

Article

A Lightweight Method for Road Defect Detection in UAV Remote Sensing Images with Complex Backgrounds and Cross-Scale Fusion

Wenya Zhang ¹, Xiang Li ^{1,*}, Lina Wang ², Danfei Zhang ¹, Pengfei Lu ¹, Lei Wang ¹ and Chuanxiang Cheng ¹

¹ Institute of Surveying and Mapping, Information Engineering University, Zhengzhou 450052, China; zwy1741386917@163.com (W.Z.); fei18337279378@163.com (D.Z.); pfpengfeilu@163.com (P.L.); gala8wll@126.com (L.W.); chuanxiangcheng@163.com (C.C.)

² School of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou 450001, China; lina@zzuli.edu.cn

* Correspondence: lixiangzzchxy@163.com

Abstract

The accuracy of road damage detection models based on UAV remote sensing images is generally low, mainly due to the challenges posed by the complex background of road damage, diverse forms, and necessary computational requirements. To tackle the issue, this paper presents CSGEH-YOLO, a lightweight model tailored for UAV-based road damage detection in intricate environments. (1) The star operation from StarNet is integrated into the C2f backbone network, enhancing its capacity to capture intricate details in complex scenes. Moreover, the CAA attention mechanism is employed to strengthen the model's global feature extraction abilities; (2) a cross-scale feature fusion strategy known as GFPN is developed to tackle the problem of diverse target scales in road damage detection; (3) to reduce computational resource consumption, a lightweight detection head called EP-Detect has been specifically designed to decrease the model's computational complexity and the number of parameters; and (4) the model's localization capability for road damage targets is enhanced by integrating an optimized regression loss function called WiseIoUv3. Experimental findings indicate that the CSGEH-YOLO algorithm surpasses the baseline YOLOv8s, achieving a 3.1% improvement in mAP. It also reduces model parameters by 4% and computational complexity to 78%. In contrast to alternative methods, the model proposed in this paper significantly reduces computational complexity while improving accuracy. It offers robust support for deploying UAV-based road damage detection models.

Keywords: road damage detection; UAV remote sensing images; YOLOv8s; complex background; lightweighting; data enhancement; cross-scale feature fusion



Academic Editors: Massimo Losa and Nicholas Fiorentini

Received: 14 May 2025

Revised: 25 June 2025

Accepted: 28 June 2025

Published: 30 June 2025

Citation: Zhang, W.; Li, X.; Wang, L.; Zhang, D.; Lu, P.; Wang, L.; Cheng, C. A Lightweight Method for Road Defect Detection in UAV Remote Sensing Images with Complex Backgrounds and Cross-Scale Fusion. *Remote Sens.* **2025**, *17*, 2248.

<https://doi.org/10.3390/rs17132248>

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

The road network serves as a fundamental infrastructure supporting national economic development and public livelihood. However, the continuous growth in traffic volume and network expansion has led to escalating management costs for road safety maintenance. Pavement distresses, primarily encompassing various cracks and potholes, are typically assessed and analyzed using traditional manual methods to gauge pavement conditions. Nevertheless, these approaches are both time-intensive and costly. If not promptly detected and managed, such distresses can inflict severe damage on national road network assets and compromise the safety of pedestrians and vehicles [1]. Meanwhile,

serving as a mobile remote sensing platform, UAVs have emerged as a potent supplement to handheld and vehicle-mounted equipment in road damage detection. This is due to their compact size, cost-effectiveness, ease of operation, high mobility, and capability for high-resolution imaging. When combined with deep learning techniques, UAV remote sensing image target detection algorithms can not only classify and pinpoint specific targets within the images but also offer superior efficiency and performance. Consequently, they are extensively employed by researchers in the field of road damage detection [2,3].

Early road damage detection techniques predominantly relied on digital image processing methods, such as threshold-based segmentation [4] and regional growth-based segmentation algorithms [5]. Nevertheless, these traditional approaches encounter substantial challenges and limitations. Firstly, they necessitate the manual extraction of damage features, a task that proves particularly arduous in complex and ever-changing natural road environments. The absence of a unified model further impedes their ability to simultaneously detect multiple types of road damage, thereby significantly compromising their practicality. Primarily, the traditional methods' limitations are also evident in their inability to adapt to diverse and complex road conditions, leading to inconsistent detection results that struggle to meet accuracy requirements. In addition, manual inspection methods are fraught with numerous limitations. They not only demand substantial labor and time investments but also yield detection results that are susceptible to variations in inspectors' skills and experience, introducing a significant subjective element. This, in turn, diminishes the feasibility of their practical application.

With the swift ascent of deep learning, traditional detection algorithms have found it increasingly challenging to meet the soaring demand for detection capabilities. In contrast, deep learning has made a significant impact in the field of target detection, showcasing exceptional performance. The technology integrates classification and localization to precisely identify and pinpoint the target object. Currently, it is broadly categorized into two main types: regression-based one-stage detection models and region-based two-stage detection approaches. Region-based two-stage detection methods focus on generating candidate regions, i.e., rectangular boxes that potentially encompass the target. They then ascertain the target's category and location by leveraging CNNs for feature extraction and classification. Prominent two-stage target detection methods encompass Fast R-CNN [6], R-FCN [7], and Mask R-CNN [8]. Conversely, one-stage detection algorithms are more streamlined and efficient. They require only a single forward pass through a convolutional network to enable the direct prediction of both the object's class and its spatial localization within the image. The primary advantage of one-stage algorithms is their rapid detection speed, making them ideal for real-time applications, particularly for deployment on platform-embedded devices like UAVs. However, their accuracy typically falls short of that of two-stage detection algorithms. Common one-stage target detection algorithms include the YOLO [9,10] series, SSD [11], and RetinaNet [12].

Numerous scholars have delved into research on road damage detection algorithms, yielding a series of notable achievements. Jiang [13] integrated the Multi-Head Self-Attention (MHSA) mechanism with a bottleneck structure into CSPNet, crafting a novel backbone module named CSPBoT. This innovation significantly bolsters the model's capability to extract global features. Furthermore, the incorporation of the adaptive spatial feature fusion (ASFF) module and the adoption of the VariFocal loss function markedly enhance the detection accuracy of various roadway damage. Wu [14], addressing the inefficiencies of traditional methods and the high costs associated with equipment deployment in roadway damage detection, proposed the YOLO-LWNet network based on a lightweight design. They validated this lightweight model in complex road scenarios, offering a valuable reference for achieving real-time road damage detection on mobile

terminal devices. Liu [15] introduced the lightweight backbone network RepViT-M1.5, the improved ConvNeXtV2-C2f (CN2C2f) module, and the novel inverted residual EMA (iEMA) attention mechanism into the RIEC-YOLO model. This integration significantly elevates the accuracy of road damage detection while maintaining a low computational cost. Wang [16] proposed the RCP-YOLO algorithm, which combines a partial Transformer with a multiple aggregated trajectory attention mechanism. The algorithm integrates a novel attention mechanism, and incorporates a cross-stage partial Transformer module, denoted as CSP-TB, alongside a reconfigured feature pyramid network termed Re-Calibration FPN. The sophisticated design substantially enhances the detection accuracy of road defects within the RDD2022 dataset while fulfilling the requisites for real-time detection. Ruggeri [17] embedded a CBAM within the YOLO11 model structure to enhance the network's focus on key image details. Simultaneously, the convolution mechanism was optimized using C3k2 blocks to reduce computational load. Additionally, neck feature fusion was enhanced by introducing cross-stage partial blocks with spatial attention (C2PSA). This approach substantially enhances both the precision and the recall metrics in damage detection, all while preserving computational efficiency.

However, current challenges in road damage detection encompass (1) complex scenes involving interfering elements like pedestrians, vehicles, uneven lighting, and tree branch occlusions, which often result in errored or missed detections; (2) the difficulty in feature extraction stemming from notable disparities in the shape and dimensions of road damage; and (3) the struggle to enhance detection accuracy without significantly increasing computational costs, leading to elevated computational demands and complexity. Consequently, accurately extracting the intricate details of road damage information has emerged as a focal area of research for many scholars.

2. Related Works

2.1. Algorithmic Solutions

Building upon prior research, this paper presents a lightweight UAV road damage detection model named CSGEH-YOLO. This model is specifically designed to tackle the challenges inherent in complex scenes in UAV remote sensing imagery, the diverse patterns of road damage, and the requirement for high computational efficiency. (1) In this model, the Star Operation Module of StarNet is incorporated into the backbone network, C2f. This integration enhances the model's capacity to capture fine details in complex scenes. Moreover, a contextual anchored attention mechanism is incorporated to boost the extraction of global features. (2) An enhanced generalized feature pyramid (GFPN) is developed. This GFPN effectively combines cross-scale feature information, thus addressing the issue of the diverse scales of targets in road damage detection. (3) To optimize the use of computational resources, a lightweight detection head (EP-Detect) is designed. This EP-Detect notably lessens the computational complexity and the quantity of parameters within the model. Additionally, by integrating the regression loss function WiseIoUv3, the model's accuracy in localizing road damage targets is further improved.

2.2. Data Enhancement

Road damage depicted in UAV remote sensing images not only presents a complex background but is also subject to significant noise interference, often causing confusion with the road background within the scene and thus acting as a camouflaged target. Therefore, to augment the data volume for model training, align with real-world application scenarios, and comprehensively assess the effectiveness of the algorithm, we performed data augmentation and expansion on the training-validation subset within the UAPD dataset, resulting in a new dataset, termed UAPD-AUG. This enhancement aims to bolster

the model's generalization performance, mitigate the effects of overfitting, and ensure that the model's training outcomes are more conducive to practical deployment. The data augmentation techniques employed in this experiment primarily involve rotating and scaling the data at various angles, performing horizontal and vertical flips, introducing Gaussian noise, compressing image quality, and adjusting image brightness and contrast. The image undergoes random rotation within an angular range of 0° to 90° , horizontal and vertical flipping by 180° , noise addition with an intensity ranging from 20% to 60%, quality compression to 50%, and adjustment of the maximum brightness contrast value to 60%.

These operations effectively doubled the data volume of the original training-validation set. Subsequently, a data cleaning process was conducted to eliminate distorted images and those with missing annotations following the augmentation. Consequently, the UAPD-AUG dataset was finalized. The enhanced UAPD-AUG dataset comprises 4284 images exhibiting road damage features. Figure 1 illustrates exemplars of both the original and augmented road damage images, providing a visual comparison of the augmentation effects.

