

Article

Optimization of Visual Detection Algorithms for Elevator Landing Door Safety-Keeper Bolts

Chuanlong Zhang ¹, Zixiao Li ², Jinjin Li ³, Lin Zou ⁴ and Enyuan Dong ^{1,*}

¹ School of Electrical Engineering, Dalian University of Technology, Dalian 116000, China; 13889469148@163.com

² School of Mechanical Engineering, Dalian Jiaotong University, Dalian 116000, China; zamasu33@163.com

³ School of Electrical Engineering, Dalian Jiaotong University, Dalian 116000, China; a379415016@163.com

⁴ Dalian Boiler and Pressure Vessel Inspection and Testing Research Institute Co., Ltd., Dalian 116000, China; 13591836882@163.com

* Correspondence: dey@dlut.edu.cn

Abstract

As the safety requirements of elevator systems continue to rise, the detection of loose bolts and the high-precision segmentation of anti-loosening lines have become critical challenges in elevator landing door inspection. Traditional manual inspection and conventional visual detection often fail to meet the requirements of high precision and robustness under real-world conditions such as oil contamination and low illumination. This paper proposes two improved algorithms for detecting loose bolts and segmenting anti-loosening lines in elevator landing doors. For small-bolt detection, we introduce the DS-EMA model, an enhanced YOLOv8 variant that integrates depthwise-separable convolutions and an Efficient Multi-scale Attention (EMA) module. The DS-EMA model achieves a 2.8 percentage point improvement in mAP over the YOLOv8n baseline on our self-collected dataset, while reducing parameters from 3.0 M to 2.8 M and maintaining real-time throughput at 126 FPS. For anti-loosening-line segmentation, we develop an improved DeepLabv3+ by adopting a MobileViT backbone, incorporating a Global Attention Mechanism (GAM) and optimizing the ASPP dilation rate. The revised model increases the mean IoU to 85.8% (a gain of 5.4 percentage points) while reducing parameters from 57.6 M to 38.5 M. Comparative experiments against mainstream lightweight models, including YOLOv5n, YOLOv6n, YOLOv7-tiny, and DeepLabv3, demonstrate that the proposed methods achieve superior accuracy while balancing efficiency and model complexity. Moreover, compared with recent lightweight variants such as YOLOv9-tiny and YOLOv11n, DS-EMA achieves comparable mAP while delivering notably higher recall, which is crucial for safety inspection. Overall, the enhanced YOLOv8 and DeepLabv3+ provide robust and efficient solutions for elevator landing door safety inspection, delivering clear practical application value.

Keywords: elevator safety inspection; YOLOv8; small-object detection; DeepLabv3+; anti-loosening-line segmentation



Academic Editor: Jiangjian Xiao

Received: 14 August 2025

Revised: 29 August 2025

Accepted: 29 August 2025

Published: 1 September 2025

Citation: Zhang, C.; Li, Z.; Li, J.; Zou, L.; Dong, E. Optimization of Visual Detection Algorithms for Elevator Landing Door Safety-Keeper Bolts. *Machines* **2025**, *13*, 790. <https://doi.org/10.3390/machines13090790>

Copyright: © 2025 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) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Elevators, as an indispensable mode of transportation in modern cities, are widely used in residential, commercial, and public facilities, greatly facilitating people's mobility. However, with the increasing number of elevators, elevator safety has become an increasingly prominent issue of public concern. The landing door, being a part directly in contact with passengers, plays a crucial role in the safe operation of the entire elevator

system [1]. One of the main causes of landing door failures is missing or loose bolts on the safety keeper, with traditional manual inspection methods being inefficient and prone to omissions or false detections [2].

With the rapid development of sensor, signal processing, and machine vision technologies, researchers, both domestically and internationally, have proposed various methods for detecting bolt loosening. For example, Pal et al. [3] utilized sensors to obtain frequency data, employing position-determined fixed coefficients as variables and continuously updating the model to determine the actual position and fixed coefficient of the bolts at varying degrees of looseness. Sun et al. [4] embedded piezoelectric sensors at predetermined positions on the bolt heads and developed a smart piezoelectric bolt to measure the ultrasonic flight time of the bolt shaft. Such sensor-based detection methods enable quantitative detection with excellent results; however, they are costly and require complex installation and calibration. In comparison, vision-based detection methods have become a hot research topic due to their lower cost, stronger environmental adaptability, and higher detection efficiency [5,6]. Numerous studies have shown that vision-based detection methods have significant advantages in bolt loosening detection. For instance, Zhang et al. [7] directly output the bolt tightening status using a region-based convolutional neural network, achieving precise qualitative detection; Kong and Li [8] applied image registration technology, incorporating the rotational angle differences caused by bolt loosening into error evaluation, enhancing detection accuracy. Huynh et al. [9] combined the RCNN deep learning algorithm with Hough line detection to achieve automatic bolt detection. Lyu [10], on the other hand, utilized YOLOv5 and error ellipse theory to identify the rotational angle of bolts, achieving good detection results. In addition, Wang et al. [11] proposed a detection algorithm for loose bolts on low-quality images of roofs of rail vehicles, using a line-array camera to capture bolt images and applying brightly colored anti-loosening lines on the bolts. By dividing the roof into multiple sub-regions and using the HALCON object detection framework for bolt positioning, they calculated the angle of the anti-loosening line on the bolts to determine if they were loose. This method successfully detects roof bolt loosening by fitting the anti-loosening line as a straight line and comparing the angles, demonstrating high accuracy.

However, detecting bolts on the elevator landing door safety keeper still faces numerous challenges. The bolts on the safety keeper have multiple characteristics, such as being covered by heavy oil stains, insufficient lighting, and being small in size [12]. These factors make it difficult for traditional visual detection methods to achieve ideal results in practical applications. As a result, traditional object detection techniques often fail to meet the requirements for high precision and high reliability.

Recent years have witnessed remarkable progress in deep learning-based object detection and semantic segmentation. For small-object and lightweight detection, numerous studies have integrated attention modules such as SE [13], CBAM [14], and EMA (Efficient Multi-scale Attention) into the YOLO framework, effectively enhancing feature representation and improving detection accuracy for tiny targets under resource constraints. In addition to conventional detectors, oriented object detection methods have been proposed to handle rotation-sensitive tasks. For instance, Yang et al. introduced R3Det [15], which refines feature maps to better capture rotated objects, while Ming et al. proposed Gradient Calibration Loss (GCL) [16], which optimizes rotated IoU loss via gradient analysis and correction, leading to faster convergence and more stable training in oriented object detection. These approaches are closely related to bolt-loosening detection, where angle deviations must be accurately measured.

In the field of semantic segmentation, lightweight architectures such as BiSeNet V2 [17] and transformer-based models like SegFormer [18] have demonstrated high efficiency and

robust performance in parsing fine-grained structures. Their ability to balance accuracy and speed makes them particularly suitable for industrial inspection scenarios. Compared with these works, our study further enhances YOLOv8 and DeepLabv3+ by incorporating lightweight and attention mechanisms, specifically targeting the challenges of oil stains, insufficient lighting, and small bolt sizes in elevator landing door inspection.

To effectively address these issues, deep learning models such as YOLOv8 [19] and DeepLabv3+ [20] have been widely applied in object detection and semantic segmentation in recent years, with significant improvements made in lightweight design, accuracy enhancement, and small-target detection. YOLOv8 has made significant progress in lightweight design and small-target detection. Zhu et al. [21] have enhanced the model's sensitivity and detection capability for small targets by introducing an attention mechanism and multi-scale feature fusion. In particular, by optimizing the network structure of YOLOv8, the researchers further reduced the computational complexity and improved detection speed, enabling efficient operation even on resource-constrained devices. In the field of semantic segmentation, DeepLabv3+ has also undergone significant improvements. Gulzar [22] reduced the model's parameter count and improved its inference speed by adopting a lightweight backbone network, such as MobileNetv2. In the Atrous Spatial Pyramid Pooling (ASPP) module, researchers introduced depthwise separable convolutions [23], which not only reduced computational overhead but also enhanced the model's segmentation accuracy. At the same time, DeepLabv3+ incorporated an attention mechanism and enhanced feature fusion strategies, further improving the model's performance in detail and small-target segmentation.

In summary, this paper proposes an intelligent detection algorithm for detecting loose bolts on the elevator landing door safety keeper. The algorithm combines the improved YOLOv8 object detection network with the DeepLabv3+ semantic segmentation network. The enhanced YOLOv8 algorithm allows for more accurate detection of all bolts and determination of whether any are missing, while analyzing the anti-loosening line features on the bolts inside the landing door track to detect whether the bolts are loose. The system captures bolt images under various operating conditions and, combined with multiple image processing techniques, performs looseness detection. While YOLOv8 and DeepLabv3+ have achieved remarkable progress in lightweight design and small-object detection, their performance still degrades under real-world challenges such as severe illumination variation, oil contamination, and noisy backgrounds. In particular, thin anti-loosening lines are difficult to segment robustly when occlusion and stains exist. To address these limitations, our method integrates an attention-enhanced YOLOv8 with an optimized DeepLabv3+, specifically designed for elevator landing door inspection. Experimental evaluation of images captured under various environmental conditions showed that the detection accuracy of the algorithm was significantly improved, maintaining high accuracy even under challenging conditions such as oil stains and insufficient lighting. By adopting the lightweight improved YOLOv8 and deep learning-based semantic segmentation algorithm, this paper not only addresses the inefficiency and false omissions of traditional manual detection but also significantly improves the detection accuracy of elevator landing door bolts, offering a new solution for intelligent elevator maintenance. The successful application of this algorithm not only optimizes the elevator maintenance workflow but also provides more reliable technical support for elevator safety.

