Received: 13 May 2024 | Accepted: 11 December 2024

DOI: 10.1111/mice.13409

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Automatic steel girder inspection system for high-speed railway bridge using hybrid learning framework

Tao Xu[1] | **Yunpeng Wu**[1] | **Yong Qin**[2] | **Sihui Long**[1] | **Zhen Yang**[1] | **Fengxiang Guo**[1]

[1]Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming, China

[2]State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing, China

Correspondence
Yunpeng Wu, Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming 650031, China.
Email: wuyunpeng@bjtu.edu.cn

Yong Qin, State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing 100044, China.
Email: yqin@bjtu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 52362048; Yunnan Fundamental Research Projects, Grant/Award Numbers: 202301BE070001-042, 202401AT070409

Abstract

The steel girder of high-speed railway bridges requires regular inspections to ensure bridge stability and provide a safe environment for railway operations. Unmanned aerial vehicle (UAV)-based inspection has great potential to become an efficient solution by offering superior aerial perspectives and mitigating safety concerns. Unfortunately, classic convolutional neural network (CNN) models suffer from limited detection accuracy or redundant model parameters, and existing CNN-based bridge inspection systems are only designed for a single visual task (e.g., bolt detection or rust parsing only). This paper develops a novel bi-task girder inspection network (i.e., BGInet) to recognize different types of surface defects on girder from UAV imagery. First, the network assembles an advanced detection branch that integrates the sparse attention module, extended efficient linear aggregation network, and RepConv to solve the small object with scarce samples and complete efficient bolt defect identification. Then, an innovative U-shape saliency parsing branch is integrated into this system to supplement the detection branch and parse the rust regions. Smoothly, a pixel-to-real-world mapping model utilizing critical UAV flight parameters is also developed and assembled to measure rust areas. Finally, extensive experiments conducted on the UAV-based bridge girder dataset show our method achieves better detection accuracy over the current advanced models yet remains a reasonably high inference speed. The superior performance illustrates the system can effectively turn UAV imagery into useful information.

1 | INTRODUCTION

High-speed steel girder bridge, as an essential rail infrastructure, is a structure constructed for the exclusive purpose of carrying railway traffic across obstructions. Unfortunately, the long-term effects of vehicle–track vibrations and rainwater corrosion would progressively damage the girder structures, causing issues such as surface rust, rusty bolts, and even missing bolts (see Figure 1; Mu et al., 2023), which pose significant risks to railway operation safety. Therefore, accurate and timely inspection for bridge steel trusses is of great significance to maintain bridge structure safety and ensure rail operation safety. Currently, the detection of railway bridge girder defects heavily relies on manual inspection, resulting in low efficiency and potential safety issues for inspectors in remote areas.

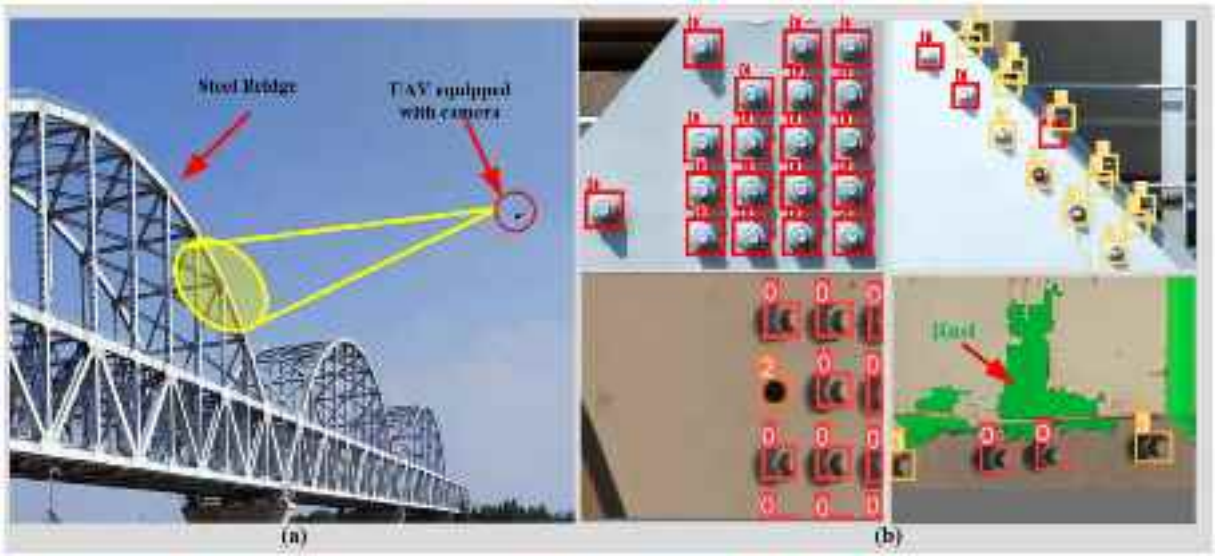

FIGURE 1 Unmanned aerial vehicle (UAV) photograph-based railway bridge girder defect detection and measurement. Note 0, 2, and 3 indicate normal nuts, lost bolts, and rusty bolts, respectively.

Especially in mountainous regions, visual inspections are challenging due to limitations in visibility and terrain, making it difficult to meet the inspection requirements of comprehensive railway lines. A promising solution to address these challenges is the utilization of unmanned aerial vehicles (UAVs) for aerial photography. UAVs with their expansive field of vision, high flexibility, low cost, and low carbon emissions present a potential solution to meet detection needs (see Figure 1). However, despite the advantages of UAV-based aerial photography, visual inspection of railway bridge girders using UAV images still suffers from subjectivity, time consumption, and labor intensity. This is because the images still need to be manually processed by inspectors and there exists a gap between the environmental images captured by the UAV on-site and the effective identification of bridge girder defects by inspectors from UAV images. Therefore, the development of a timely and accurate detection framework for high-speed railway bridge girder surface defects is crucial for improving the safety of railway operations.

In recent decades, defect detection for high-speed railway bridge steel trusses has undergone several stages of development: (1) Manual inspection: The inspection refers to visually observing steel trusses by human eyes to identify defects. Manual inspection is a labor-intensive and time-consuming method, heavily reliant on the experience of trained inspectors (Sacks et al., 2018). (2) Traditional image processing (IP): The IP techniques mainly involve feature extraction, object parsing, and defect classification. Typical algorithms such as scale-invariant feature transform and multiple feature fusion (Y. Xu et al., 2023) are commonly used to extract image features. Support vector machine or criteria based on feature differences are then employed for defect classification (Lim et al., 2021). Although these handcrafted feature designs can yield satisfactory results with limited samples, they often lack robustness when handling images containing complex backgrounds. (**3**) Convolutional neural networks (CNNs): The CNN technologies enable direct processing of complex and discriminative features extracted from raw images for end-to-end model training. Therefore, CNNs have become a focal point of current research on defect detection in railway bridge steel trusses (Z. Li et al., 2023).

Specifically, traditional IP methods could achieve satisfactory results with limited samples for the images containing prominent prospects, balanced lighting, and uniform background; however, they often fail to deal with the object recognition task under the complicated scenes (e.g., messy backgrounds, blurred foregrounds, and irregular point interference). Also, the hand-crafted feature classifiers are usually inefficient, and consequently those IP technologies are hard to process the complex drone imagery captured under aerial scene. Compared to the IP methods, CNNs based on extensive sample learning can automatically learn deep features from data, reduce dependence on manual feature design, and achieve a high detection rate and good robustness in complex environments, thus attracting widespread attention (Meng et al., 2024).

Currently, CNNs have become a mainstream method to complete automatic infrastructure inspection and assessment tasks in civil engineering applications, such as structural engineering damage detection (Gao et al., 2023; Pan et al., 2023), structure modal analysis (Pezeshki, Adeli,

et al., 2023; Pezeshki, Pavlou, et al., 2023, 2004). Classic examples of traffic infrastructure detection include road crack recognition (Yamaguchi & Mizutani, 2024; J. Zhang et al., 2024), arbitrary-orientation construction site object detection (Y. Guo et al., 2023), and concrete compressive strength estimation (Rafiei et al., 2017). Typical bridge inspection examples include bridge crack detection (Li et al., 2023; Vivekananthan et al., 2023) and structural deformation monitoring (Jia et al., 2024; Nettis et al., 2023). These CNN-based methods obtained satisfactory results, but they are not tailored for the railway girder bridge. Recently, several CNNs also exhibit remarkable effects on health monitoring of girder bridge, such as girder bolt detection (Mu et al., 2023; Pan & Yang, 2024) and truss area segmentation (Lamas et al., 2023, 2024). Nevertheless, the above works cannot simultaneously complete structural object detection and pixel-level semantic parsing in a single pass. An all-round bi-task inspection model for bridge girder structure is currently not available for field practices. Specifically, CNN-based bridge truss structure defect detection and evaluation face the following challenges:

Small but clustered object, and small sample categories: These densely arranged bolts on the bridge steel truss are usually small object. The missing bolts are approximately 60 × 60 pixels, while the rusty areas of corroded bolt account for only 10 × 10 pixels or even smaller (Figure 1). Additionally, the limited sample size has always been a challenge for railway infrastructure monitoring based on CNNs. Especially, the missing bolt samples are very rare and semantically sparse, which causes insufficient feature learning for the CNN model, making it difficult to achieve accurate prediction for this category.

Limited but redundant semantic feature: UAV-based steel truss images have limited semantic features such as edges, colors, and textures, and the limited semanteme is also very redundant (e.g., identical colors, consistent bolts, similar structures, etc.). The limited but redundant semanteme produces numerous duplicate gradients when the model utilizes ResNet and deeper network structure to learn deep features in truss imagery, thereby causing ineffective model learning, convergence, and optimization.

Rust with arbitrary size and appearance: Rust regions on the steel truss surface, as shown in Figure 1, exhibit a variety of shapes and a large variation in size. Hence, using the conventional CNN model, which solely learns object location from a global structural perspective, is hard to accurately localize the rust areas by the predicted bounding boxes. Clearly, accurately localizing rust regions is the most critical step in terms of rust area evaluation. But those classic image segmentation models U-Net and Segformer may not achieve satisfactory results either (see Section 4.3). Therefore, this study innovatively integrates a simple U-shaped pixel-level parsing branch into the proposed hybrid learning framework.