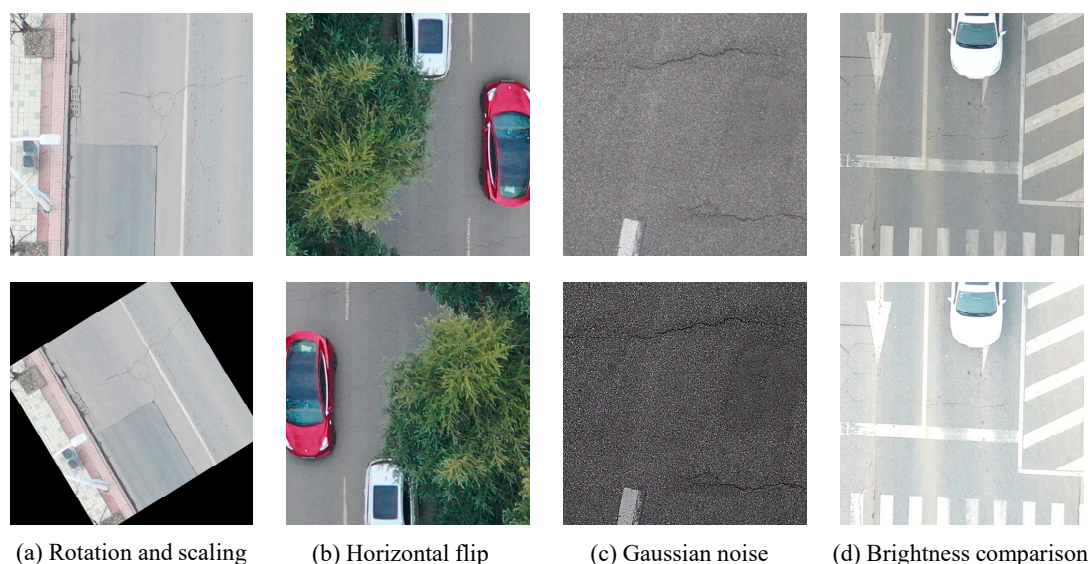


Figure 1. Sample images before and after data enhancement.

3. Materials and Methods

Considering the characteristics of road damage scenes depicted in UAV remote sensing images, including the complexity as well as the diversity in types and scales, it is challenging to accurately extract the features of different damage. In response to this, this study proposes the C2f-Star-CAA (CSC) feature extraction structure within the backbone network architecture of YOLOv8s [18]. This structure is intended to notably boost the model capability to extract features of various road damage. The StarNet achieves richer feature representations via the star operation, especially for small-scale targets and within intricate scenes, consequently enhancing the detection performance. Moreover, the CAA mechanism is employed to capture more contextual information regarding damage types. Secondly, a cross-scale feature fusion strategy (GFPN) is incorporated into the neck. This strategy connects different layers and scales through cross-scale operations, which promotes the combination of multi-level and cross-scale features. As a result, the information transfer of damage features at different scales is effectively enhanced. Furthermore, this paper designs the EP Head lightweight detection head. This module effectively reduces the amount of computation and the parameter count of the algorithm while elevating the model's overall performance, thus achieving a lightweight architecture. Finally, this study suggests

replacing the CIoU regression loss function with the WiseIoUv3 bounding box loss function. This replacement enables the model to focus on the predictive regression of samples where the prediction boxes are difficult to fit with the real boxes and where the real boxes have low confidence levels. Consequently, the harmful gradients generated by these samples are reduced, and the accurate and efficient identification and localization in road damage detection are improved. The overall architecture of the CSGEH-YOLO network designed in this paper is illustrated in Figure 2, with the locations of the model improvements indicated in the red solid box and the corresponding modules shown in the dashed box.

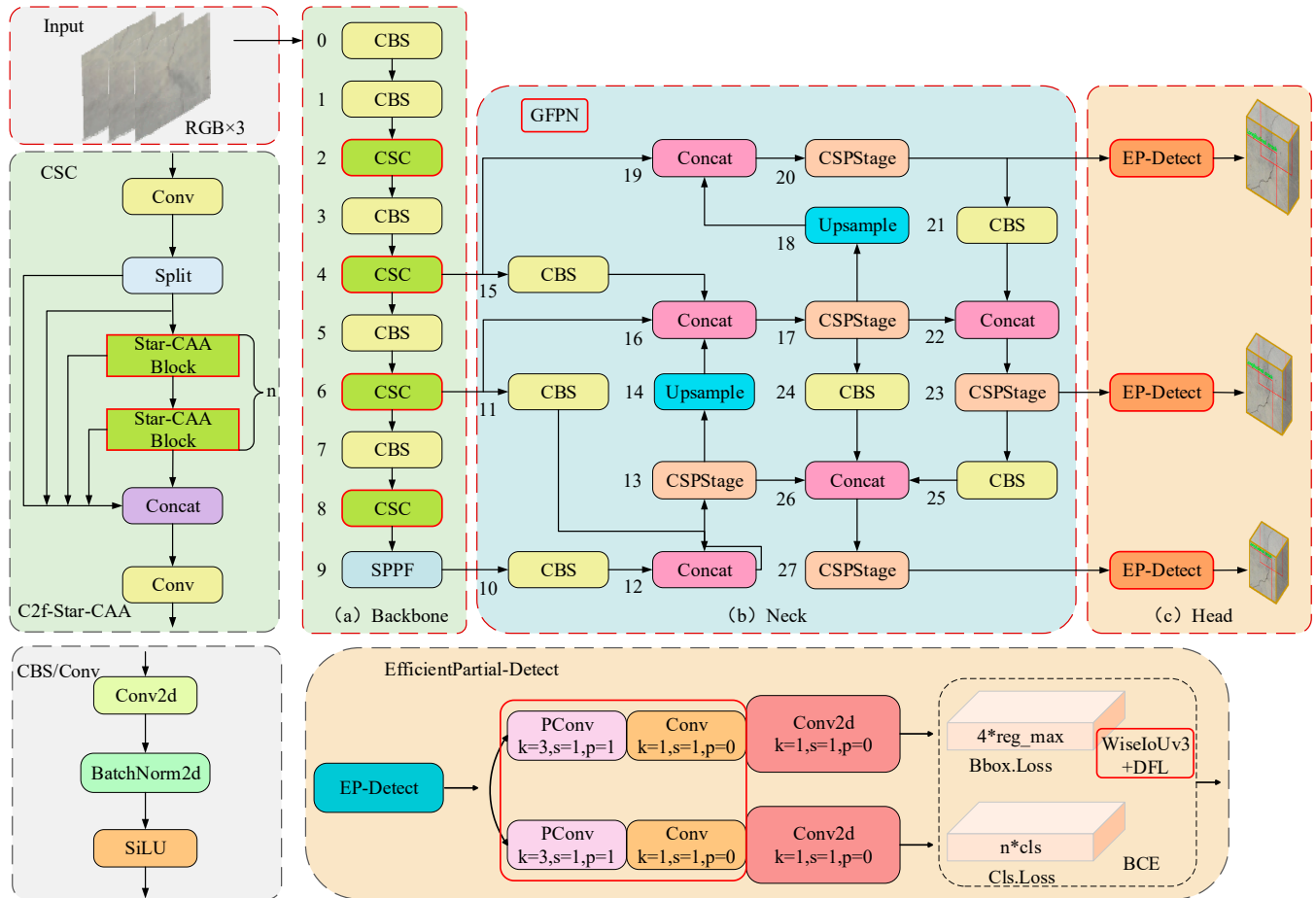


Figure 2. Overall architecture of the CSGEH-YOLO network in this paper.

3.1. CSC Feature Extraction Structure

Addressing these distinctive characteristics of road damage, this paper innovatively proposes the CSC structure, which employs StarNet to achieve high-dimensional nonlinear feature space mapping through star operations while maintaining a low computational complexity. During the feature extraction process, contextual information is paramount for the network to capture rich and semantic representations of road damage features, thereby bolstering the model's learning and inference capabilities. However, traditional convolutional feature extraction methods frequently result in the loss of contextual information. To mitigate this, this paper introduces a context-anchored attention mechanism, CAA, built upon StarNet, to capture long-range contextual feature information of road damage and augment the model's feature extraction capabilities.

3.1.1. StarNetwork

StarNet [19], an efficient and lightweight network architecture, was proposed by Microsoft in 2024. It introduces a novel approach to streamlining network design through the utilization of star operations. These star operations enable efficient feature representation, eliminating the requirement for complex feature fusion or multi-branching architectures, as depicted in Figure 3. StarNet utilizes a four-stage hierarchical framework. It makes use of convolutional layers to carry out downsampling and integrates the Star Blocks module for the purpose of feature extraction. The star operation, which is defined as element-wise multiplication, is employed to fuse features from distinct subspaces. This method involves merging features through the element-by-element multiplication of two linear transformations. By mapping input features into a high-dimensional nonlinear feature space, this operation generates a novel feature space, diverging from the conventional neural network approach of attaining high-dimensional features by augmenting the number of network channels. Instead, it aligns more closely with the polynomial kernel functions (PKFs) methodology. This is achieved by stacking multiple layers of the star operation, with each layer substantially increasing the number of implicit high-dimensional complex features.

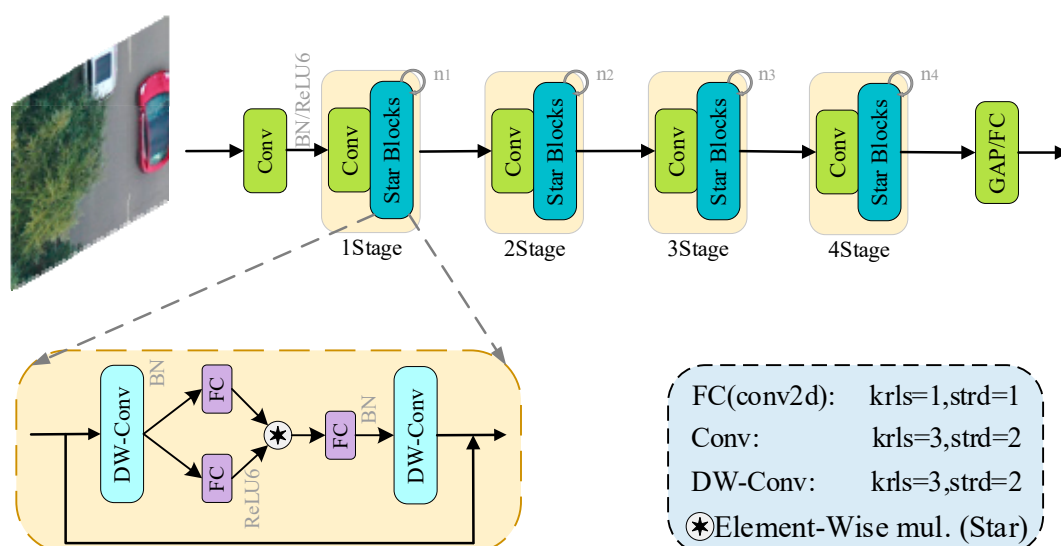


Figure 3. StarNet network architecture.

This approach facilitates a richer semantic feature representation, particularly beneficial for small targets and complex scenes. Consequently, it offers a superior representation of road damage features in complex scenarios.

The Star Blocks module within StarNet represents a type of Depth Separable Convolution (DW-Conv). Initially, a 2D convolutional block is utilized, with a stride setting of 1, an expansion ratio of 1, and a kernel size of 1, to produce a feature information map. Subsequently, these feature maps are incorporated into the block, and an “element-level multiplication” operation, known as the “star operation”, is carried out to derive the fused features. Finally, batch normalization is implemented on the output of the 2d convolution, yielding a comprehensive feature representation. The weights of the batch normalization are initialized as 1, and the biases are initialized as 0.