2. An Enhanced YOLOv8 for Small-Bolt Detection

2.1. Overview of YOLOv8

YOLOv8 is an important version in the YOLO series, offering significant performance improvements [13]. It continues to use the scaling coefficient design from YOLOv5, offering

models in five different sizes: N, S, M, L, and X, which can be flexibly adjusted based on hardware and computational resource requirements. In terms of structure, YOLOv8's backbone and neck modules reference the ELAN design from YOLOv7 [24], adopting the new C2f structure, optimizing gradient flow, and enhancing feature extraction capabilities. The head module of YOLOv8 uses a Decoupled-Head structure, separating classification and detection tasks, and shifts from an Anchor-Based approach to an Anchor-Free approach, improving detection accuracy. The loss function of YOLOv8 has also been optimized, employing a Task-Aligned Assigner for positive and negative sample matching, addressing the class imbalance issue. Additionally, Distribution Focal Loss (DFL) has been introduced, further improving the detection of challenging targets.

Despite its excellent performance in many object detection tasks, YOLOv8 still faces certain challenges in detecting bolts on elevator landing door safety keepers. The landing door bolts are relatively small and often affected by factors such as oil stains and insufficient lighting, which may limit YOLOv8's detection ability for small targets and complex backgrounds. Therefore, to meet the requirements of elevator bolt loosening detection, this paper makes targeted improvements to YOLOv8, enhancing its detection accuracy for small targets in complex environments and ensuring accurate bolt detection.

2.2. Establishment of the DSEW-YOLOv8 Network Structure

DSEW-YOLOv8 is derived from the YOLOv8n network, improved by using depthwise separable convolutions to optimize the EMA mechanism, forming the DS-EMA module, which is applied to the backbone network to enhance small-target detection capabilities. Additionally, the WIoU (Weighted IoU) loss function replaces the original CIoU loss function and is applied to the backbone, neck, and head networks, improving the model's detection accuracy in complex environments and ensuring the precision and robustness of elevator safety-keeper bolt loosening detection. The improved DSEW-YOLO network structure is shown in Figure 1.

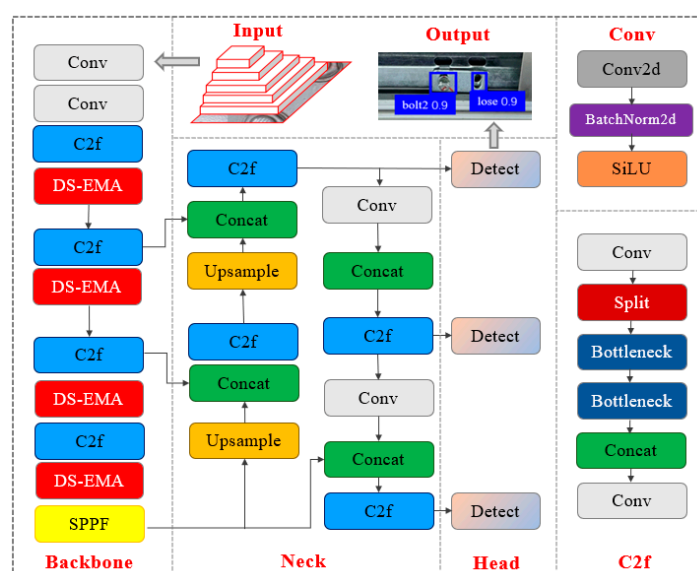


Figure 1. Overall architecture of the proposed DSEW-YOLOv8 detector, showing the integration of DS-EMA attention for small-bolt detection.

2.2.1. DS-EMA Module

Because most safety-keeper bolts in elevator landing doors are small targets, predictions with the original model often suffer from false positives and false negatives. Therefore, we introduce the Efficient Multi-scale Attention (EMA) [25] mechanism into the YOLOv8 feature-

extraction stage. By coupling efficient cross-spatial feature learning with multi-scale information fusion, EMA markedly enhances the model's ability to capture small-object features.

EMA splits the input feature map into channel-wise groups, which are processed by two parallel 1×1 convolutions and one 3×3 convolution. The 1×1 paths apply horizontal and vertical global-average pooling to capture long-range spatial dependencies and then produce attention weights via 1×1 convolution + sigmoid, while the 3×3 path gathers fine local details. Outputs from all branches are fused with cross-spatial learning: global-average pooling encodes each branch, a dot-product aggregates multi-scale information, and a final sigmoid yields the spatial-attention map that highlights salient regions. Instead of reducing channels, EMA reshapes part of the channel dimension into the batch dimension, keeping rich information yet limiting parameters. Nevertheless, the Conv2d operations in EMA still involve a large number of parameters, adding computational burden to the model.

To mitigate this complexity, we introduce depthwise separable convolutions into EMA, constructing a lightweight DS-EMA attention mechanism that reduces both computation and parameter count. Depthwise separable convolution, first proposed by Chollet in MobileNet, effectively reduces the parameters and computation of convolutional neural networks. Unlike standard convolution, depthwise separable convolution decomposes the operation into two independent stages: depthwise convolution and pointwise convolution. In the depthwise stage, each input channel is convolved separately in space; consequently, the number of depthwise kernels equals the number of input channels.

Let the input feature map be of size $D_F \times D_F \times M$, the depthwise kernel be of size $D_k \times D_k \times 1$, and the output feature map be of size $D_F \times D_F \times M$. The depthwise convolution can then be expressed as Equation (1):

$$\hat{G}_k(x, y) = \sum_{i=1}^{D_k} \sum_{j=1}^{D_k} F_k(x + i - 1, y + j - 1) \cdot K_k(i, j) \quad (1)$$

F_k denotes the k -th channel of the input feature map; K_k is the k -th depthwise convolution kernel; \hat{G}_k is the k -th channel of the output feature map after depthwise convolution.

Pointwise convolution employs a 1×1 kernel to linearly combine the depthwise features along the channel dimension, producing the final output feature map. Let the input feature map for the pointwise convolution be of size $D_F \times D_F \times M$ and the number of output channels be N ; the pointwise convolution can be expressed as Equation (2):

$$G_n(x, y) = \sum_{k=1}^M \hat{G}_k(x, y) \cdot P_{n,k} \quad (2)$$

$\hat{G}_k(x, y)$ is the k -th channel of the feature map output by the depthwise convolution; $P_{n,k}$ is the weight of the n -th pointwise kernel corresponding to input channel k ; $G_n(x, y)$ is the n -th channel of the output feature map after pointwise convolution.

The structure of the depthwise separable convolution is illustrated in Figure 2.

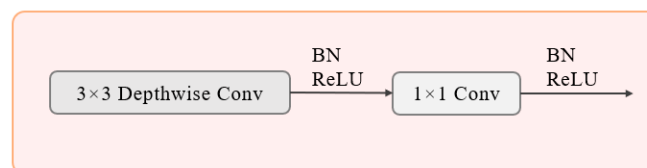


Figure 2. Depthwise separable convolution structure. (This schematic was redrawn and simplified by the authors for clarity).

Compared with a standard convolution, if the input has M channels, the output has N channels, and the kernel size is $D_K \times D_K$, the parameter count of the standard convolution is $D_K \times D_K \times M \times N$. For a depthwise separable convolution, the parameter count equals the sum of the depthwise and pointwise parts, which is $D_K \times D_K \times M + M \times N$.

It is evident that depthwise separable convolution dramatically reduces the number of parameters relative to a standard convolution, thereby improving computational efficiency. Consequently, in the EMA mechanism, we replace the 3×3 convolutions with depthwise separable convolutions, which markedly lower computation and parameter counts while retaining EMA's original multi-scale, cross-spatial attention aggregation advantages. The architecture of the improved DS-EMA module is shown in Figure 3.

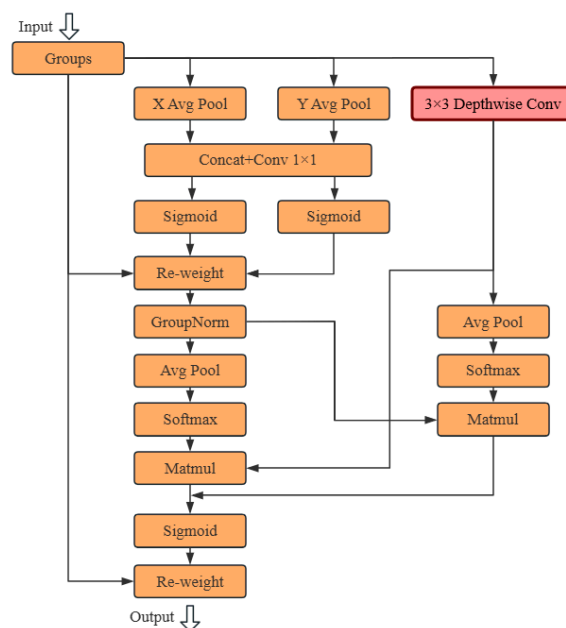


Figure 3. Architecture of the improved DS-EMA attention module.