To solve the limitations outlined above, this paper presents an innovative UAV photograph-based detection and measurement system for railway bridge girder structure defects, which comprises a novel multi-task fully convolutional framework called bi-task girder inspection network (BGInet) and a mapping model named pixel-to-real-world. The primary contributions of this study can be summarized as follows:

1. This article introduces a pioneering UAV-based inspection system specifically designed for railway bridge trusses using the BGInet framework. To our knowledge, this represents the first effort to integrate a hybrid learning framework with UAV imagery for the detection and quantification of surface defects on railway bridge girders.
2. A detection branch incorporating the sparse attention module (SAM) has been integrated into BGInet, significantly enhancing feature learning and convergence, specifically designed to address the challenges of small object detection and limited sample size in bridge inspections. Additional advanced modules (expanded efficient linear aggregation network [ELAN] and RepConv) are also combined within the detection branch to ensure efficient identification of bolt defects.
3. A simple yet effective U-shaped rust parsing branch is assembled into the BGInet framework, which is professionally suitable for girder rust of irregular shapes and different sizes. The efficiency is also demonstrated on a customized UAV dataset, exceeding other state of the art (SOTA) models in accuracy and inference speed, yet remains efficient and convenient.
4. A pixel-to-real-world mapping model is developed to accurately calculate the true area of rust using the UAV's field of view (FOV), offering a robust method for converting pixel-level predictions into actionable insights for bridge maintenance.

The remainder of this paper is organized as follows: Section 2 introduces related work. Section 3 elaborates on the architecture of the model. Section 4 describes relevant experimental results and analysis, and finally the paper concludes with a summary of the research findings and superiority.

2 | RELATED WORK

Recently, many CNN-based attempts have been devoted into applications for infrastructure in civil engineering (X. Jiang & Adeli, 2005; Rafiei & Adeli, 2017; Yin et al., 2023). In particular, CNNs have made a significant contribution to bridge inspection. For example, Zhang et al. (2024) proposed a three-level classification CNN for concrete bridge structure detection, which completes the image classification of the 100 bridge targets but could not achieve defect localization. Y. Jiang et al. (2024) developed a weakly supervised model to detect and localize bridge structure defects yet obtains low detection rate for spot and crack. Regarding concrete bridge cracks, Y. Liu et al. (2023) and J. Zhang et al. (2022) designed crack detection networks with an improved You Only Look Once (YOLO) structure, indicating superior performance; however, it is difficult to identify small cracks. To solve the issue, Ma et al. (2024) and Qi et al. (2024) proposed new CNN frameworks equipped with Transformer to recognize the small cracks, greatly improving the detection accuracy. Nevertheless, it can only detect structural defects but cannot quantitatively measure them. Zheng et al. (2022) proposed a CNN using Convolutional Active Learning Identification-Segmentation-Measurement (CAL-ISM) Framework for crack segmentation, and then developed a pixel-to-size mapping model to measure crack width, achieving satisfactory results. Besides, there are some bridge health monitoring systems using camera networking (Yin et al., 2023), 3D reconstruction (H. Li et al., 2024; F. Wang et al., 2024; Yamane et al., 2023, 2024), multimodal network (Kunlamai et al., 2024), and other sophisticated sensors (e.g., pressure transducer—Alaie & Al'Aref, 2023; and point clouds—Matono & Nishio, 2024). Nevertheless, they are only tailed for concrete bridge inspection but not steel girder bridge. Bridge girder defects are more prone to surface corrosion and rust and lost bolt, as well as rusty bolts (Mu et al., 2023).

Up to now, advanced CNNs have also been applied to the inspection of steel girder bridge. For instance, Katsamenis et al. (2022) proposed SPLAC U-Net, a three-level architecture for girder rust segmentation and corrosion level assessment, producing satisfactory results in complex environments. J. Xu et al. (2020) developed an Ensembled CNN for the surface rust evaluation in steel girder structures, which achieves over 90% of intersection over union (IoU) on rust segmentation and showed robustness and resistance to image blur. Although they achieve highly accurate girder rust recognition, these saliency detection models focus on pixel-level rust segmentation rather than object (bolt) detection. In terms of UAV-based girder bridge inspection, Mu et al. (2023) proposed an adaptive clipping shallow attention network that leverages an adaptive UAV imagery cropping strategy and a shallow attention network to address the small target and limited sample. L. -Z. Li et al. (2022) developed a new framework based on CNN and long short-term memory to detect the loss of bolts and nuts in bridge steel girder. The average detection rate reached over 90% of mean average precision (mAP), indicating that this method has high effectiveness in bridge health monitoring, but it can only detect bolts and cannot provide comprehensive structure assessment. In general, the existing CNN-based approaches are designed for either object detection or segmentation of a specific task (rust or bolt) and a versatile model for multiclass bridge girder inspections is currently not available for field practices.

3 | METHODOLOGY

Recently, the YOLO families and You Only Look At Coefficients (YOLACT) have shown outstanding performance in object detection or image segmentation. Inspired by these pioneering works, this study develops a UAV-based inspection and measurement system for railway bridge girder, which comprises a multi-task fully network BGInet for bolt defect recognition and rust area parsing, and a pixel-to-real-world mapping model for rust measurement. This section elaborates the BGInet and pixel-to-real-world mapping model algorithm.

3.1 | System overview

This system is designed for detecting and evaluating defects on bridge steel trusses based on UAV images. It mainly consists of two primary components: the BGInet network for defect detection and segmentation and the pixel-to-real-world model for area evaluation based on segmented results. The BGInet network, inspired by the YOLO series, is enhanced with a U-shaped parsing branch for rust segmentation and modules such as Sparse Attention and Extended ELAN (E-ELAN) to improve performance on complex targets. The pixel-to-real-world model complements the detection process by mapping segmented results to real-world measurements through UAV calibration parameters. The overall system structure is illustrated in Figure 2.

3.1.1 | Data preparation

In on-site experiments, a UAV captures images of bridge steel trusses from 10 to 15 m away, cruising at speeds of 2 to 15 m/s. The dataset is annotated with LabelImg and LabelMe tools. The large number and dense arrangement of the truss bolts in an image pose great difficulties for the labeling work; thus, this study also developed a

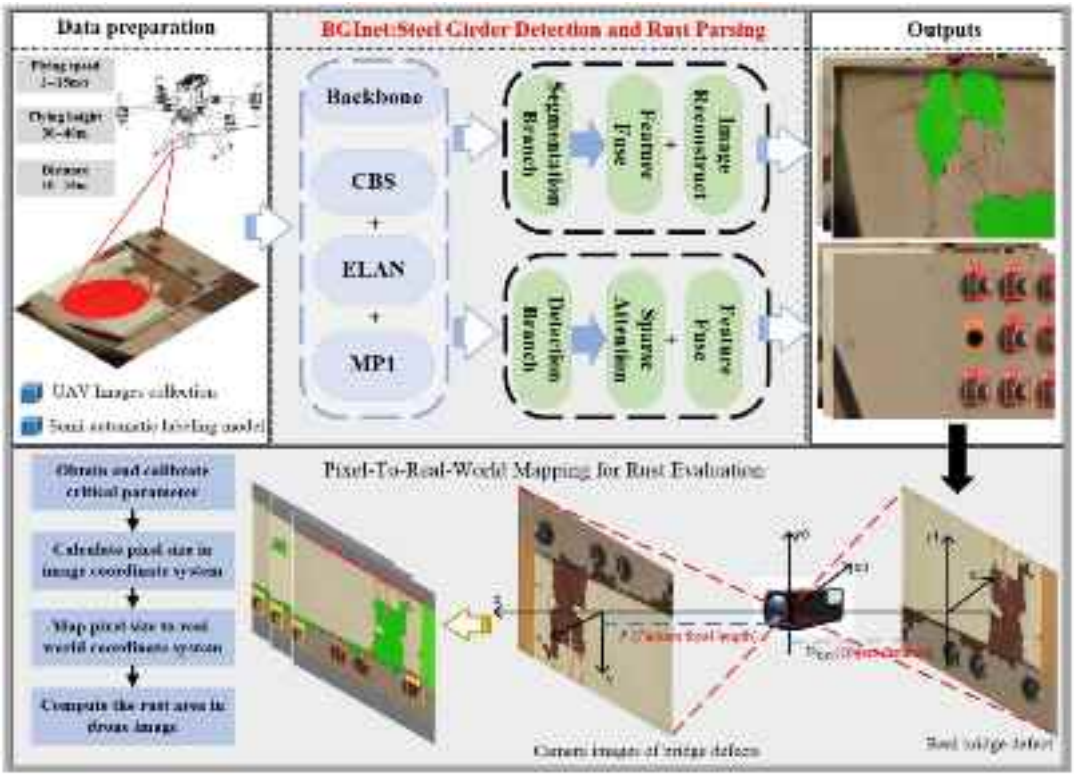

FIGURE 2 Overview of UAV-based inspection and measurement system for railway bridge girder. CBS, Conv, BN, SilU; ELAN, Efficient Layer Aggregation Network; MP1, max pooling 1.

semi-automatic annotation model to assist the manual labeling and reduce the cumbersome annotation process. During the semi-automated annotation model training, 500 truss images were manually annotated for initial training. The preliminary model's outputs on unannotated images were then fine-tuned manually. This semi-automated approach with progressive fine-tuning significantly reduces the workload of dataset annotation.

3.1.2 | Model training

Four detection object categories: normal nuts, normal bolts, missing bolts, rusty bolts, and a pixel-level parsing object: surface rust are selected for BGInet model training. The dataset is established with 2000 UAV images for training, testing, and validation purposes. The training process comprised a total of 300 epochs. The initial learning rate is 0.015, which is adjusted by a decay factor of 10 after 230 iterations. The dataset was randomly partitioned into training, test, and validation sets at an 8:1:1 ratio.

3.1.3 | UAV calibration

Some UAV flight parameters can be obtained from the operation manual, while the FOV needs to be evaluated by experimental calibration. Thereby a calibration experiment is executed to calculate FOV, and the critical parameters are subsequently fed into the pixel-to-real-world mapping model for rust area measurement.

3.1.4 | Performance evaluation

The BGInet model performance is evaluated and compared across different SOTA models based on precision-recall (PR) curves, average precision (AP), IoU, and frames per second (FPS) (Wu et al., 2023). Besides, to verify the lightweight BGInet (BGInet_l) performance, an edge deployment experiment was conducted on an NVIDIA Jetson Xavier NX. Finally, visualization and computation experiments of the total system are performed to validate the system's effectiveness.