3.1.2. CAA Attention Mechanism

The core architecture of the Context Anchor Attention (CAA) mechanism module [20] resides in the synergistic integration of its distinctive large-kernel global average pooling operation and one-dimensional band convolution. This compositional strategy empowers

the network to precisely capture the potential relationships between distant pixels within the image, thereby augmenting the feature learning capability of the central target region. The CAA mechanism emphasizes the contextual interdependencies among pixels at diverse locations within the road damage image, with a particular focus on learning information from distant pixels, which significantly influences detection accuracy. By doing so, it enhances the feature representation of the central region, providing robust feature support for the road damage detection task. This not only offers a more potent feature foundation for the task but also contributes to improving the overall detection performance and precision.

In the specific architecture depicted in the CSC network framework, the CAA mechanism spatially compresses the resultant feature information map through a global average pooling operation, succeeded by a 1×1 convolutional (Conv) operation to further capture and integrate feature information within local regions. To efficiently simulate and approximate the functionality of traditional large-kernel depth-wise convolution, CAA also incorporates two lightweight 2D depth-separable convolutional blocks oriented in two distinct directions. The depth-wise convolution is employed in the horizontal and vertical directions. This approach utilizes $1 \times k$ and $k \times 1$ convolution kernels to extend the receptive field along distinct directions, thereby circumventing the substantial computational overhead associated with traditional large convolution kernels (e.g., $k \times k$). Typically, the kernel size (ks) is determined based on the maximum dimension of the target object or the anticipated context range, with a common choice being $ks = 11$. The stride is generally set to 1 to ensure full pixel coverage of the input.

This configuration enables efficient capture and integration of contextual pixel information along both the horizontal and the vertical, which is particularly advantageous for recognizing and extracting features from target areas with elongated or specialized shapes (e.g., transverse cracks, potholes). Consequently, it better fulfills the requirements for detecting roadway lesions with varying shapes and scales.

3.1.3. CSC Architecture

In the original network architecture, the core of the C2f module comprises n Bottleneck layers, as illustrated in Figure 4. However, this structure is constrained in its ability to extract feature information from complex backgrounds and identify various types of road damage, given that each Bottleneck module within it contains merely two 3×3 standard convolutional layers. To facilitate rich feature representation and extraction of road damage, and to augment the model's reasoning capability in intricate circumstances, this study introduces the C2f-Star-CAA structure, hereafter referred to as the CSC feature extraction module. This is achieved by substituting the Bottleneck within the C2f module with an enhanced Star-CAA Block.

The CSC network architecture, as depicted in Figure 5, replaces the traditional bottleneck layer. Initially, the input features are integrated with the initial feature representations through a convolutional operation (Conv). Following the Split operation, the output from the convolutional layer is partitioned into several branches and transmitted to subsequent processing modules. Thereafter, the input features undergo processing by the initial Star-CAA Block module. Within this module, a Depthwise Separable Convolution (DWConv) layer is first applied for initial feature representation and channel reassignment. Subsequently, two pointwise Conv2D convolutional layers are employed to augment the channel dimensions, facilitating complex inter-channel interactions. One of the pointwise convolutions is activated using the ReLU6 function, followed by element-wise multiplication operations for feature interaction and fusion. The contextual information within road damage images is further extracted using the introduced CAA mechanism module. Here, a 1×1 small kernel convolution is employed to capture local features, while a set of parallel

depth-separable convolution operations are conducted to capture cross-scale contextual information within the workflow of the CAA module. Additionally, these features are integrated, and the output dimensions are adjusted through a 1×1 Conv convolutional task. The features are then output using a Sigmoid activation function. The activated features are fed through a second DWConv layer to decrease the channel count, further refining the features to align with the number of input channels. To bolster the model's generalization capacity, the DropPath regularization residual connectivity mechanism is utilized, which randomly drops a portion of the network during the training process. Furthermore, the features extracted in the preceding stage are amalgamated with the original input features via residual concatenation and feature fusion operations. The aforementioned operations are iterated in the subsequent n Star-CAA Block module processes to obtain a rich feature representation.

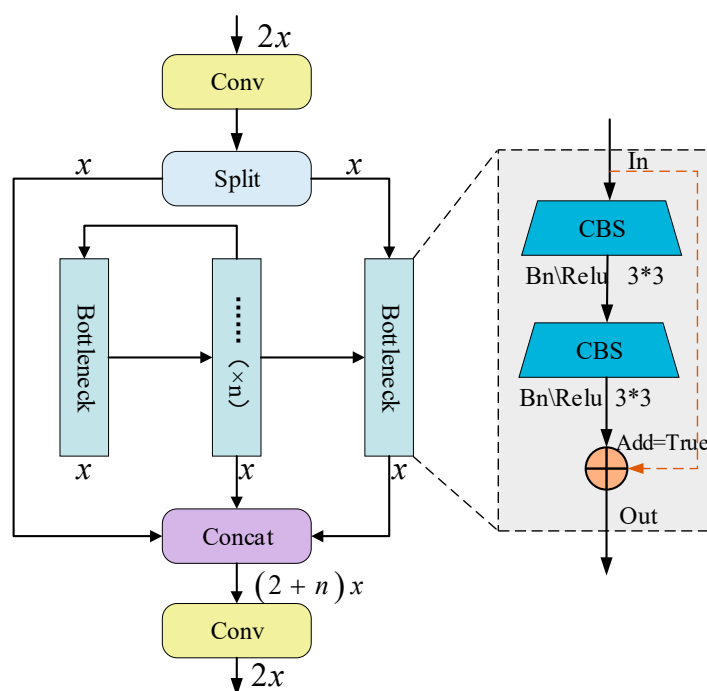


Figure 4. C2f network structure.

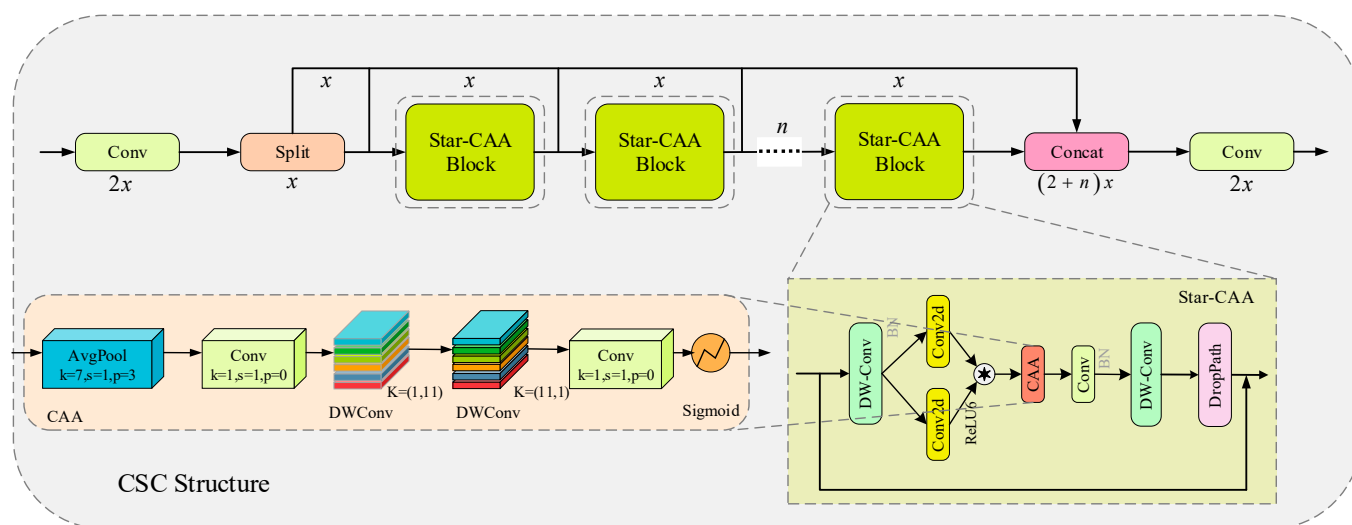


Figure 5. CSC network framework.

In summary, the outputs derived from the Star-CAA Block are concatenated with the features extracted from other branches. Features obtained through distinct pathways integrate information from multiple scales, thus strengthening the model's capacity to represent complex features. Finally, the features resulting from the Concat operation are passed through an additional convolutional layer, yielding the final output feature map.

3.2. Cross-Scale Feature Fusion Strategies

The road damage targets inherently exhibit pronounced multi-scale distribution characteristics, meaning that different types of damage vary significantly in size and, consequently, occupy a variable number of image pixels. This variability is particularly critical for the feature fusion of targets at different scales. The YOLOv8s neck network integrates features through the structure of the Path Aggregation Network (PANet) [21], as illustrated in Figure 6a, where darker colors denote the top layer of the structure and lighter colors represent the bottom layer. PANet initially facilitates the transfer of semantic information from high-dimensional to low-dimensional spaces via a top-down pathway, followed by the conduction of location information from low-dimensional to high-dimensional spaces through a bottom-up pathway. However, this architecture only fuses features at the same scale between adjacent layers and features within the same layer across neighboring scales. This limitation results in substantial semantic information loss during multi-scale feature fusion, thereby constraining the recognition performance of the original model for complex and variable damage targets. In this study, an enhanced Generalized Feature Pyramid Network (GFPN) [22] is ingeniously introduced, as depicted in Figure 6b.

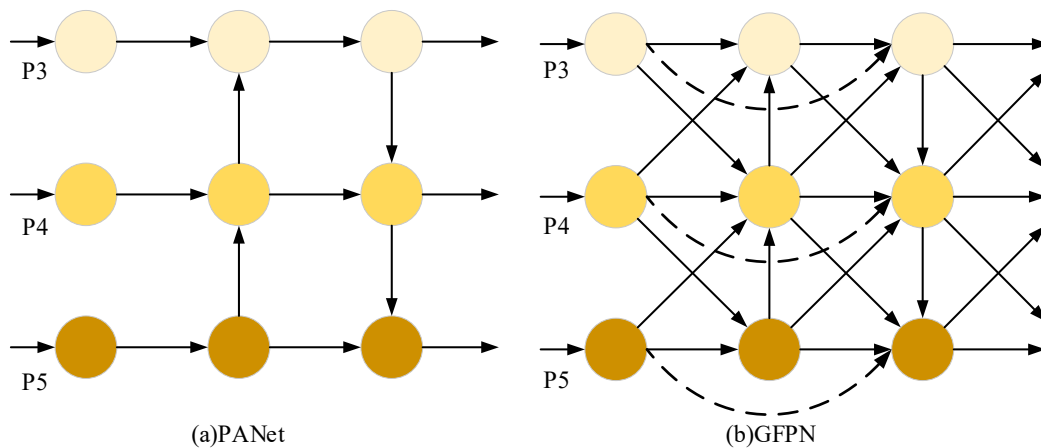


Figure 6. Structure of PANet and GFPN.