2.2.2. WIoU Loss Function

YOLOv8 computes bounding-box regression loss with the Complete IoU (CIoU) loss, which jointly penalizes overlap, aspect ratio, and center-point distance. Although this composite term accelerates convergence—especially for objects with complex shapes or coordinates—CIoU introduces many parameters, involves cumbersome calculations, and increases computational load. Its aspect-ratio term can also be excessively severe: for small objects, CIoU focuses strongly on shape details and aspect-ratio variation, so performance degrades when IoU is low, yet shape discrepancy is large. In addition, small errors may receive large penalties, causing over-fitting and poor generalization.

Accordingly, we replace CIoU with the Weighted IoU (WIoU) loss. WIoU adaptively re-weights factors such as IoU, shape, and scale differences, mitigating the performance drop that CIoU suffers when box shapes vary greatly or overlaps are small. In our scenario—long camera-to-bolt distance and tiny bounding boxes—WIoU localizes boundaries more precisely, boosting detection accuracy under challenging conditions and preserving the precision and robustness required for safety-keeper bolt-loosening detection. A schematic of WIoU is shown in Figure 4, and its formulation is given below. In the following expressions, L_{WIoU} denotes the WIoU bounding-box regression loss, L_{IoU} is the standard IoU loss, and R_{WIoU} is the distance-attention term. As defined in Equations (3)–(6), (x, y) and (x_{gt}, y_{gt}) are the center coordinates of the predicted and ground-truth boxes; W_g and H_g are the width and height of their minimum enclosing rectangle; they are detached from the computation graph to prevent R_{WIoU} from producing convergence-hindering gradients, thereby removing this obstacle to

training; δ and α are hyper-parameters; r is a non-monotonic focusing factor; β represents the outlier degree. L_{IoU}^* is the loss on the predicted IoU, and $\overline{L_{IoU}}$ is the dynamic IoU loss.

$$L_{WIoU} = R_{WIoU} L_{IoU} \quad (3)$$

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)}\right) \quad (4)$$

$$r = \frac{\beta}{\delta \cdot \alpha^{\beta-\delta}} \quad (5)$$

$$\beta = \frac{L_{IoU}^*}{L_{IoU}} \in [0, +\infty) \quad (6)$$

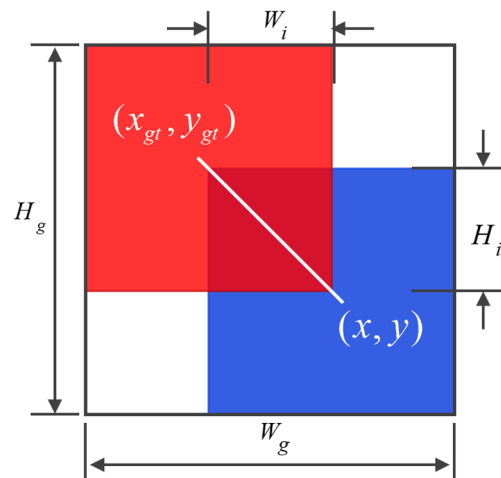


Figure 4. Schematic of the WIoU loss.

By incorporating the DS-EMA attention mechanism and the WIoU loss, the YOLOv8 network is substantially improved, delivering superior detection performance on small targets in complex backgrounds and providing a reliable foundation for subsequent detection of bolt anti-loosening lines.

3. An Upgraded DeepLabv3+ for High-Precision Segmentation of Anti-Loosening Lines

3.1. Improvement of the DeepLabv3+ Network

DeepLabv3+ is an advanced semantic-segmentation model that has been widely adopted and delivers outstanding accuracy. Compared with classical semantic segmentation models such as U-Net or PSPNet, DeepLabv3+ achieves a better trade-off between accuracy and efficiency, especially in small-object scenarios. Moreover, it is more practical for integration with YOLO-based detection frameworks than heavier instance segmentation approaches such as Mask R-CNN, which usually incur higher computational cost. In our task—segmenting anti-loosening lines on elevator-door safety-keeper bolts—the targets are extremely small with simple features on a uniform background. The original Xception backbone is overly complex, and its depthwise-separable convolutions transmit limited cross-channel information, leading to sub-optimal performance on tiny targets. Moreover, the Atrous Spatial Pyramid Pooling (ASPP) module uses large dilation rates (6, 12, 18), which markedly reduce sensitivity to small objects and fail to meet real-time, high-precision demands.

To address these issues, we make three structural modifications: (i) replace the Xception backbone with the lightweight MobileViT network, which efficiently captures global semantics; (ii) insert an efficient Global Attention Mechanism (GAM) into both shallow and output layers

of MobileViT to strengthen small-object representations; and (iii) reduce dilation rates in the ASPP module to 1, 2, 3, better adapting the network to tiny-target segmentation.

3.1.1. MobileViT Backbone Design

MobileViT is a lightweight yet powerful feature extractor that fuses MobileNet convolutions with a vision Transformer, reducing parameters while still capturing global context. As illustrated in Figure 5, the architecture comprises three parts: an initial convolutional stem, a core MobileViT-Block stack, and a final output head. The stem reduces spatial resolution and increases channel depth by standard convolution and down-sampling, enabling subsequent blocks to extract features more efficiently.

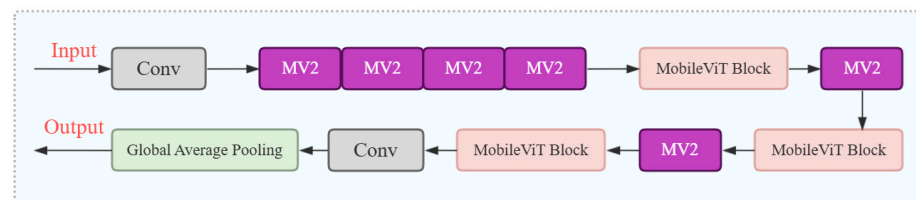


Figure 5. Overall MobileViT backbone architecture.

The MobileViT Block—shown in Figure 6—first applies a 3×3 convolution to extract local features, followed by a 1×1 convolution for channel expansion. A Transformer self-attention layer then captures global semantics and fuses spatial information. Finally, an inverse reshaping restores spatial size, a 1×1 convolution adjusts channels, and a residual connection is added. This interplay of local and global cues enhances fine-feature representation while keeping the parameter budget low, making MobileViT ideal for our tiny-target segmentation task.

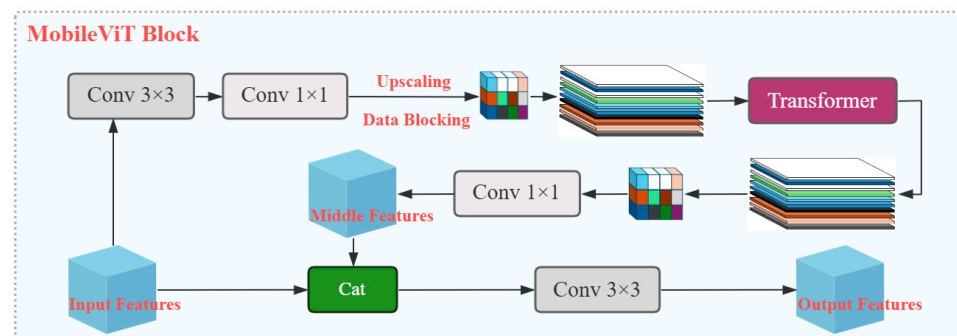


Figure 6. Internal structure of a MobileViT block.

In addition, MobileViT employs an MV2 inverted-residual block: a 1×1 expansion, a depthwise-separable convolution for high-dimensional processing, and a 1×1 projection that fuses features and forms a residual link to the input (Figure 7). This design increases non-linear capacity while lowering computation, satisfying both lightweight and performance requirements.

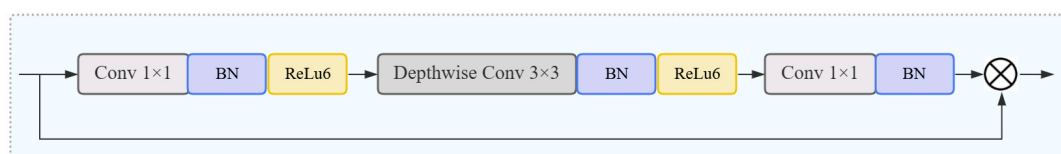


Figure 7. Structure of the MV2 inverted residual block, which is employed as a component within the MobileViT backbone.

3.1.2. Introduction and Enhancement of the Global Attention Mechanism (GAM)

To further boost sensitivity to small targets, we inject the Global Attention Mechanism (GAM) [26] into both shallow and output layers of MobileViT. The GAM comprises a channel-attention module and a spatial-attention module.

Channel attention. The input tensor is reshaped to 3-D form ($C \times H \times W$) to preserve cross-dimensional cues. A multilayer perceptron (MLP) models dependencies across channel, height, and width, produces a non-linear attention map, and multiplies it element-wise with the input to yield channel-enhanced features.

Spatial attention. Taking the channel-enhanced tensor as input, the original GAM uses two 7×7 convolutions for spatial fusion. Given mobile-platform constraints, we replace them with two 3×3 dilated convolutions. This substitution preserves the receptive field while cutting parameters and FLOPs, thus improving real-time efficiency.

The modified GAM architecture is shown in Figure 8. Its inclusion markedly increases MobileViT's focus on anti-loosening-line features and, despite fewer parameters, further raises segmentation accuracy.