3.2 | BGInet network

The proposed BGInet network integrates a novel detection branch incorporating SAMs, E-ELAN, RepConv, and the other advanced modules, along with an innovative U-shaped saliency segmentation branch, into a single framework, as illustrated in Figure 3. Compared to the

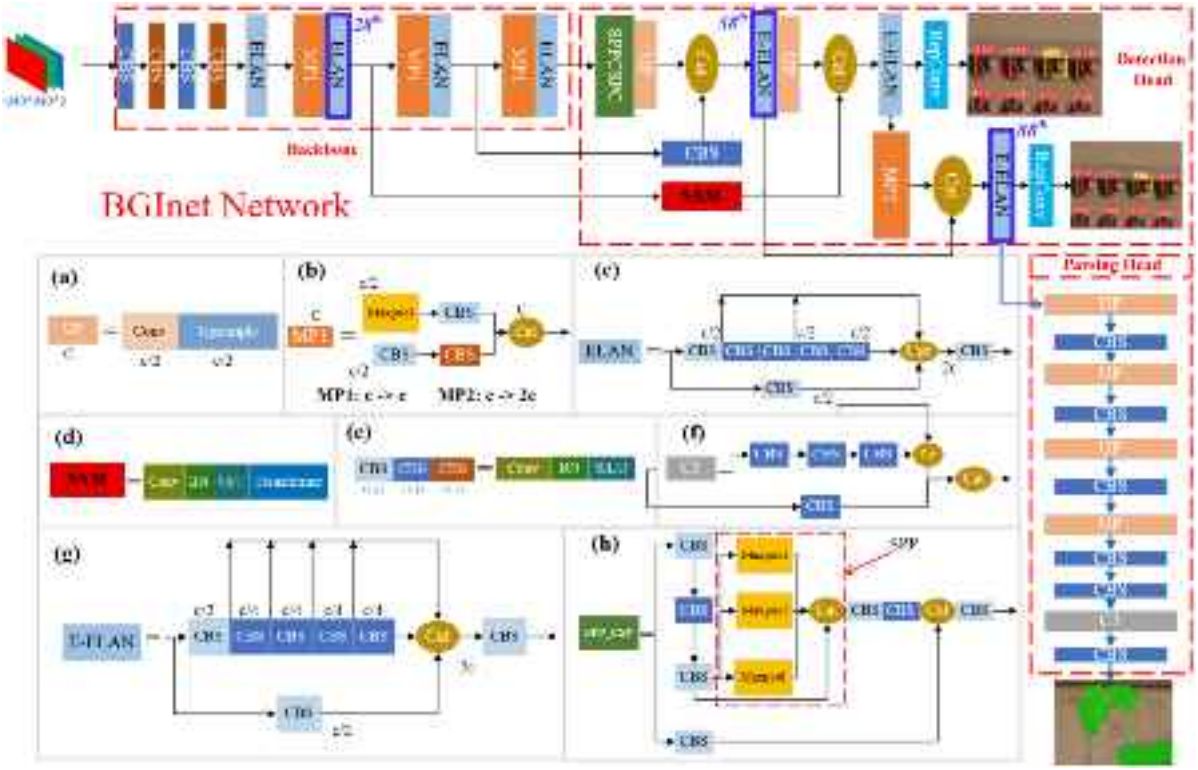

FIGURE 3 Bi-task girder inspection network (BGInet) architecture. (a) UpSampling (UP) block. (b) Max Pooling (MP) block. (c) Efficient Layer Aggregation Network (ELAN) block. (d) sparse attention module (SAM) block. (e) Conv, Batch Normalization (BN), Sigmoid Linear Unit (SilU), (CBS) block. (f) C3 block. (g) Extended ELAN (E-ELAN) block. (h) SPP_CSP block. Note that BGInet integrates both parsing and detection branches, and its blue border areas indicate the position where the segmentation heads are drawn starting from the 28^{th}, 58^{th} and 88^{th}, respectively. The SPP_CSP block is composed of the cross-stage partial (CSP) block and the spatial pyramid pooling (SPP) block, with the objective of achieving diverse receptive fields through maximum pooling to employing to images with varying resolutions.

YOLO series and other bridge girder detection models (R. Li et al., 2023; Z. Li et al., 2023), BGInet is specifically designed for inspecting railway bridge girder using drone images, which can simultaneously complete: (1) precise bolt defect detection and (2) efficient pixel-level rust area parsing.

3.3 | Backbone

The configuration of the backbone is summarized in Table 1, which consists of three modules: the Conv, BN, SilU (CBS), ELAN (C.-Y. Wang et al., 2023), and max pooling 1 (MP1; Murray & Perronnin, 2014).

3.3.1 | ELAN

The ELAN can achieve effective learning and convergence of a deeper network by controlling the shortest and longest gradient paths and enhance the model's robustness (C.-Y. Wang et al., 2022). As shown in Figure 3c, ELAN divides the inputs into two paths: one path passes through a 1 × 1 Conv layer, while the other path passes through a 1 × 1 Conv layer followed by four 3 × 1 Conv layers. Finally, before the output, concatenate the results of the first path with the results of the first and third convolutions of the second path, as well as the output of the second path.

3.3.2 | MP1

MP module can reduce computation, prevent overfitting, and increase the receptive field to enable subsequent convolutional layers to learn more global information, thereby BGInet introduces the MP to enhance network learning capability. The MP1 operator, as sketched in Figure 3b, splits the inputs into two paths: The first branch undergoes downsampling through a max-pooling operation, followed by a 1 × 1 Conv layer to change the number of channels, while the other path passes through a 1 × 1 Conv layer for channel adjustment, then through a 3 × 3 convolutional layer with a stride of 2, which serves the downsampling operator. Finally, the two information flows are concatenated as output.

TABLE 1 Backbone network configuration.

Layer	Filter size	Stride	Output channels	Output size
CBS×4	[3×3] ×4 [1×1] ×2	[1,2,1,2]	[16,16,16,64]	[640,320,320,160]
ELAN	[3×3] ×4 [1×1] [1×1] ×2	[1,1,1,1,1,1,1]	[256,256,128,256,128,128,256]	160
MP1	[3×3] ×4 [1×1] [1×1] ×2	[1,1,2]	[64,64,64]	80
ELAN	[3×3] ×4 [1×1] [1×1] ×2	[1,1,1,1,1,1,1]	[64,64,64,64,64,64,256]	80 40
MP1	[3×3] ×4 [1×1] [1×1] ×2	[1,1,2]	[128,128,128]	
ELAN	[3×3] ×4 [1×1] [1×1] ×2	[1,1,1,1,1,1,1]	[128,128,128,128,128,128,512]	40
MP1	[3×3] ×4 [1×1] [1×1] ×2	[1,1,2]	[256,256,256]	20
ELAN	[3×3] ×4 [1×1]	[1,1,1,1,1,1,1]	[128,128,128,128,128,128,512]	20

Abbreviations: CBS, Conv, BN, SilU; ELAN, Efficient Layer Aggregation Network; MP1, max pooling 1.

3.4 | Detection branch

The detection branch of BGInet in Figure 3 mainly includes the E-ELAN, SAM, and spatial pyramid pooling and cross-stage partial (SPP_CSP) modules, which only utilize two scale detection heads to match these small and similar size categories: normal nut, normal bolt, rusty bolt, and missing bolt. This also reduces computation overhead. In the training of BGInet, the inputs are firstly resized to 640 × 640 and then is fed into the model. Consequently, the branch outputs two sets of scale prediction grids (80 × 80, 40 × 40) by upsamples to match the different size objects.

3.4.1 | E-ELAN

Since the E-ELAN can leverage larger channel numbers and incorporates expanded, shuffled, and merged cardinalities to continuously enhance the network's learning capacity without disrupting the original gradient paths (C.-Y. Wang et al., 2023), compared to ELAN, we introduce E-ELAN in the detection branch to process deep object features. As sketched in Figure 3g, E-ELAN splits the input into two paths: one path goes through a 1 × 1 Conv layer, while the other path goes through a 1 × 1 Conv layer followed by four 3 × 1 Conv layers. And then, the two groups of outputs are concatenated, followed by a cascade operator to facilitate information exchange between shallow and deep layers. The E-ELAN module is connected to other modules through the connection layer, which promotes better gradient flow during backpropagation and solves problems such as gradient vanishing in deep architectures. This integration enhances the training stability and convergence speed of the model, thereby improving performance indicators.

3.4.2 | SAM

In research, accurate detection of missing bolts is a very challenging task due to small sample sizes and small objects. The classic Transformer attention mechanism has great potential to solve such problems by capturing long-distance dependencies. However, it often requires multiple attention calculations, which significantly increases computational complexity and memory requirements. In order to solve the problem of huge computing resource

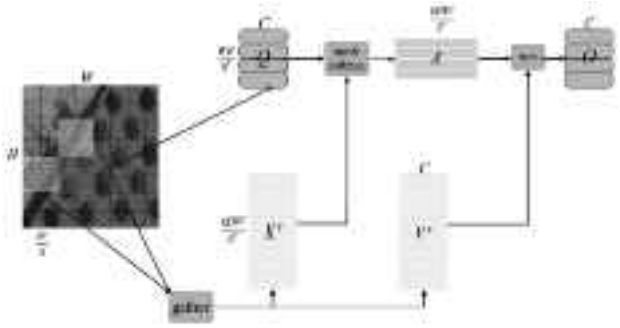

FIGURE 4 Flow chart of sparse token-to-token attention. The sparse attention can be implemented by collecting refined key-value pairs in the *top-n* relevant partitioned regions.

overhead caused by Transformer, some studies (Dong et al., 2022; Z. Liu et al., 2021; Tu et al., 2022; Zhu et al., 2023) proposed different sparse attention mechanisms, in which each query only focuses on a small number of key-value pairs instead of all. However, these existing works either use hand-crafted static patterns or share a sampled subset of key-value pairs across all queries, which may not cover all input data diversity, causing some key information to be ignored or lost. Inspired by these pioneer works, this study designed an SAM to construct the detection branch, as shown in Figure 4. The main idea of the sparse attention is intended to filter out the majority of irrelevant key-value pairs at a coarse region level while retaining refined routed regions. Then, the fine-grained token-to-token attention is employed within the union of remaining candidate regions. SAM effectively focuses on relevant features by assigning higher attention weights to key areas, such as bolt defects, and filters out irrelevant or redundant regions, such as background noise or non-defective areas. Thereby, it improves the model's ability to generalize, particularly in small sample datasets, by concentrating computational resources on informative regions and avoiding distractions from less relevant information. This contributes to better performance in the BGInet framework. This enables the model to perform promisingly even when data semantics are limited. The procedure of the sparse attention is summarized as pseudocode in Algorithm 1. Detailed explanations are provided as follows:

1. Region partition and corpus construction: Assume a feature map $X \in R^{H\times W\times C}$, we first divide it into s^2 non-overlapping regions, thereby each region contains HW/S^2 feature vectors. This step is completed by reshaping X into $X^r \in R^{S^2\times HW/S^2\times C}$, then the query, key, value tensor, $Q, K, V \in R^{S^2\times HW/S^2\times C}$ can be derived with linear projections:

$$Q = X^r W^q, K = X^r W^k, V = X^r W^v \tag{1}$$

ALGORITHM 1 Pseudocode of sparse attention

```
# In, Out: feature map (H, W, C).
# S^2: the quantity of partitioned regions
# n: the quantity of partitioned regions to attend
# projection: (H, W, C) to (S^2, HW/S^2, C)
x = projection (in, chunk_size = H//S)
# linear projection of Q, K, V
Q, K, V = lin_pro_qkv (x). bulk (3, dim = -1)
# region class Q and K
Q^r, K^r = Q_means(dim = 1), K_means (dim = 1)
# region-to-region adjacency matrix
A^r = mm (Q^r, K^r_Trans (−1, −2))
# calculates the index matrix of the routing area
I^r = topn (A^r, n).index
# gather K and V tensor
K^g = gather (K, I^r), V^g = gather (V, I^r)
# token-to-token attention
Out = Attention (Q, K^g_Trans(−1, −2), V^g)
# restore to original size
Out = unprojection (Out, chunk_size = H//S)
```

where $W^q, W^k, W^v \in R^{C\times C}$ are projection weights for the query, key, value, respectively.