The GFPN structure incorporates cross-layer and cross-scale connections into the PANet architecture, with dashed lines indicating cross-layer connections between different layers and cross arrows denoting cross-scale connections between adjacent layers. This methodology substantially augments the holistic exploitation of multi-scale feature information. The cross-layer connection feature fusion process is mathematically represented by Equation (1).

$$P_t^n = \text{Conv}(\text{Concat}(P_t^0, P_t^1, \dots, P_t^{n-1})) \quad (1)$$

In the equation, P_t^n is the feature information of the feature map at the level of n at the level of t , $\text{Concat}(\dots)$ denotes the feature splicing produced by all the previous layers, and $\text{Conv}(\dots)$ denotes the 3×3 convolution.

In cross-scale feature fusion, accounting for the scale disparities among feature maps across different layers, an adaptive feature fusion strategy is employed to ensure the effective integration of feature information [23]. Specifically, when performing cross-layer

matching between feature maps of larger dimensions and those of smaller dimensions, the MaxPool operation is typically utilized to implement downsampling on the larger feature maps. Concurrently, the smaller feature maps are upsampled using the bilinear interpolation function to achieve the requisite cross-scale connectivity between adjacent layers, as mathematically expressed in Equation (2).

$$P_t^n = \text{Conv}(\text{Concat}(\text{MaxPool}(P_t^{n-1}), P_t^n, \text{Bilinear}(P_t^{n+1}))) \quad (2)$$

Figure 7 illustrates the specific network architecture of the GFPN, which seamlessly integrates the properties of deep semantic information and low-level spatial information through a series of operations, including multi-layer upsampling, standard convolutional block downsampling, and channel fusion. This architecture facilitates the efficient fusion of cross-scale features, enabling the model to accurately extract and represent feature information of damage at varying resolutions. The GFPN structure not only augments the expressive capacity of cross-scale road damage features but also extends the network's connectivity to deep-level features without imposing substantial computational burdens, thereby enhancing the detection performance of road damage. Furthermore, by expanding the network's connection to deep-level features, it bolsters the detection efficacy for road damage without a significant increase in computational demand.

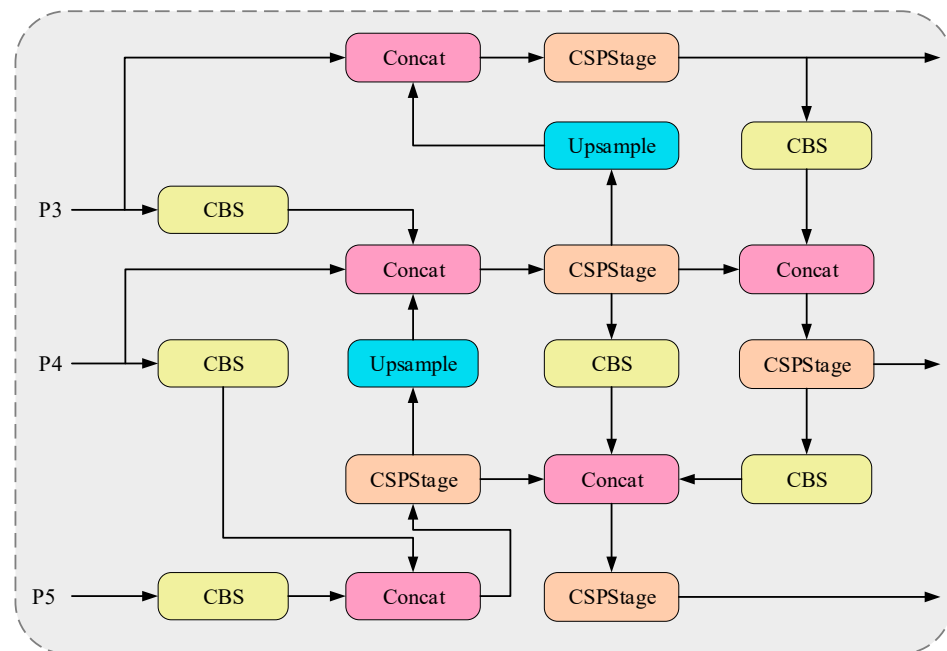


Figure 7. Overall network structure of the GFPN.

3.3. Lightweight Detection Module

Computational workload and parameter count constitute critical metrics for assessing deep learning algorithms. In the context of UAV hardware facilities, the computational workload, often referred to as computational complexity, and the parameter count both directly correlate with the consumption of computational memory resources. The architecture of the Detect head in YOLOv8s is illustrated in Figure 8. This Detect head is partially constructed by substituting the coupled head of the original YOLOv5 with a decoupled head structure, which segregates the bounding-box regression task from the classification task. Within this structural configuration, each branch incorporates two 3×3 one-dimensional convolutional (Conv) layers followed by a single 1×1 two-dimensional convolutional (Conv2d) layer. This configuration results in the prediction head account-

ing for approximately half of the entire model's computational and parametric overhead, thereby consuming a substantial amount of computational resources. Moreover, the traditional convolutional operation tends to induce redundancy in high-dimensional feature maps, subsequently leading to computational overload.

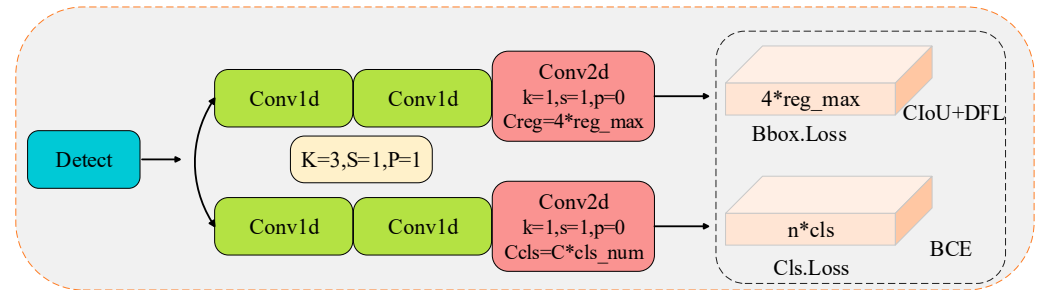


Figure 8. The structure of the Detect head.

Addressing the aforementioned challenges, this paper draws inspiration from the literature [24] and proposes an effective lightweight detection head, termed Efficient-Partial Detect (EP-Detect), as depicted in Figure 9. This architecture replaces the original convolutional block with partial convolution (PConv), which incorporates a single 1×1 pointwise convolution while utilizing the same convolutional kernel. The fundamental principle of PConv involves initially partitioning the input channels and performing convolutional operations exclusively on local channels to extract spatial features, while maintaining the features of the remaining channels (i.e., constant mapping) instead of directly discarding their information. These retained features subsequently contribute to the pointwise convolutional layers, facilitating feature propagation across all channels. Ultimately, the feature maps are fused into a novel representation by incorporating weighted combinations along the depth dimension. Consequently, the computational complexity of PConv is markedly reduced compared to that of standard convolution. The integration of PConv with pointwise convolution enables more efficient extraction of spatial features associated with road damage by mitigating memory access and computational redundancy, thereby significantly reducing both the computational and the parametric overhead of the model. Specifically, the computational FLOPs and memory accesses of PConv can be mathematically expressed as follows:

$$i_{\partial} \times j_{\partial} \times k_{\Delta}^2 \times c_{\partial p}^2 \quad (3)$$

$$i_{\partial} \times j_{\partial} \times 2c_{\partial p} + k_{\Delta}^2 \times c_{\partial p}^2 \approx i_{\partial} \times j_{\partial} \times 2c_{\partial p} \quad (4)$$

where i_{∂} , j_{∂} , and $c_{\partial p}$ show the height, width, and count of partial channels within the feature layer, respectively, and k_{Δ} denotes the dimension of the convolution kernel. The computational volume demanded by PConv exhibits a positive correlation with the quantity of partial channels $c_{\partial p}$. Consequently, the computational volume of the network model can be effectively managed by controlling this variable. Additionally, reducing convolutional operations leads to a decrease in memory access, which comprehensively underscores the efficacy of the PConv and pointwise convolutional structure devised in this study. This structure not only mitigates the memory load within the context of edge computing but also optimizes resource utilization, thereby demonstrating its practical applicability and robustness.

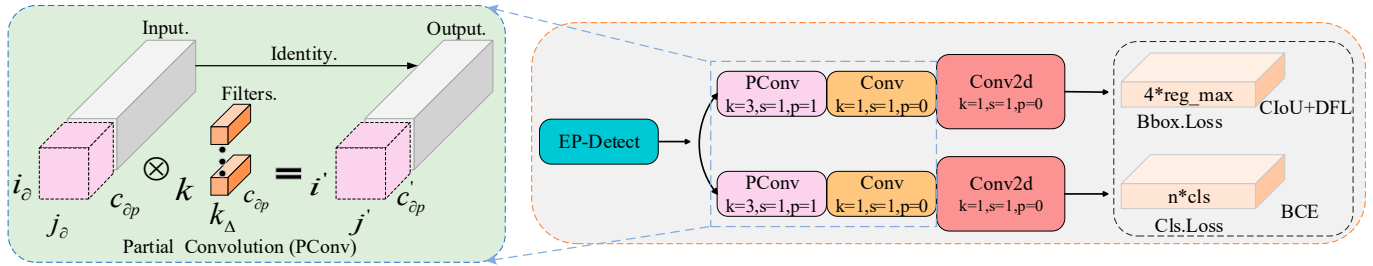


Figure 9. Structure of the EP-Detect head.

3.4. Regression Loss Function Optimization

The YOLOv8s loss function is composed of two components: the classification loss (Cls loss) and the regression loss (Bbox loss), as delineated within the dashed box in Figure 9. Specifically, the classification loss component employs the Binary Cross Entropy Loss (BCE Loss), which serves to measure the divergence between the model's output and the ground truth label in binary classification tasks. Regarding the regression loss, it is formulated by integrating the integral form derived from the Distribution Focal Loss (DFL) and the CIoU Loss [25], which collectively represent the Bbox regression loss. The amalgamated loss function is mathematically expressed in Equation (5):

$$Loss = iLoss_{Bbox} + jLoss_{Cls} + kLoss_{DFL} \quad (5)$$

where i , j , and k are the weights for the bounding box regression loss, classification loss, and DFL, respectively, with defaults of 7.5, 0.5, and 2. Given that the bounding box regression loss carries the highest weight, enhancing the accuracy of its calculation assumes paramount importance.

The inclusion of the bounding box aspect ratio as a penalty factor in the Complete Intersection over Union (CIoU) loss function for bounding box regression expedites the regression process of preselected boxes to some extent. However, when converging towards a linear ratio between the height and width of the preselected and ground truth boxes, it may impede the simultaneous adjustment of height and width in the predicted boxes during regression. This issue is particularly pronounced for geometric metrics of elongated road lesions, including distance and aspect ratio, as it intensifies the penalty applied to low-quality road lesion samples, consequently undermining the model's capacity for generalization. To address this, this paper proposes substituting the CIoU with the WiseIoUv3 bounding box regression loss function [26], abbreviated as WIoUv3.