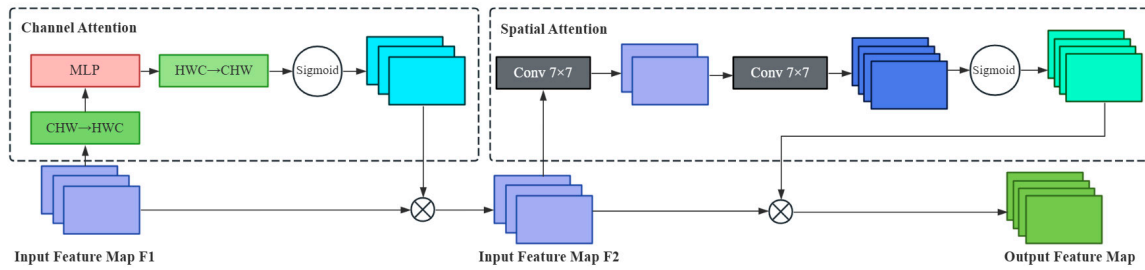


Figure 8. Modified GAM attention architecture integrated into MobileViT, highlighting the global attention mechanism's role in enhancing segmentation accuracy.

3.1.3. Parameter Optimization of the ASPP Module

A hallmark of DeepLabv3+ is its Atrous Spatial Pyramid Pooling (ASPP) module, which employs multiple atrous convolutions with different dilation rates to capture multi-scale context and greatly enhance segmentation. The conventional ASPP uses large dilation rates (typically 6, 12, 18); Given an input feature x , an output y , kernel size $k \times k$, and dilation r , the atrous convolution at position i is:

$$y[i] = \sum_{m+n \cdot r=i} x[m] \cdot w[n] \quad (7)$$

where $w[n]$ is the kernel weight, m indexes the input, and n indexes the kernel position. Increasing r enlarges the receptive field but also widens pixel spacing, giving a sparse sampling pattern suited to large objects or semantically rich regions.

Our anti-loosening lines are tiny, with plain features and backgrounds. Large dilation rates (6, 12, 18) make the kernel cover excessive area, drowning small-object cues in background and diminishing sensitivity. To remedy this, we lower the dilation rates to 1, 2, 3, yielding the following revised ASPP formulation:

$$y[i] = \sum_{m+n \cdot r=i} x[m] \cdot w[n], r \in \{1, 2, 3\} \quad (8)$$

For 3×3 kernels: $r = 1$ yields a 3×3 receptive field (spacing 1); $r = 2$ expands it to 5×5 (spacing 2); $r = 3$ extends it to 7×7 (spacing 3).

These smaller rates shrink the receptive field and sample more densely, allowing the network to focus on local structures and avoid losing fine details.

The revised DeepLabv3+ architecture (Figure 9) cuts parameters and computation while markedly improving small-object feature extraction, making it well-suited for segmenting safety-keeper bolts.

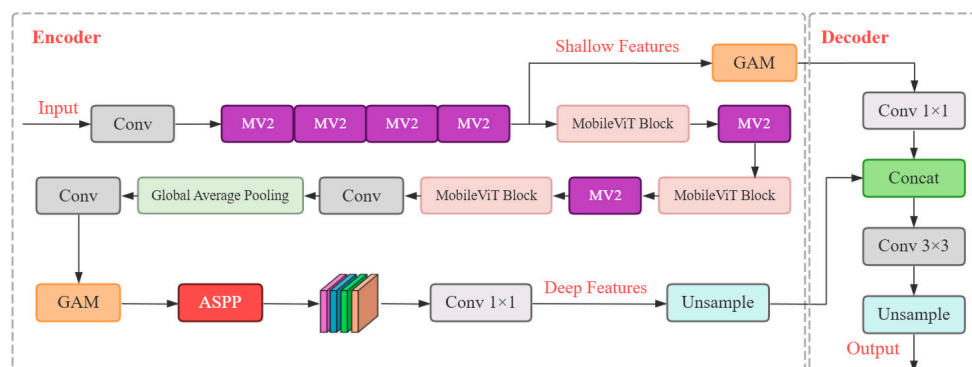


Figure 9. Revised DeepLabv3+ architecture incorporating the MobileViT backbone, GAM attention, and optimized ASPP for small-object segmentation.

3.2. Looseness Determination Method

The study involves two types of fasteners, denoted bolt 1 and bolt 2. Bolt 1 is a single screw that mates with a pre-tapped hole in the landing door structure, whereas bolt 2 is more complex and consists of a bolt, a square nut, a spring washer, and a plain washer. Because the nut and washers serve only auxiliary anti-loosening functions, looseness is judged solely by the relative motion between the bolt body and the square nut. In practice, two anti-loosening lines are painted—one on the bolt head and one on the fastened surface of the landing door—to indicate the tightening state. Under normal conditions, the two lines remain parallel; when the joint loosens, the lines deviate by a measurable angle, as shown in Figure 10.

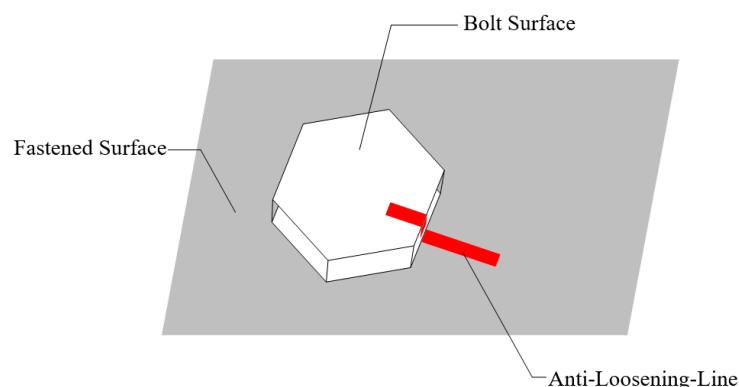


Figure 10. Schematic of the anti-loosening lines on the bolt head.

However, hand-painted lines are irregular and lie on rough surfaces, so direct parallelism assessment is unreliable. The segmentation results of the anti-loosening lines for bolt 1 and bolt 2 are shown in Figure 11a,b, respectively. To address this issue, the line regions are first extracted by semantic segmentation. A convex-hull algorithm then yields the minimum convex polygon enclosing all pixels, after which a rotating-caliper method fits the minimum bounding rectangle, providing a standardized representation of each anti-loosening line.

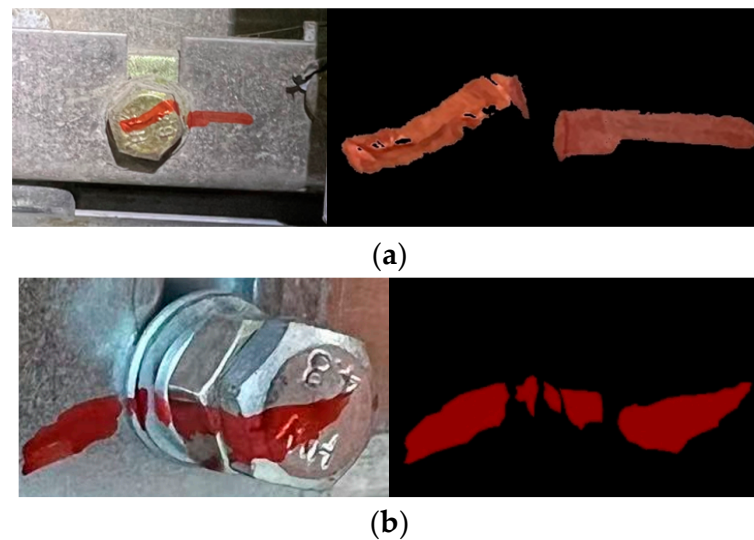


Figure 11. Examples of anti-loosening-line segmentation: (a) bolt 1; (b) bolt 2.

As illustrated in Figure 12, once the minimum bounding rectangle has been obtained, the image-pixel coordinate system—with its origin at the upper-left corner and positive x and y axes pointing rightward and downward, respectively—is adopted as the reference. Using the rectangle's four vertices (w_1, h_1) , (w_2, h_2) , (w_3, h_3) , (w_4, h_4) , the lengths of the two adjacent sides, L_1 and L_2 , are calculated as follows:

$$L_1 = \sqrt{\Delta w_1^2 + \Delta h_1^2}, \Delta w_1 = w_{w_{\max}} - w_{h_{\min}}, \Delta h_1 = h_{w_{\max}} - h_{h_{\min}} \quad (9)$$

$$L_2 = \sqrt{\Delta w_2^2 + \Delta h_2^2}, \Delta w_2 = w_{w_{\max}} - w_{h_{\min}}, \Delta h_2 = h_{w_{\max}} - h_{h_{\min}} \quad (10)$$

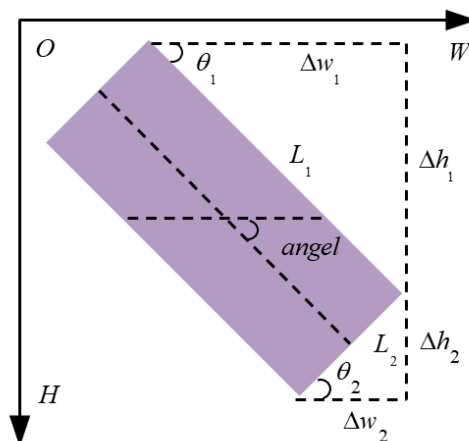


Figure 12. Minimum-bounding-rectangle geometry and angel definitions for looseness evaluation.