2. Region-to-region query and key refinement: We first get the per-region queries and keys, $Q^r, K^r \in R^{S^2\times C}$ by employing region-level average operator on Q and K, separately. Then the relation adjacency matrix, $A^r \in R^{S^2\times S^2}$ of the region-to-region affinity graph can be obtained by matrix multiplication between Q^r and transposed K^r:

$$A^r = Q^r (K^r)^T \tag{2}$$

Each element in the adjacency matrix A^r represents the degree to which two regions are semantically related. The next crucial step involves pruning the affinity graph by retaining only the *top-n* connections for each region. Specifically, we derive a routing index matrix, denoted as $I^r \in N^{S^2\times n}$ by using the row-wise *top-n* operator:

$$I^r = Top\,(n)\,Index\,(A^r) \tag{3}$$

Hence, row ith of I^r contains n indexes of the most relevant region of the ith region.

3. Sparse token-to-token attention: Finally, we can apply the routing index matrix I^r, to achieve the sparse and fine-grained token-to-token attention. Regarding each query tensor Q of region i, Q should consider to all key-value (k, v) pairs in the union of n routed areas indexed by $I^r_{(i,1)}, I^r_{(i,2)}, \ldots, I^r_{(i,n)}$ to find the relevance between regions. Note that the refined key-value pairs in the *top-n* routed regions can be gathered by a gather operation.

As a result, the sparse attention can be implemented by:

$$O = Attention\left(Q, gather\left(K, I^r\right), gather\left(V, I^r\right)\right) \quad (4)$$

where K^g, $V^g \in R^{S^2 \times \frac{HW}{s^2} \times C}$ are gathered key and value tensor.

Additionally, this study also deploys the SPP_CSP block, RepConv block (Ding et al., 2021) and MP2 (Murray & Perronnin, 2014) at the detection branch. The SPP_CSP block, as the entrance of the detection branch is employed to optimize the outputs from the backbone, which divides the output feature into two parts by CSP (C.-Y. Wang et al., 2020): one part undergoes conventional processing, while the other part passes through SPP (He et al., 2015) operator. As sketched in Figure 3, the SPP can enhance target feature learning capability by MP and multi-scale local features fusion. The RepConv effectively combines 3×1 Conv layers, 1×1 Conv layers, and identity connection within a single convolutional block. Using RepConv block can transform the multi-branch network during the training into a single-path structure for efficient and fast prediction at inference time. Besides, the detection branch still employs the MP block to reduce feature loss during feature downsampling.

The tensors of features generated by SAM and E-ELAN are concatenated to create a unified feature map after being aligned through convolutional configurations, but in order to make them interact and fuse semantically, we then process the concatenated features through another round of E-ELAN. This additional refinement ensures that the combined features interact semantically, thereby achieving centralized gradient path expansion, improving feature propagation and fusion, and ultimately enhancing robustness and generalization capability. While SAM does not interact directly with SPP_CSP, both components contribute to overall model performance by extracting and processing features at different stages. Together, the integration of SAM with E-ELAN and the standalone SPP_CSP results in a more efficient model architecture, enhancing the detection accuracy of small, complex, and sample-scarce defects while improving overall robustness.

3.5 | Parsing branch

In light of our investigation, those traditional object detection models can hardly precisely predict bounding boxes to localize the rusts due to its following characteristics: (1) limited texture and color features and (2) various sizes, shapes, and boundaries. The per-pixel rust parsing by saliency segmentation could be an attractive solution based on the pioneer study AOYOLO (Wu et al., 2023); thus, this study integrated an innovative U-shaped parsing branch into BGInet, as sketched in Figure 3, to achieve rust segmentation.

The U-shaped structure is an effective network design designed to handle semantic segmentation tasks. Its characteristic is that it can maintain important contextual information while restoring the spatial resolution of the image by combining downsampling and upsampling. Specifically, the encoder part of the U-shaped structure is responsible for extracting features and reducing the image size, while the decoder part restores the feature map to its original size by upsampling. This design enables the model to capture both detailed information and global features, thereby improving segmentation accuracy. As shown in Figure 3, in BGInet architecture, the pixel-level parsing branch is connected to the last E-ELAN outputs, the feature map dimension of which is (W/16, H/16, 256). We only use CBS and C3 as the base units for the branch. From Figure 3, to construct a U-shape parsing structure, we use four upsamples to restore these original feature maps from the detection branch to (W, H, 2) that represent the probability of each pixel for the rust and background prediction. The U-shape structure can construct information flows from shallow layers to deep layers, enabling the model to fully learn object location features in high-level layers and edge semantics in shallow layers (Wu et al., 2023). In addition, edge computing plays an important role in the application of the U-shaped structure. The concise U-shaped parsing branch adopted in BGInet can quickly achieve quantitative segmentation of rust with only a small number of parameters, thereby meeting the performance requirements of the limited onboard hardware.

Figure 5 depicts the output feature visualization of the newly developed parsing branch derived from different depth layers in BGInet. From Figure 5b, it can be easily observed that the parsing branch predicts many irregular semantics when it is fed to the 28th layer. As seen in Figure 5c, the output features are relatively pure when connecting the parsing branch to the 58th layer. Nevertheless, when BGInet assembles the parsing branch in the 88th layer of the network, the output features acquire the best visualization results as shown in Figure 5d. One possible reason is that the deeper the network, the more neurons participate in learning, and the better the pixel-level prediction effect. Consequently, we feed the parsing branch to the deepest location of the detection branch in BGInet to build the U-shape architecture. Further experimental details will be elaborated in Section 4.

The proposed BGInet network contains a novel SAM to enhance feature extraction, especially to improve the detection accuracy of small samples. It also reduces the amount of computation and memory usage. It also uses E-ELAN and simplifies the design of the detection head to improve computational efficiency and processing speed,

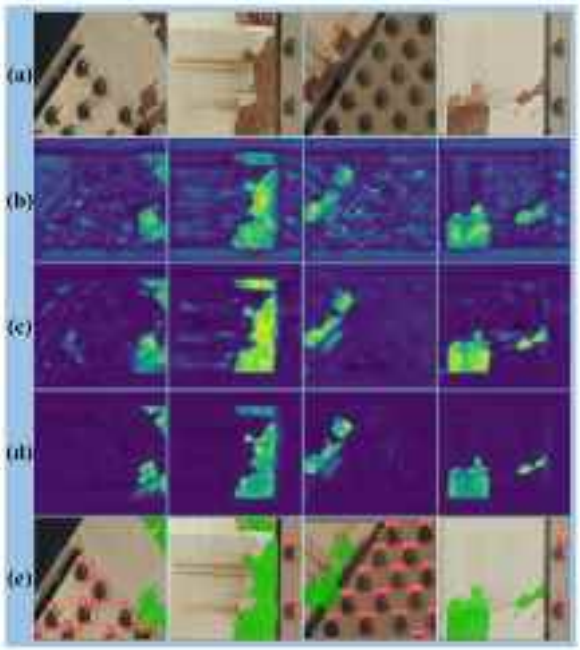

FIGURE 5 The output feature visualization of the parsing branch derived from different depth layers in BGInet. (a) Inputs. (b), (c), and (d) are feature visualizations of the parsing branch connected to the 28th, 58th, 88th layer, respectively. (e) Predicted results.

achieving a balance between detection accuracy and real-time performance.

3.6 | Loss function

Since BGInet incorporates two branches, this study designs a multi-task hybrid loss function in the model training, which consists of two parts: L_{det} for object detection and L_{seg} for rust parsing. The overall loss can be expressed by:

$$Loss = \alpha L_{det} + \beta L_{seg} \tag{5}$$

where α and are β the weighted coefficients of detection loss and parsing loss respectively.

3.6.1 | Object detection loss

The loss term L_{det} is employed to monitor the classification loss L_{cls}, bounding box regression loss L_{box}, and confidence loss L_{obj} between the predicted bounding box and the corresponding ground truth. L_{det} is the weighted sum of these losses, thereby can be expressed as

$$L_{det} = \delta_1 L_{obj} + \delta_2 L_{cls} + \delta_3 L_{box} \tag{6}$$

where L_{box} utilizes the IoU loss to measure the localization accuracy, L_{cls} represents the classification loss, which evaluates how accurately the model predicts the class of the detected object, and L_{box} accounts for the objectness loss, which assesses whether an object exists in the predicted bounding box.

3.6.2 | Rust parsing loss

The loss term L_{seg} uses the classic cross entropy (CE) to regress the rust prediction task, which aims to minimize the classification error at the pixel level between the predicted mask and the corresponding ground truth. Let an N-class dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where x and $y \in \{1, 2, \ldots, N\}$ are input and the corresponding label. For each sample pixel x in a UAV girder image, a deep model predicts its possibility that belongs to each label $n \in \{1, 2, \ldots, N\}$. Rust segmentation loss can be calculated from the formula:

$$P(n|x) = \frac{e^{Z_n}}{\sum_i^N e^{Z_i}} \tag{7}$$

where Z_i represents the logits. Thereby, the CE loss can be given as

$$L_{seg} = l_{CE} = \sum_n^N G(n|x) \log P(n|x), \quad \sum_{n=1}^N G(n|x) = 1 \tag{8}$$

where $G(n|x)$ represents the ground truth distribution of sample x across all labels. It is defined such that $G(n|x) = 0$ for all $n \neq y$, and $G(n|x) = 1$ when the ground truth label of sample n is y. In other words, $G(n|x)$ assigns a value of 1 to the true label y and 0 to all other labels, effectively encoding the true class information for the given sample x.