WiseIoUv3 incorporates a dynamic, non-monotonic focusing mechanism that assesses the quality of anchor frames by leveraging the outlier degree β , in lieu of the original Intersection over Union (IoU), to automatically allocate gradient gain values. This approach effectively mitigates the competitive and penalty gradients between high-quality and low-quality prediction frames, optimizes the learning of common sample prediction frames, and enhances the model's localization performance for road damage targets. The formulation for WIoUv3 is presented as follows:

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

$$L_{WIoUv3} = r \cdot R_{WIoU} L_{IoU}, r = \frac{\beta}{\delta \cdot \alpha^{\beta - \delta}} \quad (7)$$

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{W_g^2 + H_g^2}\right) \quad (8)$$

$$L_{IoU} = 1 - IoU \quad (9)$$

where the outlier β evaluates the pre-selected box quality and assigns the gradient gain. L_{WIoUv3} is the nonmonotonic focusing loss function of the bounding box, r is the nonmonotonic focusing coefficient, and δ and α are both hyperparameters. Specifically, when $\beta = \delta$, the value of δ ensures that $r = 1$. Moreover, when the outlier degree of the anchor box meets the condition $\beta = C$ (where C denotes a constant), the gradient gain attains its maximum value. R_{WIoU} signifies the execution weight, which is employed to dynamically modulate the focus on different road damage sample prediction frames within the WIoUv3 loss function. This mechanism optimizes the model's penalty gradient allocation, thereby enhancing the overall detection accuracy. The parameters associated with R_{WIoU} are illustrated in Figure 10, which depicts the separation of W_g and H_g from the computational graph. Given that L_{IoU} represents dynamic values, the quality segmentation criteria for bounding boxes are inherently dynamic as well. Consequently, WIoUv3 can be dynamically adjusted to the most adaptive gradient gain allocation strategy, which is custom fitted to the current situation. This characteristic renders WIoUv3 particularly well suited for the prediction and regression of road damage samples across varying shape scales.

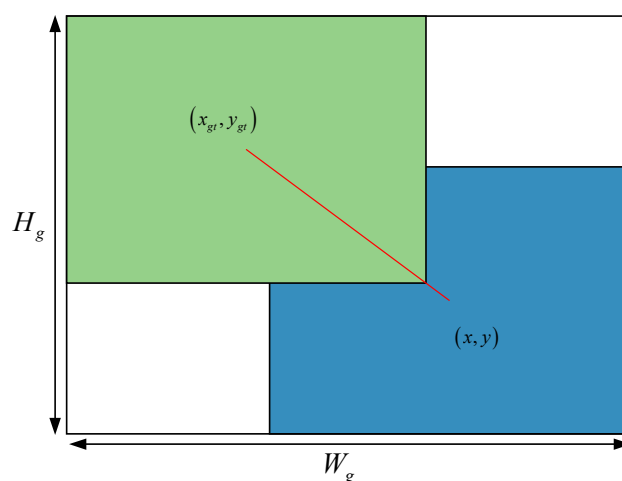


Figure 10. Schematic diagram of WiseIoU parameters.

4. Results

4.1. Introduction and Handling of Datasets

To validate the efficacy of the proposed methodology, this paper employs the open-source UAPD [27] dataset for algorithm testing. The UAPD dataset was curated by Southeast University utilizing a DJI M600 Pro hexacopter drone, equipped with a Sony Alpha 7R III digital camera, to capture images along Dong-ji Avenue in Nanjing, China. The raw drone imagery comprises 300 large-scale road damage images, each measuring 7952×5304 pixels². To ensure the preservation of the original dimensions of road damage features while aligning with the practical requirements of the inspection task, these images were subsequently resized to a resolution of 512×512 pixels² through cropping. The UAPD dataset categorizes pavement distress into six distinct classes: alligator crack, longitudinal crack, oblique crack, pothole, repair, and transverse crack. The publicly accessible dataset encompasses a total of 3151 aerial images exhibiting road damage characteristics. The UAPD dataset is partitioned into a training-validation set and a test set in an 8:2 ratio, with 10% of the dataset allocated for validation purposes. Consequently, the dataset is ultimately partitioned into training, test, and validation sets with an approximate ratio of 7:2:1, ensuring a balanced distribution for comprehensive model evaluation.

Given the correlation between photogrammetric parameters and the dimensions of road damage, the flight configuration parameters associated with the UAPD dataset were summarized and incorporated into the digitized footprint, as presented in Table 1. The parameters of the UAV, encompassing flight-related settings and camera configurations, must be accurately determined to guarantee the quality of the captured images.

Table 1. The UAPD digitized footprint.

Resolution (pixels ²)	Photogrammetric Area (m ²)	Flight Altitude (m)	Flight Speed (m/s)	Frontal Overlap	Sensor Size (mm ²)	Focal Length (mm)	Shutter Speed (s)
7952 × 5304	26 × 20.57	30	5.1425	75%	35.9 × 24	35	1/1200

4.2. Experimental Setup

To guarantee the performance optimization and facilitate effective training of the model, all experiments conducted in this study adhered to a consistent hardware configuration and software environments. The hardware setup employed for the experiments comprised an Intel Core i7-11700K processor, an NVIDIA RTX 3070 Ti graphics processing unit (GPU) equipped with 8 GB of video memory, and 32 GB of running memory. The software environments included the Windows 10 operating system, the VSCode software platform, the Pytorch 2.0.0 framework, Python version 3.9.16, and the accelerated computing architecture supported by CUDA 11.7. The experimental parameters were configured as follows: an input image resize of 512 × 512 pixels, a fixed batch size of 300 training epochs, optimization via SGD with an initial learning rate of 0.01, a momentum coefficient of 0.937, and a weight decay of 0.0005. The selection of these parameters was executed with meticulous precision to guarantee the robustness and replicability of the experimental outcomes.

The evaluation metrics for detection algorithms can mainly be classified into two categories: model complexity and detection accuracy. Model complexity is typically assessed through the quantification of floating point operations (FLOPs) and the aggregate count of model parameters (Parameters). Specifically, a higher value of FLOPs or Parameters indicates greater model complexity, whereas a lower value suggests a more lightweight model architecture. Model detection accuracy, on the other hand, is assessed using metrics including Precision (P_{re}), Recall (R_{ec}), and mean Average Precision (MAP). The mathematical formulations for Precision and Recall are presented in Equation (10) and Equation (11), respectively.

$$P_{re} = \frac{TP_{re}}{TP_{re} + FP_{re}} \quad (10)$$

$$R_{ec} = \frac{TP_{re}}{TP_{re} + FN} \quad (11)$$

In this study, the mAP is employed as a metric to assess the overall performance of the target detection model. The mAP represents the average precision across all categories within the dataset, as calculated in Equation (12). Here, “cls” denotes the quantity of categories encompassed within the dataset, and the Average Precision (AP) for each category is derived by integrating the area under the Pre-Rec curve.

$$mAP = \frac{1}{cls} \times \sum_{k=1}^c AP_k \quad (12)$$

4.3. Ablation Experiments

To comprehensively validate the effectiveness of the algorithm, this study performed a series of module ablation experiments in the UAPD dataset, utilizing YOLOv8s as the baseline model (Baseline). An in-depth exploration was conducted on the CSC (A), GFPN

(B), and Efficient-PConv Head (C) modules, along with the integration of the WiseIoUv3 (D) regression loss function, to assess their impact on the overall performance of the model. In total, five sets of ablation experiments were carried out in this study, and the experimental results are presented in Table 2.

Table 2. Ablation experiments.

Algorithm	CSC	GFPN	EH	WiseIoUv3	Params/M	FLOPs/10 ⁹	mAP@0.5/%
Baseline					11.4	28.8	59.5
+A	✓				12.6	30.9	61.2 (+1.7)
+A+B	✓	✓			12.5	29.6	61.6 (+2.1)
+A+B+C	✓	✓	✓		11.0	22.6	62.1 (+2.6)
+A+B+C+D	✓	✓	✓	✓	11.0	22.6	62.6 (+3.1)

As illustrated in the table above, the first experiments utilized the YOLOv8s algorithm as the baseline model, yielding an mAP@0.5 score of 59.5%. Subsequent experiments were conducted and analyzed in comparison to this baseline. In the second experiments, the C2f module in the original network was supplanted by the CSC structure. This alteration enhanced the model feature extraction capability, leading to a 1.7% increase in the mAP@0.5 score. Building upon the second experiments, the third set introduced the GFPN cross-scale feature fusion network strategy. This integration not only reduced the model's parameter count and computational burden but also effectively addressed the challenge of cross-scale target detection for road damage in drone imagery. Consequently, the mAP@0.5 score was elevated by 0.4%. In the fourth set of experiments, an Efficient-PConv Head (EH) detection head was designed to mitigate the computational load and parameter quantity while ensuring accuracy improvement. By employing Partial Convolution (PConv), which convolves only a portion of channels, the redundancy of feature maps was mitigated, and the accuracy of target localization was enhanced. This optimization enabled the model to reduce the computational load and processing time while curtailing the parameter count, thereby conserving computational resources. Finally, in the fifth set of experiments, the Wise-IoUv3 loss computation function was incorporated to precisely localize the target area without affecting the parameter count or computational load. This integration raised the overall algorithm accuracy, with mAP@0.5 reaching 62.6%, representing a 3.1% enhancement over the baseline YOLOv8s model. This outcome underscores the suitability of the Wise-IoUv3 loss function for drone road damage detection tasks.

The ablation experiments collectively substantiate the rationality of the algorithm designed in this study, effectively balancing detection accuracy, computational load, and parameter count.

As shown in Table 3, which provides a comparative analysis of the experimental outcomes between the baseline model and the algorithm proposed in this study, it becomes apparent that the enhanced CSGEH-YOLO algorithm exhibited improvements in mAP50-95 across all categories. Notably, in the Pothole category, which had the fewest instances in the test set, the algorithm achieved an improvement of 5.1%. Furthermore, despite the challenges posed by the Inclined Crack category—characterized by limited feature information and a complex background, resulting in a small number of instances—the algorithm still attained a 1% enhancement in average detection accuracy.

Table 3. Comparison of UAPD categories mAP@0.5:0.95. Note: crk denotes crack, unit/%.

Model	All	Alligator crk	Longitudinal crk	Oblique crk	Pothole	Repair	Transverse crk
Baseline	36.2	16.5	38	14.1	39.1	64.3	42.9
Ours	37.4	18.2	38.2	15.1	44.2	66.7	44.3

4.4. Comparison of Data Enhancement Results

To rigorously substantiate the robustness and feasibility of the algorithm, data augmentation strategies were implemented on the UAPD dataset. Moreover, to comprehensively elucidate whether the performance metrics of the modules in this experiment were enhanced due to the diversity introduced by the augmented dataset or due to the inherent performance of the modules proposed herein, a series of ablation experiments were conducted on the UAPD-AUG dataset.