Based on the above principle, looseness for the two bolt types is determined following the procedure shown in Figure 13. The thresholds A_{thr} and θ_{thr} are set to 0.03/0.06 and 15° for bolt 1 and bolt 2, respectively; the rectangle-selection rule is configured at run time.

To avoid redundancy, the detailed visual demonstration and experimental validation of each step in the workflow are presented later in Section 5.3.3.

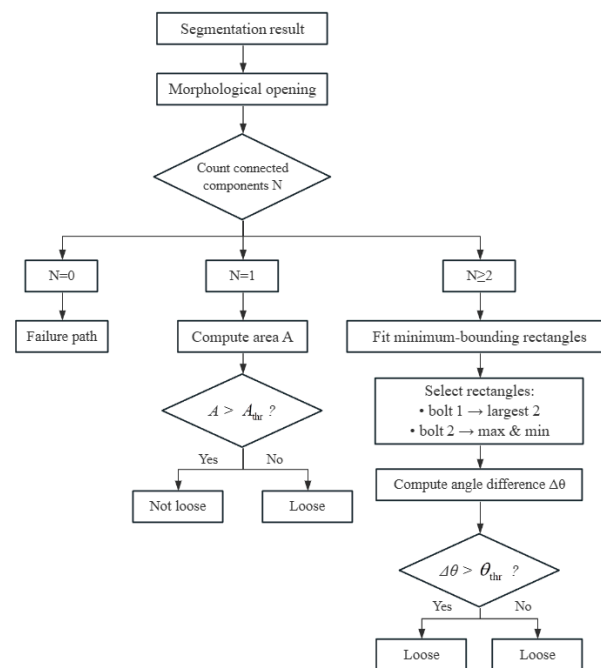


Figure 13. Looseness determination workflow, including preprocessing, component counting, area thresholding, and angular deviation analysis.

4. Experimental Platform and Dataset Construction

4.1. Experimental Platform Setup

Acquiring images of in-service landing door safety keepers requires taking the elevator out of service and accessing the car roof under the supervision of safety personnel; moreover, loosened or dropped bolts are rare in the field. To obtain sufficient data—including diverse conditions (normal, loosened, and missing bolts)—we therefore built a simulated elevator landing door inspection platform that reproduces the safety keeper location and the anticipated installation/operating configuration of the system, as shown in Figure 14a. A representative image of the real landing door and its safety keeper captured inside the shaft is provided in Figure 14b. During both the simulated and in situ inspections, the camera and lens were mounted at a fixed distance of approximately 1 m from the safety keeper. This distance was selected to ensure a clear field of view and stable imaging conditions. The safety-keeper bolts used in both the simulated and in situ platforms are standard elevator fasteners with a head diameter of approximately 10–12 mm. Due to their small size, sufficient resolution and an appropriate working distance are required to ensure clear image acquisition. The bolts to be detected are standard elevator safety-keeper bolts, with their positions and shapes clearly illustrated in Figure 15.



Figure 14. Data acquisition scenes: (a) Simulated landing door inspection platform; (b) In-shaft landing door image with safety keeper.

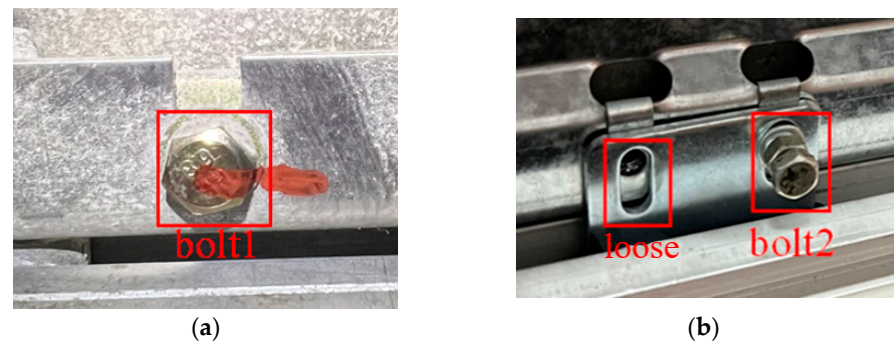






Figure 15. Examples of bounding-box annotations: (a) bolt 1; (b) missing bolt and bolt 2.

To satisfy application requirements, the attributes and parameters of the experimental components are listed in Table 1.

Table 1. List of experimental platform components, including camera, lens, illumination, and industrial PC, with their key specifications.

Component	Model	Key Specifications	Live Picture
Camera	MV-CS200-10GC (HIKROBOT, Hangzhou, China)	20 MP color CMOS area camera; Resolution 5472×3648 ; Exposure time $46 \mu\text{s}$ – 2.5 s	
Lens	MV-CS200-10GC (HIKROBOT, Hangzhou, China)	Focal length 12 mm; F2.4 – F16; Image circle $\Phi 19.3 \text{ mm}$ (1.2"); Resolution 25 MP	
Illumination	MV-CS200-10GC (HIKROBOT, Hangzhou, China)	Brightness 2000 lx; CT 6000–7500 K; Power 3.4 W	
Industrial PC	MV-CS200-10GC (HIKROBOT, Hangzhou, China)	Intel Core i3-6100; 8 GB RAM/512 GB SSD	

4.2. Dataset Construction and Annotation

By inspecting elevator shafts in dozens of residential buildings, shopping malls, and schools across Dalian, we collected 440 in situ images of the bolt 1 class (Figure 15a). An additional 200 bolt 2 images were captured on the simulated platform (Figure 15b). To improve model generalization, a Mosaic augmentation strategy was applied, expanding the dataset to 1344 images. The data were split into training, validation, and test sets at a 7:2:1 ratio, yielding 1137, 224, and 112 images, respectively. With LabelImg, 648 fasteners were annotated and assigned to two categories. Semantic segmentation masks were produced with Labelme.

5. Experimental Results and Analysis

5.1. Evaluation Metrics

The experiments were run on an NVIDIA GeForce RTX 4090 GPU (24 GB RAM) and an Intel Core i9-13900KF CPU @ 2.50 GHz. The software environment comprised Python 3.11.9, PyTorch 2.2.1, and CUDA 12.4. Training employed the modified DSEW-YOLOv8n.yaml configuration with a batch size of 32, 300 epochs, and an input resolution of 640×640 .

Label smoothing was set to 0.1 to mitigate over-fitting, the learning rate to 0.001, and weight decay to 5×10^{-4} .

For bolt detection, six metrics were used: Precision (P), Recall (R), mean Average Precision at IoU = 0.5 (mAP@0.5), number of parameters, computational cost (GFLOPs), and frames per second (FPS). Precision is the proportion of correct predictions among all predictions, whereas Recall is the proportion of correct predictions among all ground-truth positives; these are computed as:

$$P = \frac{T_p}{T_p + F_p} \quad (11)$$

$$R = \frac{T_p}{T_p + F_N} \quad (12)$$

where T_p is the number of true positives, F_p the number of false positives, and F_N the number of false negatives.

mAP@0.5 is the mean of the Average Precision (AP) for all classes at an IoU threshold of 0.5, calculated as:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \times 100\% \quad (13)$$

where N is the total number of classes; AP is the area under the precision–recall curve; and mAP is the mean of all AP values.

For anti-loosening-line extraction, four metrics were adopted: mean IoU (mIoU), mean Pixel Accuracy (mPA), mean Precision (mP), and FPS. The first three measure accuracy, whereas FPS reflects deployability. The specific calculation formulas are given below:

$$IoU_i = \frac{TP_i}{TP_i + FP_i + FN_i} \quad (14)$$

$$mIoU = \frac{1}{N} \sum_{i=1}^N IoU_i \quad (15)$$

where TP_i denotes the number of correctly predicted pixels of class i , FP_i the missed pixels of class i , FN_i the misclassified pixels of class i , and N the total number of classes.

mPA evaluates the model at the pixel level and is defined as the mean pixel accuracy over all classes, where PA_i is the pixel accuracy of class i .

$$PA_i = \frac{TP_i}{TP_i + FP_i} \quad (16)$$

$$mPA = \frac{1}{N} \sum_{i=1}^N PA_i \quad (17)$$

mPrecision measures predictive precision; it is obtained by calculating class-wise precision values and averaging them.

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (18)$$

$$mPrecision = \frac{1}{N} \sum_{i=1}^N Precision_i \quad (19)$$

FPS gauges runtime efficiency: a higher FPS indicates faster image processing, where T denotes the time required to process a single image.

$$FPS = \frac{1}{T} \quad (20)$$

5.2. YOLOv8 Experiments for Small-Bolt Detection

To assess the DS-EMA module under different configurations, several comparative experiments were designed. First, the EMA module was inserted either after the C2f block in the neck or after the C2f block in the backbone, and its impact on detection—especially small-object detection—was evaluated. Second, we compared three variants—plain C2f, C2f + EMA, and DS-EMA with depthwise-separable convolutions—focusing on improvements in small-object accuracy.

5.2.1. Effect of EMA Placement

The EMA module was placed either in the neck or in the backbone after the corresponding C2f block; results are summarized in Table 2.

Table 2. Detection results of YOLOv8n with EMA placed at different locations (neck vs. backbone).

Model	P (%)	R (%)	mAP (%)	Params/M	FPS/ms	FLOPs/G
YOLOv8n	92.2	85.4	90.7	3.0	151	8.1
EMA-neck	91.3	87.2	91.8	2.9	141	8.4
EMA-backbone	93.2	87.1	91.5	3.1	128	8.4

Results show that placing EMA in the backbone slightly reduced Recall and mAP@0.5 compared with placing it in the neck, but Precision rose by 1.9%. Therefore, EMA was fixed after the backbone C2f block, which offers a better trade-off for tiny-object detection.