3.7 | Pixel-to-real-world mapping model

When flying drones to collect bridge girder data along bridge, the camera lens should be perpendicular to the bridge to accurately calculate the corrosion area of the steel truss surface. Also, some UAV key parameters need to be obtained to model rust measurement model. These parameters include the distance D_{UAV} from the drone to the bridge, the focal length f, and the image resolution $W \times H$. These values are predetermined or readily available based on the UAV's You Only Look At Coefficients (LiDAR) ranging module and camera specifications and are used during the pixel-to-real-world mapping process. In this study, we took advantage of the known focal length and distance from the drone to the bridge. By adjusting these

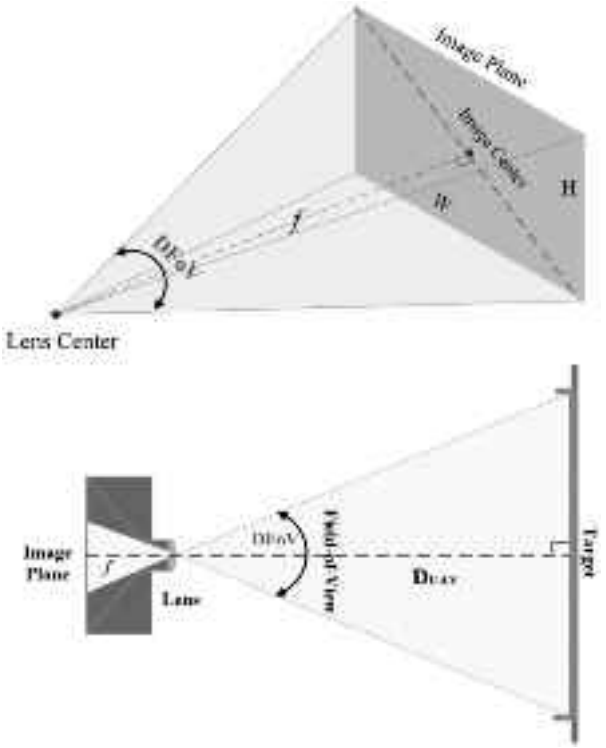

FIGURE 6 Field-of-view (FOV) relationship and lens imaging.

parameters to reflect the actual data collection situation and using the calibration plate to occupy the shooting interface, we calculated the diagonal distance of the camera image. As shown in Figure 6. we employed simple trigonometric calculations to derive the diagonal FOV (D_{FoV}) from the known drone distance and the diagonal distance. Long et al. (2024) propose a novel tunnel-lining defect approach based on a skeleton extraction algorithm that employs pixel-to-real-world mapping to calculate the actual length of tunnel lining void. Inspired by pioneer work, we developed a pixel-to-real-world mapping model for rust regions. The details are as follows:

First, the unit pixel size δ_{pixel} in image coordinate system can be calculated by the geometric relationship between the FOV D_{FoV}, image size (W, H), and focal length f as shown in Figure 6. The calculation formula is as follows:

$$\delta_{pixel} = \frac{2\tan\left(\frac{D_{FoV}}{2}\right) \times f}{\sqrt{W^2 + H^2}} \tag{9}$$

Then, the actual unit pixel size Δ_d in real-world coordinate system can be obtained based on the theory of similar triangles as follows:

$$\Delta_d = \frac{D_{UAV} \times \delta_{pixel}}{f} \tag{10}$$

where δ_{pixel} is unit pixel size in image coordinate system, D_{UAV} denotes the distance from the drone to the bridge, and f is the focal length.

Finally, the area of the unit pixel in real world can be given by $\Delta_s = \Delta_d^2$. Therefore, the actual area of rust on the steel truss of railway bridges can be estimated using the following formula:

$$S_{rust} = \varepsilon_{pixel} \times \Delta_s \tag{11}$$

where ε_{pixel} is the number of the predicted masks.

4 | EXPERIMENTS

In this section, we conduct ablation studies on BGInet (e.g., the sparse attention and parsing branch) to validate the performance of the proposed BGInet using a dataset of bridge steel truss images acquired from UAV. Additionally, substantial comparative experiments with the classical and SOTA networks are conducted on the dataset to prove the superiority of the proposed BGInet. Finally, the model edge deployment experiments are performed based on the NVIDIA Jetson Xavier NX kit to verify the practical deployment capabilities of BGInet.

4.1 | Experiment setup and semi-automatic annotation

The girder images of railway bridges were captured using the Zenmuse H20 aerial camera mounted on the DJI Matrix 300 RTK UAV and DJI Phantom 4. The Zenmuse H20 camera, equipped with a focal length range of 6.83 to 119.94 mm, can capture bridge images at a rate of 30 frames/s with video resolutions of 3840 × 2160 or 1920 × 1080. The drone is maintained at a controlled cruising speed of 2–15 m/s. These images were taken along a railway bridge in Hebei and Shandong, China. The distance between the drone and the bridge is 10–15 m when collecting UAV imagery. Various sample examples of railway steel-bridge based on UAVs are shown in dataset in Figure 7. The dataset includes a total of 2000 images covering various bridge surface diseases, which are collected throughout the year. The specific number of categories is shown in Table 2. We employed online data augmentation (such as random rotation, random flipping, random color change, etc.) during model training. The training process comprised a total of 300 epochs. The initial learning rate is 0.015, which is adjusted by a decay factor of 10 after 230 iterations. The dataset was randomly partitioned into training, test and validation sets at an 8:1:1 ratio. The weighted factors δ_1, δ_2, and δ_3 in detection loss are set as 0.1, 0.5, and 1.0, respectively. The weighted factors α and β in the overall loss are set to 1, respectively.

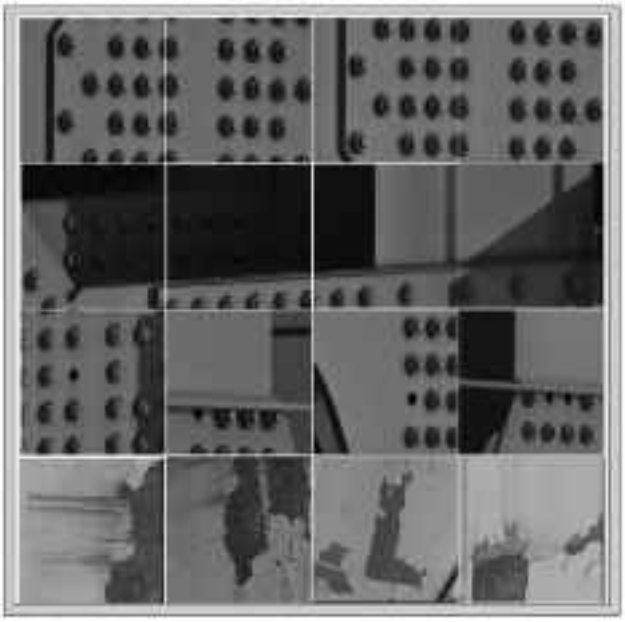

FIGURE 7 Extensive examples of various samples from UAV-based railway steel-bridge dataset.

TABLE 2 The number of each type of label.

Normal nut	Normal bolt	Missing bolt	Rusty bolt	All
12,056	310	124	950	13,440

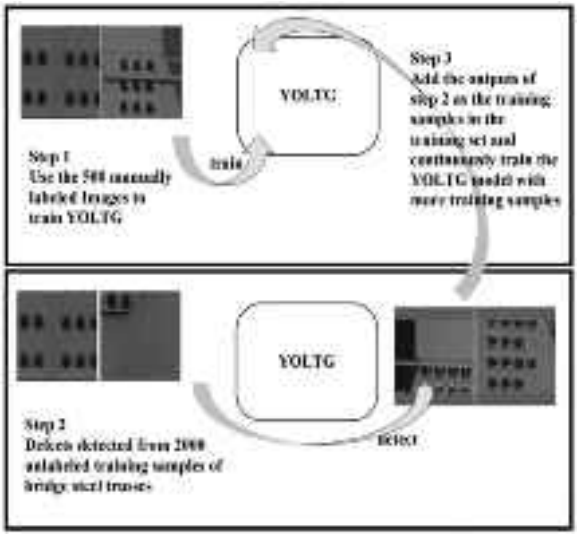

FIGURE 8 Semi-automatic labeling model.

In addition, as sketched in Figure 7, the surface bolt defect is very small and densely ranged. This brings a huge workload to our manual data annotation; thus, this study designs a semi-automatic annotation approach to assist this annotation work. Figure 8 describes the processing of the semi-automatic annotation: (1) First, we manually label only 500 truss images and train the BGInet using the small dataset. (2) The trained model is utilized to output coarse annotation boxes for part of unlabeled images by roughly predicting the bolt location. (3) Then, these coarse annotations are manually adjusted slightly and are fed to the training set to further train and fine the BGInet model. (4) Repeat steps 2 and 3. By implementing the gradual training processing based on the semi-automatic annotation, the workload associated with annotating the dataset can be significantly alleviated.

4.2 | Ablation study on BGInet

4.2.1 | Ablation study on sparse attention

This study employs an Precision-Recall (PR) curve to validate the effectiveness of the sparse attention mechanism (SAM) within the detection branch of the BGInet framework. Figure 9a–e exhibits comparable performance across all classes for BGInet adding SAM or not. As shown in Figure 9e, BGInet equipped with SAM attains an overall mAP of 89.2%, reflecting a 5.8% increase, compared to the original BGInet (89.2% vs. 83.4%). In our experiments, the original BGInet without SAM only gets 71.4% AP for the missing bolt; however, the SAM greatly increases the detection rate of the category from 71.4% to 95.8% (+24.4%), illustrated by the triangle line in Figure 9c. This is because SAM builds region-to-region relationship and captures long-range dependencies by applying fine-grained token-to-token attention in the union of routed regions within an image, thereby enriching the feature space and enhancing the model's representation capability. It is worth noting that SAM also slightly improves the detection rate for the targets with sufficient sample sizes, such as normal and rusty bolts, as seen in Figure 9a,b. Therefore, the proposed sparse attention can substantially elevate BGInet's detection capability for the missing bolts and is indeed necessary and helpful to solve the small sample and limited semantic category issues in bridge gird inspection.