As presented in Table 4, several key observations can be made. Firstly, the baseline YOLOv8s model, when evaluated on the UAPD-AUG dataset, demonstrated a 2.5% improve in mAP@0.5 compared to its counterpart on the original dataset. This emphasizes the effectiveness of data augmentation in boosting the performance of the baseline. Upon integrating the C2f structure within the baseline main network into the Star-CAA module, the model became more adept at capturing distant contextual information and fully extracting multi-deformation feature information pertinent to road defects. Consequently, mAP@0.5 increased by 0.6% relative to the original model. Furthermore, by introducing the GFPN cross-scale feature fusion strategy to optimize and fuse high-level semantic information with shallow spatial features, both mAP@0.5 and mAP@0.5:0.95 witnessed increases of 1% and 1.8%, respectively, albeit with a slight rise in computational cost and parameter count compared to the original model. Secondly, the incorporation of the newly designed Efficient-Partial Head detection head into the YOLOv8s model yielded improvements in mAP@0.5 and mAP@0.5:0.95, registering increases of 1.7% and 1.9%, respectively, compared to other modules. Notably, this enhancement was accompanied by a reduction in both parameter quantity and computational cost, with the parameter decreasing by 1.8M and the computational cost being reduced by 25.3%. This fully demonstrates that the PConv partial convolution operation not only cuts down on the quantity of parameters and computational expenses but also boosts accuracy, thus offering robust substantiation for the deployment of UAV road defect detection systems. Subsequently, the bounding box loss in the original YOLOv8s network was redesigned, and the WiseIoUv3 weight function was introduced. This effectively enhanced the model's attention to road defects that are challenging to learn and prone to errors due to complex backgrounds, enabling the model to fully capture the features of road defects. Consequently, mAP increased by 0.7%, indicating the applicability of this loss function to UAV road defect detection. Finally, after integrating all four modules into the baseline model, the CSGEH-YOLO algorithm achieved optimal performance, exhibiting a 2.5% improvement over YOLOv8s.

Table 4. UAPD-AUG ablation experiment.

Algorithm	Params/M	FLOPs/10 ⁹	mAP@0.5/%	mAP@0.5:0.95/%
YOLOv8s	11.4	28.8	63	38.2
+CSC	12.6	30.9	63.6	38.2
+GFPN	12.1	29.3	64	40
+EH	9.6	21.5	64.7	40.1
+WiseIoUv3	11.4	28.8	63.7	39.4
CSGEH-YOLO	11.0	22.6	65.5	40.3

The experimental findings reveal that both the incorporation of a single module and the simultaneous integration of four modules can effectively enhance the performance of YOLOv8s. Notably, the most pronounced improvement was observed when all four modules were added concurrently. This outcome unequivocally demonstrates that the CSC feature extraction structure and the GFPN and Efficient-PConvHead modules, along with the optimization of the WiseIoUv3 loss function, jointly contributed to the performance augmentation of the CSGEH-YOLO algorithm.

Under identical experimental conditions, a comparative analysis was conducted between the algorithm and the YOLOv8s, utilizing datasets both before and after data augmentation. As indicated in Table 5, due to the relative simplicity of the road defect images prior to data augmentation, the YOLOv8s model was prone to premature convergence during the training phase. Consequently, the model attained mAP@0.5 and mAP@0.5:0.95 accuracies of 59.5% and 36.2%, respectively, on the UAPD dataset. Following data augmentation, the original model's accuracy on the UAPD-AUG dataset improved to 63% and 38.2%, representing increases of 3.5% and 2%, respectively. Furthermore, upon implementing data augmentation, the accuracy of the algorithm also exhibited a notable enhancement of 2.9%. These findings collectively suggest that incorporating a more diverse and comprehensive set of road defect data samples into the training-validation sets through data augmentation can mitigate overfitting to a certain extent. Additionally, data augmentation exerts a positive influence on the improvement in model accuracy metrics.

Table 5. Comparative experiments before and after UAPD data enhancement.

Dataset	Modeling Approach	mAP@0.5/%	Params/M	FLOPs/G	mAP@0.5:0.95/%
UAPD	YOLOv8s	59.5	11.4	28.8	36.2
	CSGEH-YOLO	62.6	11.0	22.6	37.4
UAPD-AUG	YOLOv8s	63 (+3.5)	11.4	28.8	38.2 (+2.0)
	CSGEH-YOLO	65.5 (+2.9)	11.0	22.6	40.3 (+2.9)

A confusion matrix acts as a visual depiction of an algorithm's performance, wherein each row of the matrix corresponds to an actual category, and each column denotes a predicted category. The values situated on the main diagonal of the matrix reflect the overall accuracy in correctly identifying the pest categories. The lower left triangular region of the matrix signifies the categories that the model failed to detect. A high value in this region implies that the model had a tendency to miss these categories, indicating that it did not accurately identify the genuine existing targets. Conversely, the upper right triangular region pertains to instances of model misdetection, suggesting that the model may have had a substantial number of misclassified categories, such as detecting a specific damage category as background or another target.

A comparative analysis of the confusion matrices for YOLOv8s and the proposed method, CSGEH-YOLO, on the UAPD-AUG test set is illustrated in Figure 11. In this figure, darker hues denote higher accuracy for the corresponding categories. By juxtaposing the main diagonals of the confusion matrices for the two models, it becomes evident that the proposed method exhibited a marked advantage in overall accuracy. Specifically, for potholes, mesh cracks, and inclined cracks, the detection accuracy was augmented by 11%, 8%, and 5%, respectively.

Moreover, as depicted in Figure 11, the proposed method attained the highest classification accuracy for transverse cracks, with an accuracy rate of 85% and only 13% confusion with the background category. Apart from the repair category, which showed a marginal dif-

ference compared to YOLOv8s, the proposed method achieved an accuracy improvement across all other categories. Among these, inclined cracks exhibited the lowest classification accuracy, with significant confusion between them and the background. Additionally, rotated images were more prone to confusion in distinguishing between transverse and inclined cracks, leading to 11% of inclined cracks being misclassified as transverse cracks and 3% as longitudinal cracks. Despite an 8% increase in accuracy, mesh cracks remained the category most susceptible to confusion with the background, indicating that the model often misclassified mesh cracks as background. Potholes, on the other hand, demonstrated the most substantial accuracy improvement, of 11%, resulting in an 11% reduction in confusion with the background. However, 6% of potholes were still misclassified as mesh cracks.

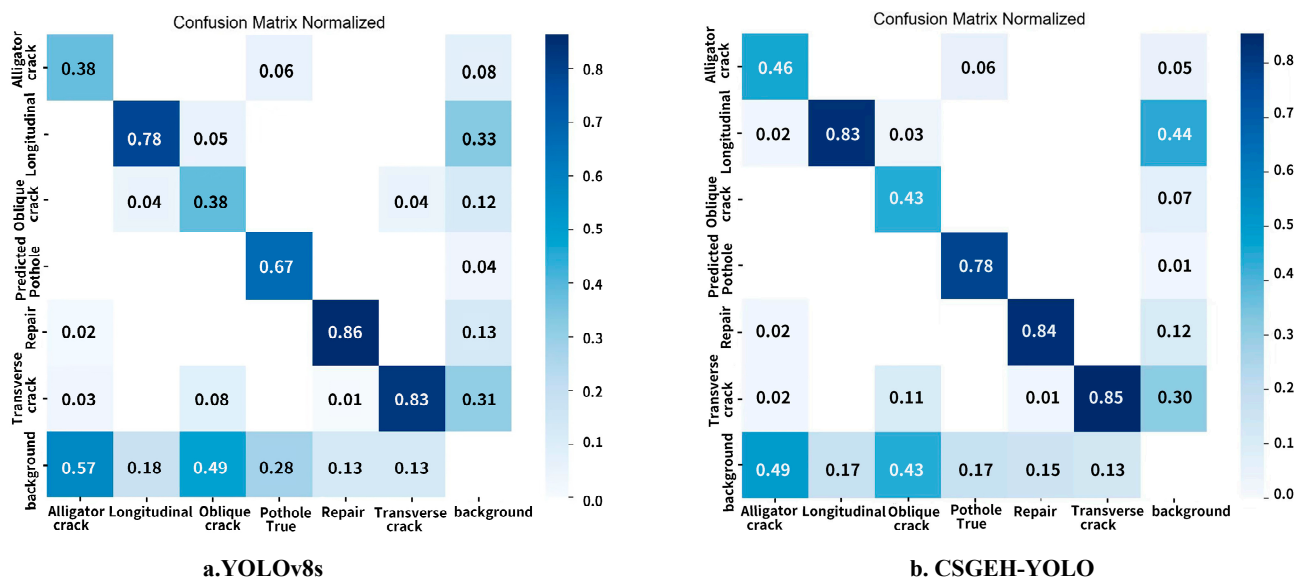


Figure 11. Test set confusion matrix comparison diagram.

4.5. Comparison Experiments

To further substantiate the superiority of the enhanced algorithm, a comparative analysis was undertaken between the enhanced algorithm, CSGEH-YOLO, and several cutting-edge object detection algorithms, namely YOLOv3-tiny [28], YOLOv5s [29], YOLOv6s [30], YOLOv7-tiny [31], RT-DETR-l [32], YOLOv8n [18], YOLOv9s [33], and YOLOv10s [34], as well as the recently introduced YOLO11n and YOLO11s [35] models. This comparison was carried out under the premise of maintaining the same dataset and experimental conditions, utilizing the UAPD dataset for the evaluation. The outcomes are delineated in Table 6, wherein the optimal performance metrics are emphasized in boldface.

As evidenced by the table, the proposed CSGEH-YOLO model outperformed its counterparts, including the more complex RT-DETR-l model; anchor box-based detection algorithms such as YOLOv3-tiny, YOLOv5s, YOLOv6s, and YOLOv7-tiny; and popular anchor-free detection algorithms like YOLOv8n, YOLOv9s, and YOLOv10s, along with the latest iterations of YOLO11's n and s series models. Specifically, CSGEH-YOLO achieved the highest recall rate, mAP@0.5, and mAP@0.5:0.95 values. In comparison with the original YOLOv8n, CSGEH-YOLO demonstrated a significant improvement of 5.2% in mAP@0.5. Furthermore, a comparison with mainstream anchor-free algorithms, YOLOv9s and YOLOv10s, reveals that a reduction in parameter quantity did not necessarily achieve a balance between accuracy enhancement and the mitigation of computational costs. For instance, the YOLO11n model attained the lowest parameter quantity and computational cost but fell short in accuracy. Conversely, while the YOLO11s model attained the highest precision rate, it suffered from missed detections and necessitated an increase in both

parameter quantity and computational cost. These observations offer valuable insights for subsequent researchers aiming to improve upon existing models.

Table 6. Comparison of algorithm performances on the UAPD.