5.2.2. Ablation on the Improved EMA Module

To confirm that EMA enhances small-object recognition and that depthwise-separable convolutions reduce parameters, we compared the plain C2f block, C2f + EMA, and the DS-EMA variant. Numerical results are listed in Table 3.

Table 3. Ablation results of EMA improvements for bolt detection.

Model	P (%)	R (%)	mAP (%)	Params/M	FPS/ms	FLOPs/G
YOLOv8n	92.2	85.4	90.7	3.0	151	8.1
EMA-neck	93.2	87.1	91.5	3.2	128	8.4
EMA-backbone	93.4	89.5	92.2	2.8	126	8.5

As shown in Table 3, the baseline C2f achieved an mAP of 90.7%. Adding EMA raised mAP to 91.5%, and the DS-EMA variant reached 92.2% while further reducing parameter count under identical compute resources. By integrating depthwise-separable convolution, DS-EMA not only improves accuracy but also lowers computational complexity, confirming its effectiveness and practicality for small-object detection.

5.2.3. Bolt-Detection Ablation Study

To verify the effectiveness of our improvements, we compared three variants by sequentially adding the DS-EMA module and the WIoU loss. Results are summarized in Table 4.

Table 4. Ablation study of DSEW-YOLOv8, showing the effect of DS-EMA and WIoU modules on detection metrics.

Model	P (%)	R (%)	mAP (%)	Params/M	FPS/ms	FLOPs/G
YOLOv8n	92.2	85.4	90.7	3.0	151	8.1
YOLOv8n+DS-EMA	93.4	89.5	92.2	2.8	126	8.5
YOLOv8n+DS-EMA+WIoU	94.8	89.2	93.5	2.8	135	8.5

As shown in Table 4, inserting DS-EMA into the backbone raised Precision, Recall, and mAP@0.5 by 1.2%, 4.1%, and 1.5%, respectively, while reducing parameters by 0.2 M. Replacing the loss with WIoU further improved Precision, Recall, and mAP@0.5 by 2.6%, 3.8%, and 2.8% over the previous YOLOv8n baseline. The final model achieved 94.8% Precision, 89.2% Recall, and 93.5% mAP@0.5, demonstrating reliable performance for tiny-bolt detection on landing door safety keepers.

To provide a clearer comparison, Figure 16 displays the precision-and-recall curves before and after the network modifications. The improved network shows a noticeably steeper initial slope, indicating a higher effective learning rate and a stronger ability to capture target features than the original model.

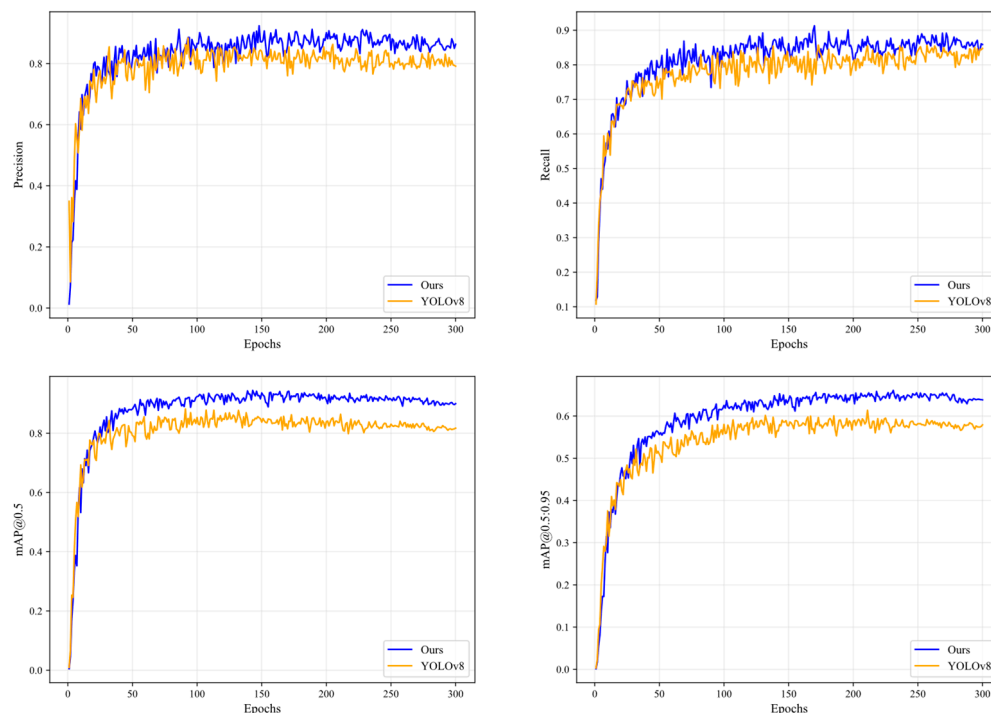


Figure 16. Training curves (baseline vs. ours) for precision, recall, mAP@0.5, and mAP@0.5:0.95.

5.2.4. Comparison with Other Detectors

To demonstrate the superiority of the proposed detector, we compared it with models of similar size—YOLOv5n, YOLOv6n, YOLOv7-tiny, and YOLOv8n—under identical datasets and settings.

Table 5 demonstrates that the proposed DSEW-YOLOv8 achieves the best overall balance among all lightweight counterparts. Compared with YOLOv8n, our method improves precision, recall, and mAP by 1.2, 4.1, and 1.5 percentage points, respectively, indicating markedly stronger capability in identifying and recalling tiny bolts under challenging conditions. Although FLOPs increase slightly (from 8.1 G to 8.5 G), the parameter count is reduced from 3.0 M to 2.8 M, and the detector still maintains a practical real-time speed of 126 FPS.

When compared with earlier lightweight baselines such as YOLOv5n, YOLOv6n, and YOLOv7-tiny, our model surpasses them by 7.0%, 6.5%, and 3.0% mAP, respectively, while only moderately increasing computational cost. Furthermore, even against the most recent lightweight variants—YOLOv9-tiny and YOLOv11n—our method remains competitive: DSEW-YOLOv8 achieves a comparable mAP (92.2%), only marginally lower than that of YOLOv11n (92.3%), while providing a substantially higher recall (1.6 percentage points higher than YOLOv11n and 0.6 percentage points higher than YOLOv9-tiny). This

highlights its robustness in minimizing missed detections, which is especially critical for safety inspection.

Overall, these results demonstrate that DSEW-YOLOv8 achieves an excellent trade-off between accuracy and efficiency, and is particularly advantageous for real-world inspection of small targets, especially elevator landing door safety-keeper bolts.

Table 5. Comparison of DSEW-YOLOv8 with YOLOv5n, YOLOv6n, YOLOv7-tiny, YOLOv8n, YOLOv9-tiny, and YOLOv11n in terms of precision, recall, mAP, parameters, FPS, and FLOPs.

Model	P (%)	R (%)	mAP (%)	Params/M	FPS/ms	FLOPs/G
Yolov5n	86.3	84.4	85.2	2.9	148	7.8
Yolov6n	86.5	84.9	85.7	3.3	150	8.0
YOLOv7-tiny	90.1	88.7	89.2	3.1	147	8.2
Yolov8n	92.2	85.4	90.7	3.0	151	8.1
YOLOv9-tiny	93.1	88.9	91.1	3.6	149	8.3
YOLOv11n	93.5	87.9	92.3	2.6	154	6.5
DSEW-YOLOv8	93.4	89.5	92.2	2.8	126	8.5

5.2.5. Visual Comparative Analysis of Bolt-Detection

To visually compare detection performance, we carried out a qualitative analysis on the curated dataset using both DSEW-YOLOv8 and YOLOv8n, and the results were visualized (Figure 17). As illustrated, the proposed model detects more critical bolts than the baseline and yields no false positives. The visual evidence confirms that DSEW-YOLOv8s provides stronger bolt-detection capability, avoiding missed detections and delivering a clear performance advantage.

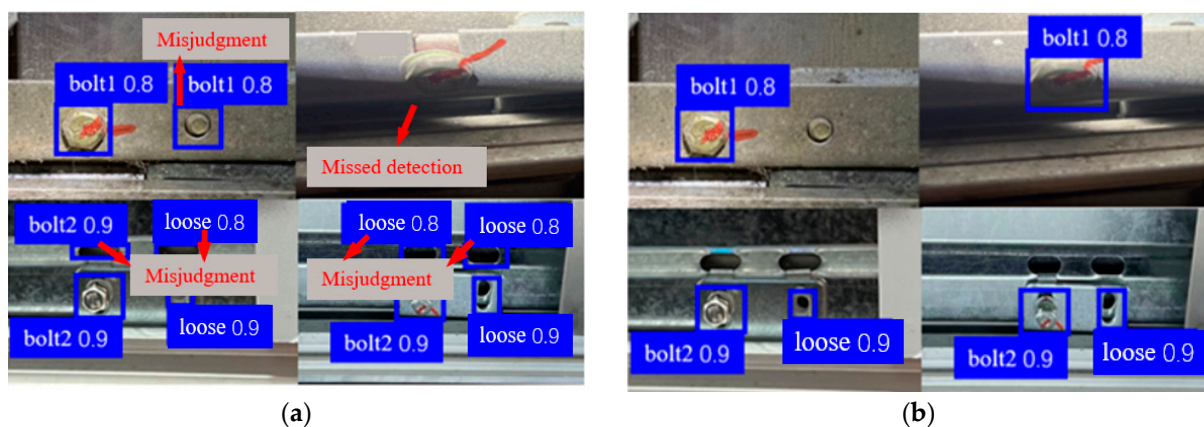


Figure 17. Visual comparison of bolt detection: (a) baseline YOLOv8n; (b) DSEW-YOLOv8, demonstrating the improved detection of small bolts and reduced missed detections by our method.