Besides, Table 3 shows the comparison results of BGInet using SAM and traditional attention mechanisms in small sample detection. From Table 3, SAM outperforms Channel Attention (Q. Wang et al., 2020), Spatial Attention (Hu et al., 2018), and Convolutional Block (CB) Attention (Woo et al., 2018) by 20.6%, 8.2%, and 7.9% APs, in bolt missing detection, respectively. SAM also shows mAP improvements of 6.4%, 1.3%, 1.5%. This is attributed to SAM's coarse-grained screening of irrelevant key-value pairs and retaining of refined routed regions. This enables SAM to enhance feature learning, convergence, and detection accuracy for small sample targets.

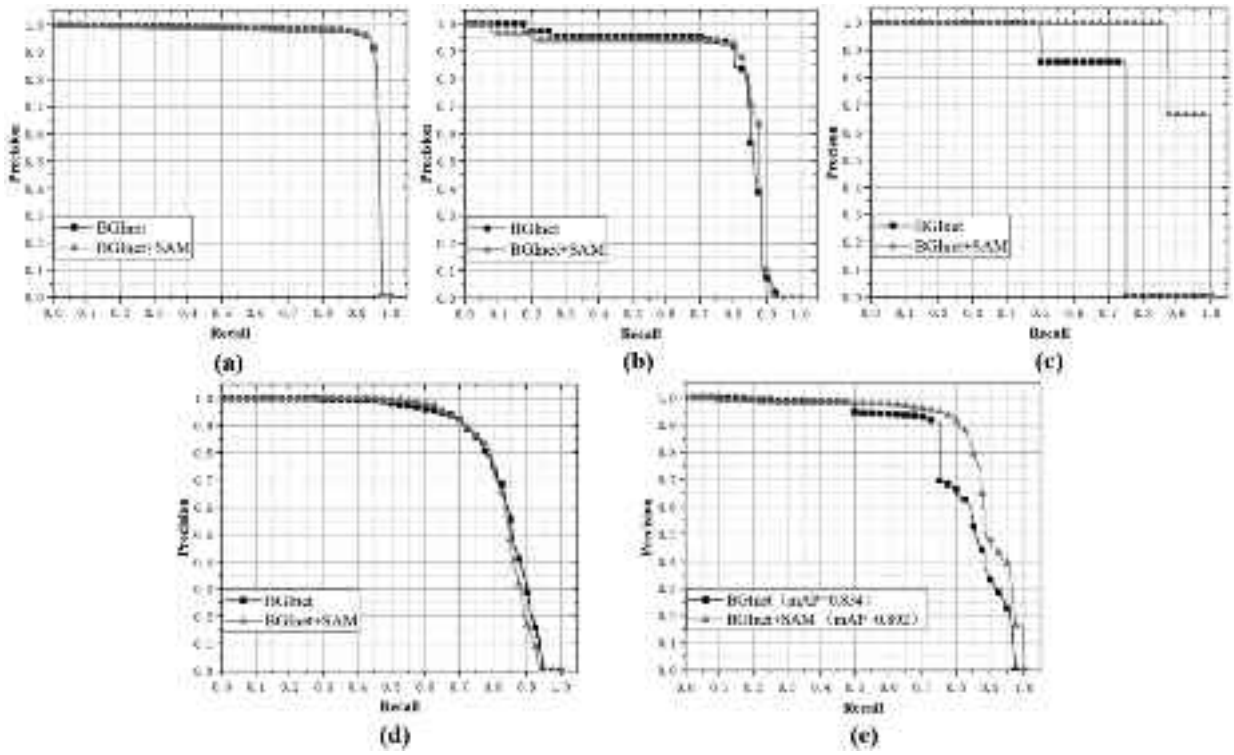

FIGURE 9 Precision-recall (PR) curves of BGInet adding sparse attention or not. (a)–(e) are the PR curves of normal nuts, normal bolts, missing bolts, rusty bolts, and all categories, respectively.

TABLE 3 Ablation experiment results of traditional attention mechanism.

Attention mechanisms	Missing bolt (AP)	Mean average precision (mAP)
Channel Attention	0.748	0.828
Spatial Attention	0.872	0.879
CB Attention	0.875	0.877
SAM	**0.954**	**0.892**

Abbreviations: AP, average precision; SAM, sparse attention module.

4.2.2 | Ablation study on parsing branch structure position

Enlightened by AOYOLO (Wu et al., 2023), this study builds a U-shape rust segmentation branch in BGInet, but which part of the network is better to export this branch from? Is shallow backbone or deep detection branch in the network architecture? Hence, this study executes three groups of comparisons in which the parsing branch is fed to the 28th, 58th, and 88th layer of BGInet, respectively, to confirm the best position. From Figure 10a, BGInet exhibits superior segmentation performance (around 94% IoU) when deriving from the deepest 88th layer, followed by the middle layer 58th layer. The shallow 28th layer in the backbone gets the lowest 89.3% IoU. We speculate that the model could excavate more nuanced features related to rust object in the high-level semantic feature space. And from Figure 10b, the parsing and detection loss from 88th layer get very smooth curves. Given all of that, as sketched in Figure 3, the parsing branch is connected to the deepest layer in BGInet to construct a shallow-to-deep U-shape structure.

4.2.3 | Ablation study on BGInet adding E-ELAN and SPP_CSP or not

We evaluated the impact of the E-ELAN and SPP_CSP modules on key performance indicators, including mAP and IoU, in the BGInet model. As shown in Table 4, the baseline model without these modules achieved an mAP of 0.832 and an IoU of 0.696. Adding the E-ELAN module improved the mAP to 0.877 and IoU to 0.932, demonstrating its significant impact on detection accuracy. Incorporating the SPP_CSP module further enhanced performance, resulting in a mAP of 0.878 and IoU of 0.932. When both modules were included, the model reached its highest mAP of 0.892 and IoU of 0.937, indicating a strong synergistic effect on accuracy and robustness. These results underscore the critical roles of the E-ELAN and SPP_CSP modules in improving bridge bolts detection and rust parsing tasks.

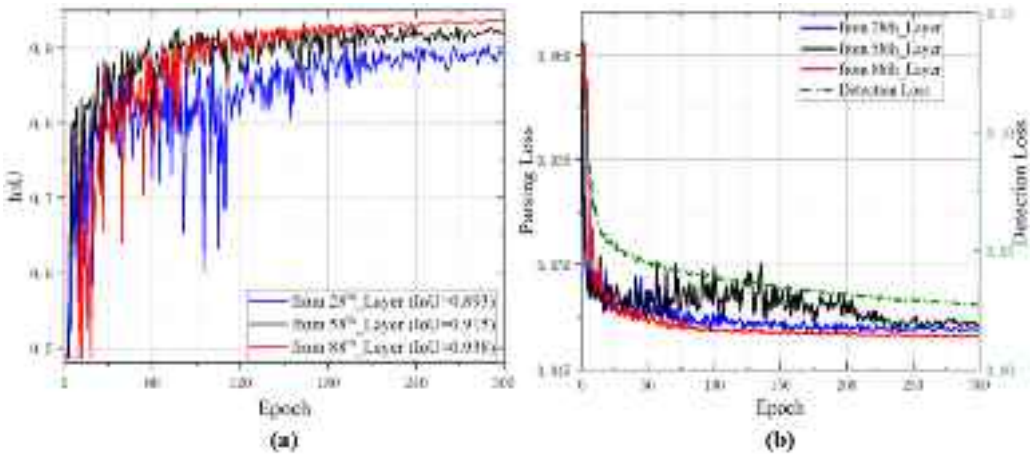

FIGURE 10 Comparison of the parsing branch feed to the 28th, 58th, 88th layer of BGInet. (a) Intersection over union (IoU) curves. (b) Detection and parsing loss curves. Note 28th layer belongs to the shallow backbone, and the 58th and 88th layers are in the deep detection branch.

TABLE 4 Comparison results of BGInet adding E-ELAN and SPP_CSP modules or not.

E-ELAN	SPP_CSP	mAP	IoU
N/A	√	0.878	0.932
√	N/A	0.877	0.932
N/A	N/A	0.832	0.696
√	√	**0.892**	**0.938**

Abbreviations: E-ELAN, extended efficient linear aggregation network; IoU, intersection over union; mAP, mean average precision; SPP_CSP, spatial pyramid pooling and cross-stage partial.

4.3 | Comparison between BGInet and SOTA models

In terms of target detection, the extensive AP comparison and mAP experiments with the classic and SOTA models such as Faster Regions with CNN Features (RCNN) (Ren et al., 2015), Cascade RCNN (Cai & Vasconcelos, 2018), Single Shot Multibox Detector (SSD) (W. Liu et al., 2016), EfficientNet (Tan and Le, 2019), YOLACT (Bolya et al., 2019), YOLOv5 (Jocher et al., 2020), YOLOv7 (C.-Y. Wang et al., 2023), YOLOv8, and YOLOv9 (Wang et al., 2024) are conducted to check the superiority of BGInet. Besides, the IoU comparison experiments with the leading methods, U-Net (Ronneberger et al., 2015), PSPNet (Zhao et al., 2017), Semantic Edge-aware Feature Pyramid Network (SEM-FPN) (Lin et al., 2017), Segformer (Xie et al., 2021), and Segnext (M. -H. Guo et al., 2022) are also carried out to verify BGInet in terms of object segmentation.

4.3.1 | Comparisons on object detection

Table 5 presents AP and mAP comparison results between BGInet and the SOTA models. In normal category detection, as shown in Table 5, all the models acquire the similar results for simple normal nut detection task, but BGInet performs best in detecting the normal bolts that have complicated appearances, limited semanteme, and shadow interference. Likewise, detecting lost bolts is a highly challenging visual task due to its small sample, small object, and rare semantics. The two-stage models, Faster RCNN and Cascade RCNN, and the one-stage YOLO families such as YOLOv7, YOLOv8, and YOLOv9 have not reached 80% AP for lost bolt detection. It is worth noting that YOLOv5 gets 83.6% AP for the defect category, ranking second among all models by using the classical darknet, CSP, and SPP structures. Nevertheless, the newly developed BGInet gets the highest 95.4% AP in lost bolt detection, which far outperforms YOLOv5 by about 11.8% AP. Additionally, in terms of mAP across all categories, our BGInet still performs best, which is 3.2%, 5.5%, 8.1%, 7.6%, 7.2% higher than the one-stage models YOLOv5, AOY-OLO, YOLOv7, YOLOv8, and YOLOv9, respectively. It is because BGInet assembles the novel sparse attention, parsing branch, and the other advanced network structures, eliminating the dilemmas of girder inspection, such as small object, small sample, limited semantics, and so forth. Therefore, we believe that BGInet is a high-efficiency and accurate solution for bridge girder surface defect detection.