Algorithms	Precision\%	Recall\%	mAP@0.5\%	mAP@0.5:0.95\%	Params\M	FLOPs\G
YOLOv3-tiny	60.2	53.8	57.4	32.1	9.5	14.3
YOLOv5s	60.4	55	59.1	35	9.1	23.8
YOLOv6s	51.4	53	52.6	30.9	16.2	44.2
YOLOv7-tiny	55.4	53.3	54.5	31.5	6.2	13.7
RT-DETR-l	52.9	57.4	52.9	30.6	28.4	100.3
YOLOv8n	58.6	60	57.4	34.6	3.2	8.7
YOLOv9s	62.8	58.3	60.7	36.6	7.1	26.4
YOLOv10s	60.9	54.4	56.6	33.6	7.2	21.6
YOLO11n	55.6	53.9	56.3	33.8	2.5	6.3
YOLO11s	64.8	55	59.2	35.1	9.4	21.5
CSGEH-YOLO	64.1	61.9	62.6	37.4	11	22.6

In summary, the proposed unmanned aerial vehicle image-based road defect detection model, CSGEH-YOLO, exhibits superior detection performance for defect targets in complex scenarios, coupled with notable advantages in terms of computational cost. It effectively balances accuracy and computational efficiency, thereby demonstrating high practicality.

Figure 12 illustrates a comparative analysis of the detection outcomes obtained by the algorithm, alongside the most intricate RT-DETR-l model and the top-performing YOLOv5s, YOLOv9s, and YOLO11s models. Upon examining the four sets of road defect images, the following observations can be made: In group (a), the algorithm proposed herein not only precisely localized the transverse cracks but also attained the highest confidence score. In group (b), when dealing with mesh cracks against a complex background, the proposed algorithm demonstrated a notably superior detection performance over that of the other algorithms, particularly in identifying longitudinal cracks. In group (c), the inclined crack defects within the image background exhibited strong camouflage characteristics and were susceptible to noise interference. Despite these challenges, the proposed algorithm accurately identified the most elusive inclined cracks without any false positives. In group (d), the images contain numerous small branches, which were highly prone to being mistaken for the background. The proposed algorithm not only avoided false detections but also achieved the highest confidence level for repair-type defects.

These findings collectively suggest that the CSGEH-YOLO model possesses a superior capacity for extracting shallow features. Whether confronted with complex and dynamic backgrounds or targets exhibiting certain interference, the algorithm can effectively mitigate missed detections and false positives, accurately and efficiently identifying the types of defects while demonstrating robust anti-interference capabilities.

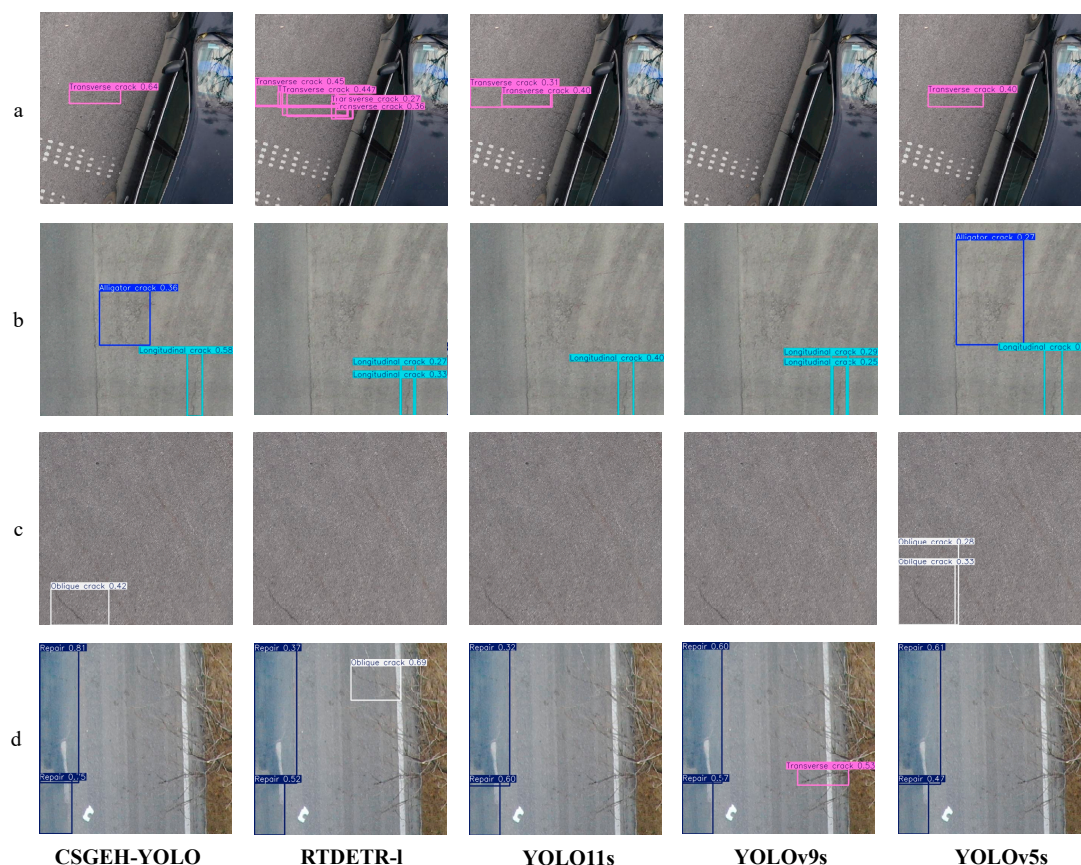


Figure 12. Detection result comparison among distinct algorithms.

4.6. Model Generalizability Validation

To further verify the practical generalizability of the algorithm and investigate the road damage detection algorithm for UAV imagery presented in this study, road damage data acquired from Chinese UAVs within the RDD-2022 dataset [36] were selected to assess the experimental detection capability of the model's generalizability. The images in this dataset encompass five distinct categories of road damage: D00 (longitudinal cracks), D10 (transverse cracks), D20 (alligator cracks), D40 (potholes), and Repair (repairs). Given the limited volume of data in this dataset, comprising only 2401 UAV road damage images, data augmentation was performed under identical conditions to those outlined in this study, thereby expanding the dataset to a total of 4282 augmented RDD-2022 images. To maintain the authenticity of the experimental setup, these augmented data were randomly stratified into training, test, and validation sets, maintaining the same proportional allocation as utilized throughout this study. The experimental data are detailed in Table 6, wherein the optimal values are accentuated in boldface for clarity.

The results in Table 7 once again underscore the superior performance of the CSGEH-YOLO model in this research for road damage detection using UAV imagery. Despite a marginal disparity in the repair category, the model still achieved the highest detection accuracy index among the five categories. Furthermore, the model's precision, recall rates, and precision mAP50 for the remaining four categories were enhanced to varying extents, with all overall performance indicators for all categories exhibiting significant improvement. Notably, the detection capability for the D40 (pothole) category, which had the least amount of data, demonstrated the most substantial enhancement. Specifically, it showed a 4.8% increase in precision, an 8.4% enhancement in recall, and a 3.4% increase in mAP50 compared to the baseline model YOLOv8s. Regarding the overall detection

performance for road damage categories, the precision rate, recall rate, and mAP50 metric increased by 1.1%, 1.0%, and 1.6%, respectively.

Table 7. RDD2022-Drone dataset generalization ability test.

Categories	YOLOv8s			CSGEH-YOLO		
	Precision (%)	Recall (%)	mAP50 (%)	Precision (%)	Recall (%)	mAP50 (%)
D00	67	69.7	65.7	70.2 (+3.2)	71 (+1.3)	70.7 (+5.0)
D10	73.2	78.3	80.5	74.1 (+0.9)	79.1 (+0.8)	81(+0.5)
D20	50.8	44	44.7	51.7 (+0.9)	44(+0.0)	46.4 (+1.7)
D40	86	61.6	67.8	90.8 (+4.8)	70 (+8.4)	71.2 (+3.4)
Repair	75.3	82.6	84.7	71.1	76.7	82.2
all	70.5	67.2	68.7	71.6 (+1.1)	68.2 (+1.0)	70.3 (+1.6)

Figure 13 illustrates the comparison of detection results from a generalizability experiment involving five groups of different road damage types. The top imagery displays the ground truth, which represents the manually labeled valid values. The middle imagery illustrates the detection outcomes yielded by the baseline model, while the bottom imagery presents the outcomes of the proposed algorithm. Through this comparison, several observations can be made. In group a, the monotonous background led to confusion between longitudinal cracks and the pavement background, leading to the cracks being easily misidentified as background objects. However, the proposed algorithm could effectively detect these cracks. In group b, the right-hand side of the transverse cracks was obscured by shadows, creating significant interference. Nevertheless, the proposed algorithm accurately identified the crack location with the highest confidence level. For groups c and d, despite the highly interfering background environments, the algorithm neither produced false detections nor missed any damage, precisely identifying mesh cracks and pothole damage. In group e, both the baseline and the proposed algorithm could correctly detect repair types under complex backgrounds, but the algorithm revealed a higher confidence level.

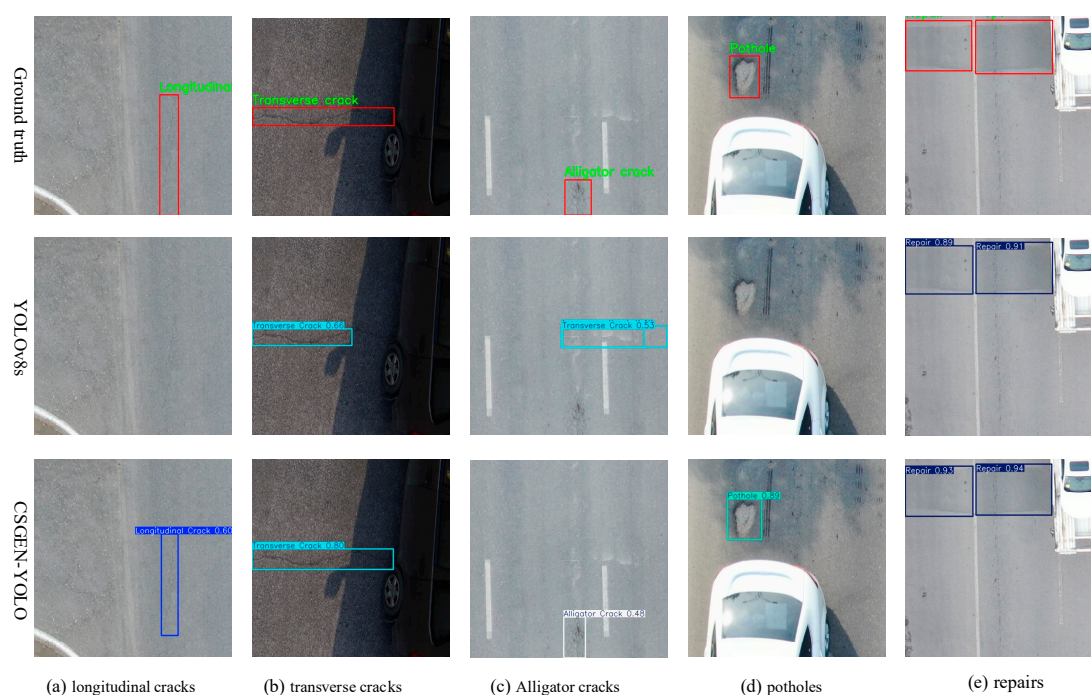


Figure 13. Comparison plot of the detection of the generalization experiment.