5.3. DeepLabv3+ Experiments for Anti-Loosening-Line Segmentation

5.3.1. Ablation Study on Anti-Loosening-Line Segmentation

To validate the proposed DeepLabv3+ improvements for anti-loosening-line segmentation, we successively evaluated three measures—the MobileViT backbone, GAM attention, and ASPP dilation optimization. The ablation results are listed in Table 6.

Replacing Xception with MobileViT increased mIoU from 80.4% to 82.1%, cut parameters from 57.6 M to 36.9 M, and boosted FPS from 26.5 to 56.2, confirming that MobileViT improves both accuracy and efficiency. Adding GAM attention enhanced small-target sensitivity, raising mIoU to 84.6% and mPrecision to 89.3%; parameters increased slightly to 38.5 M and FPS decreased marginally to 53.9. Finally, optimizing ASPP dilation pushed mIoU to 85.8%, mPrecision to 92.9%, and mPA to 90.5%, with negligible changes in parameters and FPS.

Table 6. Ablation study of improved DeepLabv3+ network for anti-loosening-line segmentation, showing the effect of MobileViT, GAM, and improved ASPP optimization.

Model	mIoU (%)	mPA (%)	Params/M	mP	FPS/ms
DeepLabv3+	80.4	85.5	57.6	80.1	26.5
DeepLabv3+&MobileViT	82.1	87.4	36.9	85.5	56.2
DeepLabv3+&MobileViT&GAM	84.6	88.3	38.5	89.3	53.9
DeepLabv3+&MobileViT&GAM&I-ASPP	85.8	90.5	38.5	92.9	53.7

5.3.2. Visual Comparative Analysis of Anti-Loosening-Line Detection

To visually compare segmentation performance, the baseline DeepLabv3+ and the improved model were applied to the same dataset, and the results are visualized in Figure 18. The baseline output exhibits two typical failures—Missing edge, where the tip of the anti-loosening line is lost, and Distractors, where stains or scratches are falsely segmented as lines. In contrast, the proposed model produces continuous, complete anti-loosening lines and effectively suppresses background noise. These visual findings confirm that the improved network is more robust and precise, substantially reducing both false negatives and false positives in anti-loosening-line segmentation.

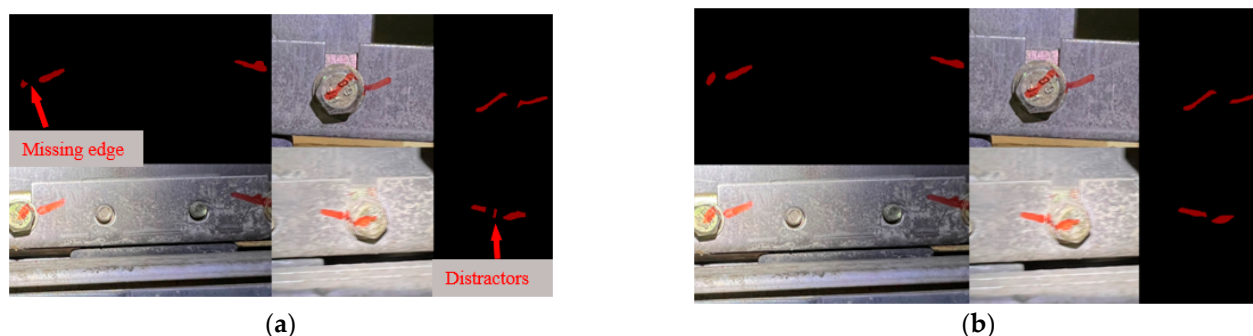


Figure 18. Visual comparison of anti-loosening-line segmentation: (a) baseline DeepLabv3+; (b) improved model.

5.3.3. Looseness Determination Results

Based on the workflow described in Section 3.2, we conducted experiments to verify the effectiveness of the proposed looseness determination method. The preprocessing and analysis pipeline, including binarization, morphological opening, and connected-component extraction, is illustrated in Figure 19. These results demonstrate how the workflow is applied in practice and how looseness of different bolt types is determined under real inspection conditions.

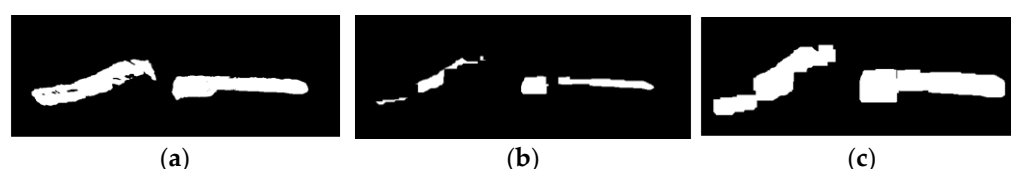


Figure 19. Preprocessing steps: (a) binarization; (b) erosion; (c) dilation. These steps correspond to the initial stages illustrated in Figure 13.

The segmentation mask is first converted to grayscale and binarized with an empirical threshold of 20, suppressing background noise while preserving the line features. A single morphological opening (erosion 20×20 followed by dilation 15×15) is then applied to the binary image to eliminate residual artifacts, as shown in Figure 19.

If no connected component remains after opening ($N = 0$), the anti-loosening line is considered occluded or absent, and the image is routed to the failure path. For images with $N \neq 0$, all contours are extracted via connected-component analysis (`cv2.findContours`), and two cases are distinguished.

(1) $N = 1$ —single-line case

Let A be the number of line pixels and A_{tot} the total number of pixels; the pixel ratio is defined as

$$r = \frac{A}{A_{\text{tot}}} \quad (21)$$

The decision threshold depends on the fastener type: $r_{\text{thr}} = 0.03$ for bolt 1 and 0.06 for bolt 2. If $r \geq r_{\text{thr}}$, the joint is classified as not loose; otherwise, it is loose. This rule handles both complete-loop and partially occluded cases.

(2) $N \geq 2$ —multi-line case

Each component is fitted with a minimum-bounding rectangle using `cv2.minAreaRect`, and the angle θ between its long edge and the W -axis is computed. For bolt 1, the two rectangles with the largest areas are selected and their angle difference is calculated as

$$\Delta\theta = |\theta_a - \theta_b| \quad (22)$$

Figure 20 visualizes this step: the two anti-loosening lines are tightly enclosed, their long-edge orientations being 30.77° and 90.00° , i.e., $\Delta\theta = 59.23^\circ$. A value of $\Delta\theta > 15^\circ$ is interpreted as loose; otherwise, the joint is not loose.

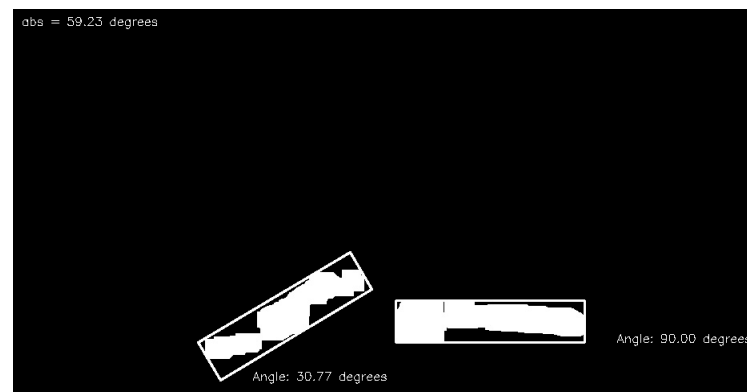


Figure 20. Multi-line case ($N \geq 2$): min-area rectangles and angle difference.

For bolt 2, the rectangles with the smallest and largest W -coordinates—corresponding to the bolt head and the fastened surface—are retained, and $\Delta\theta$ is computed in the same manner. Figure 21 illustrates the multi-rectangle selection: green boxes denote candidates, whereas blue and red boxes mark the rectangles at W_{min} and W_{max} . The resulting $\Delta\theta = 9.01^\circ$ is below the threshold; hence, the joint is classified as not loose.

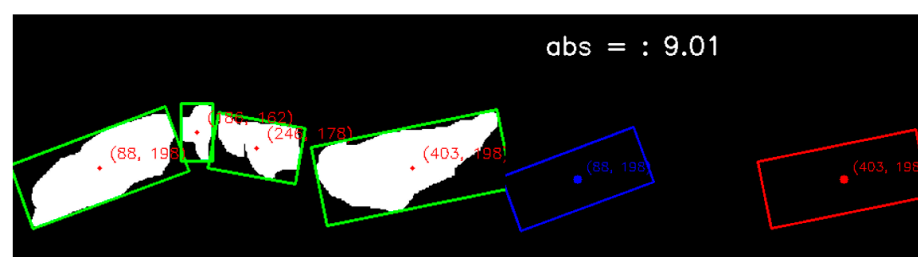


Figure 21. Bolt-2 multi-rectangle selection and angle calculation.