4.3.2 | Comparisons on per-pixel parsing

Figure 11 shows the IoU comparison results of BGInet and the classic models in pixel-level object parsing. From Figure 11, it can be clearly observed that the proposed BGInet performs best, which far exceeds U_Net, PSPNet, SEM-FPN, SegFormer, Segnext, and AOYOLO by 89.8%, 86.5%, 86.5%, 82.1%, 64.8% and 2% IoU, respectively. One

TABLE 5 APs of each model on four classes from the steel bridge disease dataset.

Method	Normal nut	Normal bolt	Missing bolt	Rusty bolt	mAP
Faster RCNN	0.951	0.807	0.789	0.812	0.839
Cascade RCNN	0.945	0.819	0.734	0.817	0.828
SSD	0.940	0.809	0.812	0.828	0.849
RetinaNet	0.948	0.625	0.681	0.778	0.758
YOLACT	0.928	0.751	0.782	0.795	0.814
YOLOv5	0.965	0.803	0.836	0.836	0.860
AOYOLO	0.928	0.811	0.807	0.802	0.837
YOLOv7	0.948	0.822	0.656	0.820	0.811
YOLOv8	0.952	0.802	0.707	0.804	0.816
YOLOv9	0.960	0.788	0.702	0.830	0.820
BGInet	**0.953**	**0.829**	**0.954**	**0.833**	**0.892**

Abbreviations: BGInet, bi-task girder inspection network; mAP, mean average precision.

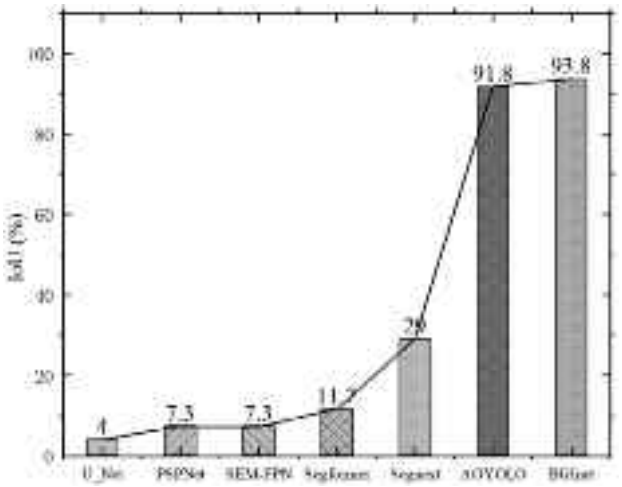

FIGURE 11 IoU comparisons of BGInet and leading models on track and road parsing.

TABLE 6 Real-time running speed of different methods.

Method	Frames per second (FPS)	Time (ms)
Faster RCNN	14.8	67.6
Cascade RCNN	9	111.1
MS RCNN	7.2	139.1
REGNet	14.7	67.9
YOLACT	32.3	31
RBGNet	6.25	160
YOLOv7	84	11.9
YOLOv9	26	37.2
BGInet	**33**	**30.3**
BGInet_l	**96.1**	**10.4**

Abbreviation: BGInet_l, bi-task girder inspection network.

possible reason is these classic semantic segmentation models, trained on the public dataset such as *PASCAL VOC 2012*, are tailored to parse general objects containing prominent and abundant semantics such as textures, colors, and edges. However, the limited semantics, irregular shapes, and random distribution of rusts make it difficult for the models equipped with sophisticated and complex network structures to learn rich semantic information. In other words, the neurons in this sophisticated network structure may not fully learn rust knowledge in the weak-semantic rust dataset, and thus unable to function effectively. In this study, this study designs a simple U-shape parsing branch using only the upsamples, conv, normalization and SiLU, which is fed to the deepest layers in BGInet. The concise structure indeed achieves satisfactory effects in the rust parsing by 94% IoU. Given all of that, we are confident BGInet is a very effective solution for all girder structure surface inspection and provides a solid foundation for the following rust measurement.

4.4 | BGInet edge deployment

Table 6 presents comparative results of BGInet and the advanced models in terms of the real-time processing speed. Remarkably, YOLOv7 and YOLACT rank second and fourth with 84 and 32.3 FPS, respectively. The proposed BGInet ranks third at 33 FPS, weaker than YOLOv7. This is because BGInet integrates a pixel-level parsing branch dedicated to rust parsing, which greatly increases the computational parameters, consequently slowing down the inference speed. Nevertheless, our BGInet still defeats the real-time instance model YOLACT (33 vs. 32.3 FPS). Furthermore, the processing speeds exceeding 30 FPS prove more than sufficient for on-site deployment and practical application. Note this study also

TABLE 7 The deep learning environment and hardware for model training.

Item	Strategy
CPU	NVIDIA Carmel ARM CPU
GPU	NVIDIA Volta GPU
Operate system	Ubuntu 18.04
Programming language	Python 3.7
Deep learning framework	PyTorch 1.8.0
Optimizer	Stochastic gradient descent
Momentum	0.9
Total training epochs	300
Batch size	4

developed a lightweight BGInet variant, (BGInet_l), specifically for detecting bolts and nut defects of bridge girder surface. As shown in Table 6, the BGInet_l can achieve 96.1 FPS, ranking first among all models. Therefore, both BGInet and BGInet_l is able to complete high-accuracy bridge girder inspection with a reasonable processing speed.

Since YOLOv7 performs best among other SOTA models and YOLOv7 can only achieve target detection but not segmentation, comparative experiments were conducted on NVIDIA Jetson Xavier NX embedded devices (see the Table 7 for specific experimental configuration) and NVIDIA GeForce RTX 2070 Super cards to compare the performance of YOLOv7 and the proposed BGInet_l in target detection. Tables 8 and 9 show the inference time of the lightweight BGInet_l and YOLOv7 running on Xavier NX and NVIDIA Geforce RTX 2070 super cards. Note that we employ Open Neural Network Exchange without any C++ extension or acceleration methods to transform the two models for the board deployment. Then, the inference speed and object detection rate (mAP) of various models are evaluated under different image resolutions. From Tables 8 and 9, under all the image resolutions, the lightweighted BGInet_l defeats YOLOv7 in processing speed (FPS). Especially for mAP, BGInet_l far exceeds YOLOv7 under resolutions of 286 × 286, 320 × 320, and 400 × 400. Also, our BGInet_l only occupies 7.8 M of memory resources, which has a significantly smaller memory footprint than YOLOv7. As a result, it remains the best choice for onboard computers mounted on drones for girder bolt and nut inspection, underscoring its suitability for real-time applications with limited memory resources.

4.5 | Visualization

Extensive visual experiments were conducted to further validate the performance of the proposed BGInet. Overall, BGInet effectively detects structural defects in railway girder bridges as illustrated in Figure 12. For instance, the red, yellow, and orange boxes in the third row of Figure 12 depict the successful detection and accurate classification for normal nuts, normal bolts, rusty bolts, and missing bolts, respectively. This underscores the capability of BGInet in efficiently detecting dense targets, particularly challenging samples like missing bolts. Besides, From Figure 12, BGInet successfully parses all rust areas by precisely predicting pixel-level masks. It is noteworthy that in the first row, the image in the second column lacks annotations for two rusted areas, while the image in the first column of the third row contains densely distributed nuts; nevertheless, BGInet is still able to accurately predict their positions and classifications. Additionally, various bolts in bad weather in the third column of the last row and normal nuts hidden in shadows in the fourth column were also accurately identified. In conclusion, our BGInet presents an effective and accurate solution for detecting girder surface defects in railway bridges.

Finally, this study assembles the proposed BGInet and pixel-to-real-world rust measurement model into a system. Then the rust assessment experiments are conducted based on the images in the second row of Figure 12. Table 10 presents the area measurement results from the system in which the rusts are predicted by our BGInet. From Figure 12, we can observe an interesting phenomenon: The larger the rust area, the greater the impact on the model's processing speed. This is because as the corrosion areas increase, the information that the system needs to process also increases. To summarize, we are confident that the proposed approaches and system are a very effective solution for girder structure defect detection in railway bridges.

5 | CONCLUSION

This study addresses the key challenges of using drones for railway bridge inspection, focusing on the need for real-time performance and high detection accuracy. Engineering contributions: (1) Develop a dedicated inspection system for railway bridge truss assessment based on drones. (2) A "pixel to real world" mapping model based on key drone parameters is established and integrated into the system for corrosion area assessment. In addition, our on-site experiments have indicated the proposed pixel-to-real-world mapping model can reach over 96% accuracy. Note the on-site rust repair involves coating and painting the minimum bounding rectangle of the corroded area. Hence, the 96% accuracy does not affect the repair decision and can meet the maintenance requirements. Theoretical contributions: (1) Introduce the BGInet framework, which includes a novel SAM to enhance feature extraction,

TABLE 8 Comparison of deploying on a single NVIDIA Jetson Xavier NX embedded device.

	mAP		FPS		Model size	
Image size	**YOLOv7**	**BGInet_l**	**YOLOv7**	**BGInet_l**	**YOLOv7**	**BGInet_l**
286 × 286	0.575	0.819	7.6	10.6	74.8 M	7.8 M
320 × 320	0.671	0.798	6.7	11	74.8 M	7.8 M
400 × 400	0.808	0.83	5.7	10.8	74.8 M	7.5 M
480 × 480	0.804	0.789	5.2	8.6	74.8 M	7.8 M
512 × 512	0.812	0.806	4.5	7.7	74.8 M	7.8 M

Abbreviation: BGInet_l, lightweight bi-task girder inspection network.

TABLE 9 Comparison of inference time on a single NVIDIA Geforce RTX 2070 super card.

	mAP		FPS		Model size	
Image size	**YOLOv7**	**BGInet_l**	**YOLOv7**	**BGInet_l**	**YOLOv7**	**BGInet_l**
286 × 286	0.575	**0.819**	85.4	**100**	74.8 M	**7.8 M**
320 × 320	0.672	**0.798**	83.3	**100**	74.8 M	**7.8 M**
400 × 400	0.809	**0.83**	70.4	**98**	74.8 M	**7.5 M**
480 × 480	0.804	0.789	67.1	**97**	74.8 M	**7.8 M**
512 × 512	0.812	0.806	62.1	**96.1**	74.8 M	**7.8 M**

Abbreviations: BGInet_l, lightweight bi-task girder inspection network.

