#)
13. Zhou, Z.; Zhang, J.; Gong, C. Automatic detection method of tunnel lining multi-defects via an enhanced you only look once network. *Comput.-Aided Civ. Infrastruct. Eng.* **2022**, *37*, 762–780. [\[CrossRef\]](#)
14. Chen, L.; Yao, H.; Fu, J.; Ng, C.T. The classification and localization of crack using lightweight convolutional neural network with CBAM. *Eng. Struct.* **2023**, *275*, 115291. [\[CrossRef\]](#)
15. Liu, H.; Lin, C.; Cui, J.; Fan, L.; Xie, X.; Spencer, B.F. Detection and localization of rebar in concrete by deep learning using ground penetrating radar. *Autom. Constr.* **2020**, *118*, 103279. [\[CrossRef\]](#)
16. Kaur, P.; Dana, K.J.; Romero, F.A.; Gucunski, N. Automated GPR rebar analysis for robotic bridge deck evaluation. *IEEE Trans. Cybern.* **2015**, *46*, 2265–2276. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Han, J.H.; Kim, I.S.; Lee, C.H.; Moon, Y.S. Crack detection method for tunnel lining surfaces using ternary classifier. *SII Trans. Internet Inf. Syst.* **2020**, *14*, 3797–3822. [\[CrossRef\]](#)
18. Yan, X.; Zhou, G.Y.; Zhao, X.F. Method for rapid detection and treatment of cracks in tunnel lining based on deep learning. In *Health Monitoring of Structural and Biological Systems IX*; SPIE: Bellingham, WA, USA, 2020; Volume 11381, pp. 331–339.
19. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *arXiv* **2015**, arXiv:1510.00149. [\[CrossRef\]](#)

20. Xue, Y.; Wang, L.; Liu, Y.; Ranjith, P.G.; Cao, Z.; Shi, X.; Gao, F.; Kong, H. Brittleness evaluation of gas-bearing coal based on statistical damage constitution model and energy evolution mechanism. *J. Cent. South Univ.* **2025**, *32*, 566–581. [[CrossRef](#)]
21. Xue, Y.; Wang, L.; Cao, Z.; Xie, H.; Ranjith, P.G.; Cai, C. Thermo-mechanical degradation and crack evolution of granite subjected to liquid nitrogen cooling cycles for thermal reservoir stimulation. *J. Rock Mech. Geotech. Eng.* **2025**, *18*, 3447–3470. [[CrossRef](#)]
22. Xue, Y.; Wang, L.; Cai, Y.; Liu, J.; Ranjith, P.G.; Gao, F.; Xie, H.; Liu, Y. Effects of heating–cooling on the mechanical properties and damage evolution of granite for underground cryogenic storage. *Acta Geotech.* **2026**. [[CrossRef](#)]
23. Khanam, R.; Hussain, M. YOLOv11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725. [[CrossRef](#)]
24. ULTRALytics. YOLOv11 Documentation. 2024. Available online: <https://docs.ultralytics.com/models/yolov11/> (accessed on 27 August 2025).
25. 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*; IEEE: Piscataway, NJ, USA, 2019; pp. 1314–1324.
26. 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. [[CrossRef](#)]
27. 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*; IEEE: Piscataway, NJ, USA, 2018; pp. 4510–4520.

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.