5.3.4. Overall Experimental Results

Looseness was assessed on the entire dataset using four binary-classification metrics: mean Intersection-over-Union (M_{IoU}), mean classification accuracy (M_{CA}), mean precision (M_P), and mean recall (M_R). These metrics are widely adopted [14] and provide a comprehensive assessment of model performance. Their formulations are given in Equations (23)–(26).

$$M_{IoU} = \frac{1}{2} \left(\frac{n_{TP}}{n_{TP} + n_{FN} + n_{FP}} + \frac{n_{TN}}{n_{TN} + n_{FP} + n_{FN}} \right) \quad (23)$$

$$M_{CA} = \frac{1}{2} \left(\frac{n_{TP}}{n_{TP} + n_{FN}} + \frac{n_{TN}}{n_{TN} + n_{FP}} \right) \quad (24)$$

$$M_P = \frac{1}{2} \left(\frac{n_{TP}}{n_{TP} + n_{FP}} + \frac{n_{TN}}{n_{TN} + n_{FN}} \right) \quad (25)$$

$$M_R = \frac{1}{2} \left(\frac{n_{TP}}{n_{TP} + n_{FN}} + \frac{n_{TN}}{n_{TN} + n_{FP}} \right) \quad (26)$$

In these equations, n_{TP} , n_{FP} , n_{FN} , n_{TN} are the entries of the 2×2 confusion matrix: true positives (correctly predicted “not loose”), false positives (incorrectly predicted “not loose”), false negatives (incorrectly predicted “loose”), and true negatives (correctly predicted “loose”), respectively. The confusion matrix aggregated over both fastener types is summarized in Table 7. The resulting metrics are listed in Table 8.

Table 7. Quantitative results of looseness determination for bolt 1 and bolt 2.

Decision Outcomes	n_{TP}	n_{FP}	n_{FN}	n_{TN}
Bolt 1	98	21	435	4
Bolt 2	57	12	273	2

Table 8. Evaluation metrics for bolt-loosening determination of two fastener types (bolt 1 and bolt 2).

Evaluation Metrics	M_{IoU}	M_{CA}	M_P	M_R
Bolt 1	0.926	0.913	0.954	0.937
Bolt 2	0.917	0.965	0.958	0.963

As reported in Table 8, the model achieves outstanding detection accuracy for both bolt 1 and bolt 2. For bolt 1, the model yields an M_{IoU} of 0.926, an M_{CA} of 0.913, an M_P of 0.954, and an M_R of 0.937, indicating high accuracy and stability in looseness recognition. For bolt 2, the corresponding scores are 0.917 (M_{IoU}), 0.965 (M_{CA}), 0.958 (M_P), and 0.963 (M_R), demonstrating the model’s robustness when dealing with more complex assemblies involving nuts and washers.

6. Conclusions

Two dedicated algorithms are proposed for elevator landing door maintenance: an enhanced YOLOv8 for small-bolt detection and an upgraded DeepLabv3+ for high-precision segmentation of anti-loosening lines. Targeted architectural and algorithmic optimizations yield significant gains in detection accuracy, segmentation quality, and computational efficiency.

For tiny-bolt detection, YOLOv8 was refined into the DS-EMA model by introducing depthwise-separable convolutions and an Efficient Multi-scale Attention (EMA) module. The EMA block markedly improves small-object precision—achieving about a 2.8% increase on bolt heads and painted lines—while preserving high inference speed and low computational cost. In comparative experiments, DS-EMA also surpasses recent lightweight detectors such as YOLOv9-tiny and YOLOv11n, delivering notably higher recall, which is

especially critical for safety inspection. Consequently, YOLOv8 attains superior accuracy and real-time performance in landing door bolt inspection.

For anti-loosening-line segmentation, an improved DeepLabv3+ was developed by adopting a MobileViT backbone and integrating a Global Attention Mechanism (GAM). Experiments demonstrate significant improvements in M_{IoU} , M_{CA} , M_P and M_R , confirming the superiority of the modified DeepLabv3+; computational efficiency is likewise enhanced, enabling accurate and fast extraction of anti-loosening lines on elevator landing doors.

Comparative studies against mainstream lightweight detectors and segmenters show that the enhanced YOLOv8 and DeepLabv3+ achieve state-of-the-art accuracy while maintaining an excellent balance between speed and model size.

In summary, the improved YOLOv8 and DeepLabv3+ perform robustly on landing door bolt detection and line segmentation, providing practical value for real-world elevator safety inspection. Future work will focus on improving robustness in more complex industrial scenarios, including adaptation to diverse elevator models and resistance to noise, stains, and occlusion.

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

Funding: This research was funded by the National Natural Science Foundation of China, grant number 52407163.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the lead author.

Conflicts of Interest: Author Lin Zou was employed by the company Dalian Boiler Pressure Vessel Inspection and Testing Research Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wang, E.; Liu, Z.; Zhang, M. Technical development and trends of China's elevator industry. *Build. Sci.* **2018**, *34*, 110–118. [\[CrossRef\]](#)
2. Li, H.; Chen, Z.; Han, J.; Du, E. Analysis of factors affecting elevator safety. *Electromech. Inf.* **2013**, *33*, 70–71+73. [\[CrossRef\]](#)
3. Pal, J.; Banerjee, S.; Chikermane, S.; Banerji, P. Estimation of fixity factors of bolted joints in a steel frame structure using a vibration-based health monitoring technique. *Int. J. Steel Struct.* **2017**, *17*, 593–607. [\[CrossRef\]](#)
4. Sun, Q.; Yuan, B.; Mu, X.; Sun, W. Bolt preload measurement based on the acoustoelastic effect using smart piezoelectric bolt. *Smart Mater. Struct.* **2019**, *28*, 055005. [\[CrossRef\]](#)
5. Huang, J.; Liu, J.; Gong, H.; Deng, X. A comprehensive review of loosening detection methods for threaded fasteners. *Mech. Syst. Signal Process.* **2022**, *168*, 108652. [\[CrossRef\]](#)
6. Liu, X.; Deng, Z.; Yang, Y. Recent progress in semantic image segmentation. *Artif. Intell. Rev.* **2019**, *52*, 1089–1106. [\[CrossRef\]](#)
7. Zhang, Y.; Sun, X.; Loh, K.J.; Su, W.; Xue, Z.; Zhao, X. Autonomous bolt loosening detection using deep learning. *Struct. Health Monit.* **2020**, *19*, 105–122. [\[CrossRef\]](#)
8. Kong, X.; Li, J. Image registration-based bolt loosening detection of steel joints. *Ital. Natl. Conf. Sens.* **2018**, *18*, 1000. [\[CrossRef\]](#)
9. Huynh, T.C.; Park, J.H.; Jung, H.J.; Kim, J.T. Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing. *Autom. Constr.* **2019**, *105*, 102844. [\[CrossRef\]](#)
10. Lyu, M. Study on Bridge Connecting Bolt Loosening Detection Method Based on Image Processing and Deep Learning. Master's Thesis, Southwest Jiaotong University, Chengdu, China, 2021; pp. 1–122. [\[CrossRef\]](#)
11. Wang, Q.; Wang, R.; Li, H.; Cao, H.; Liu, C. Visual detection method of rail vehicle bolt looseness and pre-tightening force. *J. Railw. Sci. Eng.* **2023**, *20*, 3511–3524. [\[CrossRef\]](#)
12. Liu, C.; Wu, Y.; Liu, J. Research progress on visual detection methods for bolt/rivet faults. *Chin. J. Sci. Instrum.* **2025**, *46*, 143–160. [\[CrossRef\]](#)

13. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [\[CrossRef\]](#)
14. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV 2018), Munich, Germany, 8–14 September 2018; pp. 3–19. [\[CrossRef\]](#)
15. Yang, X.; Hou, L.; Yang, J.; Sun, H.; Yan, J.; Guo, Z. R3Det: Refined single-stage detector with feature refinement for rotating object. *IEEE Trans. Image Process.* **2021**, *30*, 2918–2930. [\[CrossRef\]](#)
16. Ming, Q.; Miao, L.; Zhou, Z.; Song, J.; Pizurica, A. Gradient Calibration Loss for Fast and Accurate Oriented Bounding Box Regression. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5611. [\[CrossRef\]](#)
17. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; Sang, N. BiSeNet V2: Bilateral network with guided aggregation for real-time semantic segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 3051–3068. [\[CrossRef\]](#)
18. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12077–12090.
19. Yaseen, M. What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. *arXiv* **2024**, arXiv:2408.15857. [\[CrossRef\]](#)
20. Chen, L.-C.; Zhu, Y. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv* **2018**, arXiv:1802.02611. [\[CrossRef\]](#)
21. Zhu, S.; Zhou, Y. MRP-YOLO: An improved YOLOv8 algorithm for steel surface defects. *Machines* **2024**, *12*, 917. [\[CrossRef\]](#)
22. Gulzar, Y. Fruit image classification model based on MobileNetV2 with deep transfer learning technique. *Sustainability* **2023**, *15*, 1906. [\[CrossRef\]](#)
23. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 22–25 July 2017; pp. 1800–1807. [\[CrossRef\]](#)
24. Wang, C.-Y.; Bochkovskiy, A. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696. [\[CrossRef\]](#)
25. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the ICASSP 2023—IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5. [\[CrossRef\]](#)
26. Luong, M.-T.; Pham, H. Effective approaches to attention-based neural machine translation. *arXiv* **2015**, arXiv:1508.04025. [\[CrossRef\]](#)

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.