These results reveal that the CSGEH-YOLO model has superior generalization performance. It can accurately detect road lesions even under the interference of complex scenes and shadow coverage, and it also provides higher confidence in its detections, rendering it highly valuable for practical applications.

5. Discussion

5.1. Theoretical Mechanisms of Performance Enhancement

The experimental findings reveal that the CSGEH-YOLO model demonstrates exceptional performance in road damage detection within complex scenes. This model offers the advantages of reduced computational and parametric overheads, coupled with higher accuracy, effectively conserving computational resources. It strikes a good balance between enhancing model accuracy and simultaneously diminishing computational demands, thereby alleviating the computational burden. The following enhancements and innovations have been introduced to the baseline model:

- (1) This paper innovatively introduces the CSC structure, which facilitates the mapping of high-dimensional nonlinear feature spaces through the integration of the star operation and the CAA attention mechanism. This integration is particularly beneficial for generating richer semantic feature representations, especially for small targets and complex scenes. The enhanced feature representations significantly improve the detection performance, enabling the improved network to exhibit superior shallow spatial information extraction capabilities for road damage features within complex backgrounds.
- (2) In this study, a cross-scale feature fusion strategy is devised, incorporating multilayer up-sampling, standard convolutional block down-sampling, and channel fusion operations. This strategy effectively integrates the characteristics of deep semantic information and shallow spatial information, achieving efficient cross-scale feature fusion. The proposed structure not only enhances the expressive capability of multi-scale road damage features but also extends the network's connectivity to deep-level features.
- (3) In response to the computational burden imposed by the decoupling head in the baseline model, while recognizing that partial convolution (PConv) reduces feature map redundancy by convolving only a subset of channels without affecting the remaining channels, this paper ingeniously designs a lightweight detection head, EP-Detect. This improvement effectively reduces the model's computational cost, fully leveraging the device's computational performance to alleviate the computational burden. Simultaneously, it achieves efficient aggregation of global spatial information.
- (4) This paper optimizes the bounding box regression loss function by introducing the dynamic non-monotonic focusing WIoUv3 weighting function. This enhancement effectively focuses the model's attention on challenging and interfering road damages within complex backgrounds, prompting the model to comprehensively learn the characteristics of road damage. Consequently, it significantly improves the detection accuracy index while accelerating model convergence and maintaining the model's parameter count and computational volume.

5.2. Limitations and Future Directions

The detection efficacy of the algorithm presented in this paper still demonstrates potential for enhancement, particularly under circumstances marked by an excessive number of target damage categories and overlapping features across distinct categories. Future research endeavors will primarily center on the following two pivotal aspects:

On the one hand, considering the characteristic variability in target angles inherent in UAV remote sensing imagery, multi-modal data fusion strategies will be adopted. These

encompass infrared imagery data, laser scanning, and LiDAR applications [37], which are instrumental in mitigating the detrimental effects of adverse weather conditions, such as precipitation and snowfall, as well as fluctuations in lighting, on the imagery quality. By harnessing the complementary information embedded within different modal data, the detection algorithm will be refined to further augment the accuracy of road damage detection in UAV remote sensing imagery.

On the other hand, the research will tackle the challenge of harmonizing the enhancement of fine-grained road damage detection accuracy with the attainment of a lightweight model. This will involve a continuous optimization of the model architecture through knowledge distillation and pruning techniques, thereby elevating the practical applicability of the model.

6. Conclusions

In this study, to mitigate the issues of leakage and misdetection stemming from inconspicuous road damage types, variable scales, complex backgrounds, and strong interferences in aerial imagery acquired from low-altitude UAVs, a lightweight CSGEH-YOLO algorithm is proposed, which is based on the YOLOv8s model. The algorithm undergoes training, validation, and testing procedures on the UAPD dataset. Additionally, to assess the robustness and feasibility of the algorithm, a data-enhanced UAPD-AUG dataset is constructed. Furthermore, the data-enhanced RDD-2022 UAV dataset is utilized to assess the model generalizability. The performance of the methodology is assessed in comparison with mainstream algorithms. Relative to the baseline, the novel CSGEH-YOLO model achieves a 3.1% improvement in mAP. It achieves a reduction in the number of model parameters to 96% of those in the baseline model and decreases the computational complexity to 78%. Moreover, it demonstrates a 2.5% increase in mAP metrics for the UAPD-AUG dataset and a 1.6% increase for the RDD-2022 dataset, respectively, compared to the baseline. Consequently, it provides an effective solution for deploying a UAV-based road damage detection model.

Author Contributions: Conceptualization, W.Z. and X.L.; methodology, W.Z.; software, X.L.; validation, W.Z., X.L. and L.W. (Lina Wang); formal analysis, D.Z.; investigation, P.L.; resources, L.W. (Lei Wang); data curation, C.C.; writing—original draft preparation, W.Z.; writing—review and editing, X.L.; visualization, W.Z.; supervision, L.W. (Lina Wang); project administration, X.L.; funding acquisition, L.W. (Lina Wang). 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 42201490.

Data Availability Statement: The novel contributions delineated within this study are fully incorporated into the article. For any further inquiries, please direct them to the corresponding author.

Acknowledgments: All the authors gratefully thank the reviewers and editor for their insightful and constructive comments.

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

References

1. Huang, M.; Dong, Q.; Ni, F.; Wang, L. LCA and LCCA based multi-objective optimization of pavement maintenance. *J. Clean. Prod.* **2021**, *283*, 124583. [[CrossRef](#)]
2. Qiu, Q.; Lau, D. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. *Autom. Constr.* **2023**, *147*, 104745. [[CrossRef](#)]
3. Silva, L.A.; Leithardt, V.R.Q.; Batista, V.F.L.; Villarrubia González, G.; De Paz Santana, J.F. Automated Road Damage Detection Using UAV Images and Deep Learning Techniques. *IEEE Access* **2023**, *11*, 62918–62931. [[CrossRef](#)]

4. Devi, M.P.A.; Latha, T.; Sulochana, C.H. Iterative thresholding based image segmentation using 2D improved Otsu algorithm. In Proceedings of the Global Conference on Communication Technologies (GCCT), Thuckalay, India, 23–24 April 2015.
5. Hu, W.; Wang, W.; Ai, C.; Wang, J.; Wang, W.; Meng, X.; Liu, J.; Tao, H.; Qiu, S. Machine vision-based surface crack analysis for transportation infrastructure. *Autom. Constr.* **2021**, *132*, 103973. [\[CrossRef\]](#)
6. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015. [\[CrossRef\]](#)
7. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 379–387.
8. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
9. Redmon, J.; Farhadi, A. YoloV3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
10. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YoloV4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C. SSD: Single Shot Multibox Detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
12. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K. Focal loss for dense object detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
13. Jiang, T.-Y.; Liu, Z.-Y.; Zhang, G.-Z. YOLOv5s-road: Road surface defect detection under engineering environments based on CNN-transformer and adaptively spatial feature fusion. *Measurement* **2025**, *242*, 115990. [\[CrossRef\]](#)
14. Wu, C.; Ye, M.; Zhang, J.; Ma, Y. YOLO-LWNet: A Lightweight Road Damage Object Detection Network for Mobile Terminal Devices. *Sensors* **2023**, *23*, 3268. [\[CrossRef\]](#)
15. Liu, T.; Gu, M.; Sun, S. RIEC-YOLO: An improved road defect detection model based on YOLOv8. *Signal Image Video Process.* **2025**, *19*, 285. [\[CrossRef\]](#)
16. Wang, X.; Gao, H.; Jia, Z.; Zhao, J. A road defect detection algorithm incorporating partially transformer and multiple aggregate trail attention mechanisms. *Meas. Sci. Technol.* **2024**, *36*, 026003. [\[CrossRef\]](#)
17. Ruggieri, S.; Cardellicchio, A.; Nettis, A.; Renò, V.; Uva, G. Using Attention for Improving Defect Detection in Existing RC Bridges. *IEEE Access* **2025**, *13*, 18994–19015. [\[CrossRef\]](#)
18. Yaseen, M. What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. *arXiv* **2024**, arXiv:2408.15857.
19. Ma, X.; Dai, X.; Bai, Y.; Wang, Y.; Fu, Y. Rewrite the Stars. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 5694–5703.
20. Cai, X.; Lai, Q.; Wang, Y.; Sun, Z.; Yao, Y. Poly kernel inception network for remote sensing detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 27706–27716.
21. Liu, S.; Qi, L.; Qin, H.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
22. Jiang, Y.; Tan, Z.; Wang, J.; Sun, X. GiraffeDet: A heavy-neck paradigm for object detection. *arXiv* **2022**, arXiv:2202.04256.
23. Shi, T.; Gong, J.; Hu, J.; Zhi, X.; Zhu, G.; Yuan, B.; Sun, Y.; Zhang, W. Adaptive feature fusion with attention-guided small target detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–16. [\[CrossRef\]](#)
24. Chen, J.; He, H.; Zhuo, W.; Wen, S. Run, don't walk: Chasing higher FLOPS for faster neural networks. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
25. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [\[CrossRef\]](#)
26. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051.
27. Zhu, J.; Zhong, J.; Ma, T.; Huang, X.; Zhang, W.; Zhou, Y. Pavement distress detection using convolutional neural networks with images captured via UAV. *Autom. Constr.* **2022**, *133*, 103991. [\[CrossRef\]](#)
28. Adarsh, P.; Rath, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS) IEEE, Coimbatore, India, 6–7 March 2020; pp. 687–694.
29. Jocher, G.; Stoken, A.; Borovec, J.; Changyu, L.; Hogan, A.; Diaconu, L.; Poznanski, J.; Yu, L.; Rai, P.; Ferriday, R. ultralytics/yolov5: V3.0. *Zenodo* **2020**.
30. Li, C.; Li, L.; Jiang, H.; Weng, K. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.

31. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
32. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
33. Wang, C.-Y.; Yeh, I.-H.; Mark Liao, H.-Y. Yolov9: Learning what you want to learn using programmable gradient information. In Proceedings of the Computer Vision—ECCV 2024, Milan, Italy, 29 September–4 October 2024; pp. 1–21.
34. Wang, A.; Chen, H.; Liu, L.; Chen, H.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.
35. Khanam, R.; Hussain, M. Yolov11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725.
36. Arya, D.; Maeda, H.; Ghosh, S.K.; Toshniwal, D.; Sekimoto, Y. RDD2022: A multi-national image dataset for automatic road damage detection. *Geosci. Data J.* **2024**, *11*, 846–862. [[CrossRef](#)]
37. Yu, J.; Jiang, J.; Fichera, S.; Paoletti, P.; Layzell, L.; Mehta, D.; Luo, S. Road Surface Defect Detection—From Image-Based to Non-Image-Based: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 10581–10603. [[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.