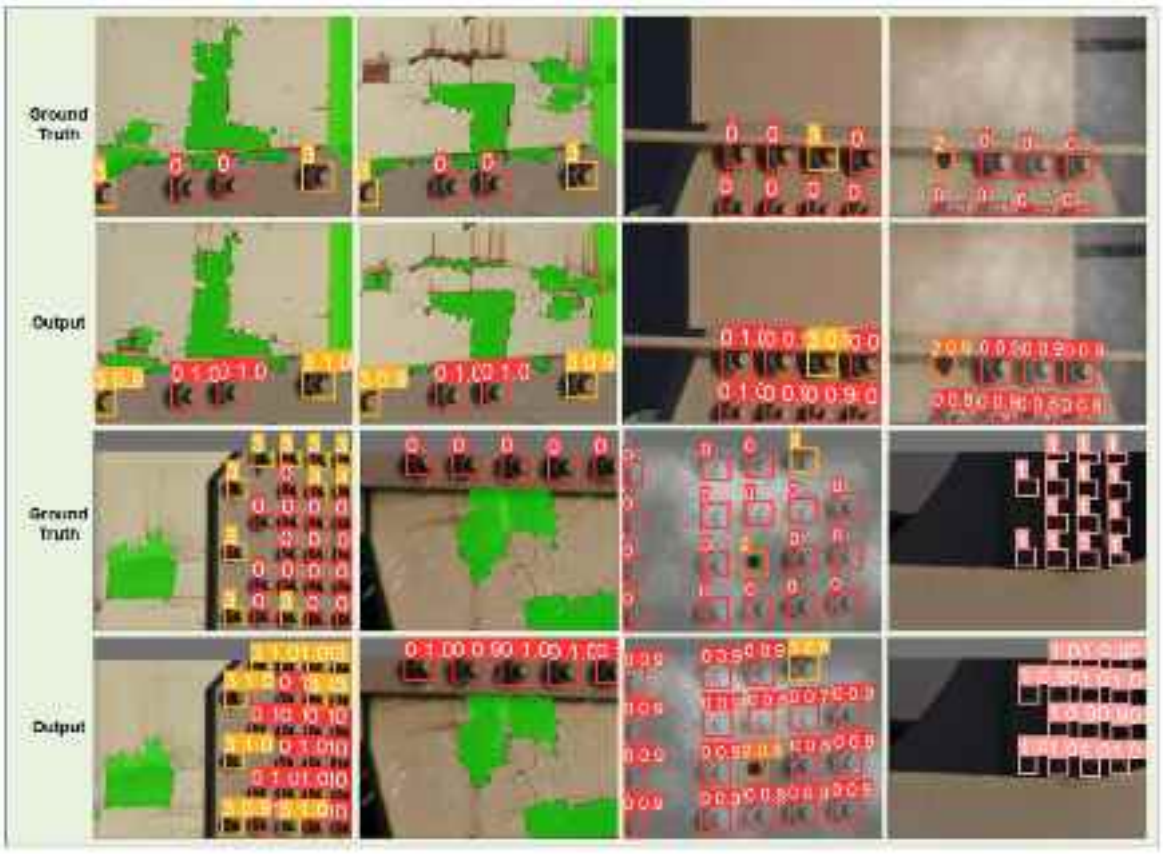

FIGURE 12 Extensive visualization results from the drone-based railway girder bridge dataset. Note the second and third columns of the last row show the images containing rain and haze, respectively.

TABLE 10 Rust area assessment of the total system.

Rust	Actual area	FPS
Image 3	0.16040 m^2	31.60
Image 4	0.35537 m^2	31.25
Image 2	0.38901 m^2	30.89
Image 1	0.39987 m^2	30.48

especially improve the detection accuracy of small samples, and reduce the amount of computation and memory usage. (2) Use E-ELAN and simplify the detection head design to improve computational efficiency and processing speed. Compared with those advanced segmentation methods, BGInet achieves the highest 93.8% IoU among all models. When comparing to those leading object detection models, BGInet also performs best with 89.2% mAP. In the industry, on-site railway bridge inspection pays more attention to the "recall" rate than mAP or precision because a miss-detection may lead to severe consequences (Meng et al., 2024; Wu et al., 2022). Note that our model gets 89.2% of mAP, while it can achieve over 91% of recall. This can meet the needs of on-site bridge maintenance personnel. In conclusion, BGInet excels in simultaneously performing both defect detection and rust parsing tasks for railway bridge girders. Also, the total inspection system is an effective solution for the surface defect detection and evaluation of bridge girder. Nevertheless, there is still room for further improvement:

1. The network parameter of BGInet still remains redundant. Hence, efforts will be focused on further optimizing the algorithm to make the model lighter.
2. A software package of low computer and memory consumption is currently in development. In future, drones will be employed with a customized software package to achieve an on-site inspection of high precision and low energy consumption.
3. Adverse weather conditions like heavy snow, rain, or dense fog challenge UAV-based bridge defect inspection by reducing visibility and causing blurriness, distortion, and low-light issues in images. Therefore, it is crucial to develop an efficient Potential safety hazard (PSH) inspection algorithm that enhances the model's adaptability and resilience in these conditions.
4. Our U-shaped parsing branch is tailored for bridge girder rust parsing, while the network structures that can capture linear features and finer edge information, such as residual networks or networks with dilated convolutions, may be more suitable for crack parsing.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 52362048, in part by Yunnan Fundamental Research Projects (No. 202301BE070001-042, No. 202401AT070409).

REFERENCES

Alaie, S., & Al'Aref, S. J. (2023). Application of deep neural networks for inferring pressure in polymeric acoustic transponders/sensors. *Machine Learning with Applications*, *13*, 100477.

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: real-time instance segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, South Korea (pp. 9157–9166).

Cai, Z., & Vasconcelos, N. (2018). Cascade R-CNN: Delving into high quality object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT (pp. 6154–6162).

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). RepVGG: Making VGG-style convnets great again. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN (pp. 13733–13742).

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., & Guo, B. (2022). CSWin Transformer: A general vision transformer backbone with cross-shaped windows. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA (pp. 12124–12134).

Gao, Y., Yang, J., Qian, H., & Mosalam, K. M. (2023). Multiattribute multitask transformer framework for vision-based structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *38*(17), 2358–2377.

Guo, M.-H., Lu, C.-Z., Hou, Q., Liu, Z., Cheng, M.-M., & Hu, S.-M. (2022). SegNeXt: Rethinking convolutional attention design for semantic segmentation. *Advances in Neural Information Processing Systems*, *35*, 1140–1156.

Guo, Y., Xu, Y., Niu, J., & Li, S. (2023). Anchor-free arbitrary-oriented construction vehicle detection with orientation-aware Gaussian heatmap. *Computer-Aided Civil and Infrastructure Engineering*, *38*(7), 907–919.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1904–1916.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT (pp. 7132–7141).

Jia, H., Chen, S., Guo, D., Zheng, S., & Zhao, C. (2024). Track-bridge deformation relation and interaction of long-span railway suspension bridges subject to strike-slip faulting. *Engineering Structures*, *300*, 117216.

Jiang, X., & Adeli, H. (2005). Dynamic wavelet neural network for nonlinear identification of highrise buildings. *Computer-Aided Civil and Infrastructure Engineering*, *20*(5), 316–330.

Jiang, Y., Pang, D., Li, C., & Wang, J. (2024). A method of concrete damage detection and localization based on weakly supervised learning. *Computer-Aided Civil and Infrastructure Engineering*, *39*(7), 1042–1060.

Lim, H. J., Hwang, S., Kim, H., & Sohn, H. (2021). Steel bridge corrosion inspection with combined vision and thermographic images. *Structural Health Monitoring*, *20*(6), 3424–3435.

Jocher, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Diaconu, L., Poznanski, J., Yu, L., Rai, P., & Ferriday, R. J. Z. (2020). Ultralytics/Yolov5: V3. 0. [Online]

Katsamenis, I., Doulamis, N., Doulamis, A., Protopapadakis, E., & Voulodimos, A. (2022). Simultaneous precise localization and classification of metal rust defects for robotic-driven maintenance and prefabrication using residual attention U-Net. *Automation in Construction*, *137*, 104182.

Kunlamai, T., Yamane, T., Suganuma, M., Chun, P. J., & Okatani, T. (2024). Improving visual question answering for bridge inspection by pre-training with external data of image–text pairs. *Computer-Aided Civil and Infrastructure Engineering*, *39*(3), 345–361.

Lamas, D., Justo, A., Soilán, M., Cabaleiro, M., & Riveiro, B. (2023). Instance and semantic segmentation of point clouds of large metallic truss bridges. *Automation in Construction*, *151*, 104865.

Lamas, D., Justo, A., Soilán, M., & Riveiro, B. (2024). Automated production of synthetic point clouds of truss bridges for semantic and instance segmentation using deep learning models. *Automation in Construction*, *158*, 105176.

Li, H., Chen, Y., Liu, J., Che, C., Meng, Z., & Zhu, H. (2024). High-resolution model reconstruction and bridge damage detection based on data fusion of unmanned aerial vehicles light detection and ranging data imagery. *Computer-Aided Civil and Infrastructure Engineering*, *39*(8), 1197–1217.

Li, R., Yu, J., Li, F., Yang, R., Wang, Y., & Peng, Z. (2023). Automatic bridge crack detection using unmanned aerial vehicle and faster R-CNN. *Construction and Building Materials*, *362*, 129659.

Li, Z., Lin, W., & Zhang, Y. (2023). Real-time drive-by bridge damage detection using deep auto-encoder. *Structures*, *47*, 1167–1181.

Li, Z.-J., Adamu, K., Yan, K., Xu, X.-L., Shao, P., Li, X.-H., & Bashir, H. M. (2022). Detection of nut–bolt loss in steel bridges using deep learning techniques. *Sustainability*, *14*(17), 10837.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI (pp. 2117–2125).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Lecture notes in computer science: Vol. 9905. Computer vision–ECCV 2016: 14th European conference* (pp. 21–37). Springer.

Liu, Y., Zhou, T., Xu, J., Hong, Y., Pu, Q., & Wen, X. (2023). Rotating target detection method of concrete bridge crack based on Yolo V5. *Applied Sciences*, *13*(20), 11118.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, BC, Canada (pp. 10012–10022).

Long, S., Yang, T., Qian, Y., Wu, Y., Xu, F., Tang, Q., & Guo, F. (2024). GPR imagery based internal defect evaluation system for railroad tunnel lining using real-time instance segmentation. *IEEE Sensors Journal*, *24*(21), 35997–36010.

Ma, M., Yang, L., Liu, Y., & Yu, H. (2024). A transformer-based network with feature complementary fusion for crack defect detection. *IEEE Transactions on Intelligent Transportation Systems*, *25*(11), 16989–17006.

Matono, G., & Nishio, M. (2024). Component-level point cloud completion of bridge structures using deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *39*(17), 2581–2595.

Meng, F., Qin, Y., Wu, Y., Shao, C., & Jia, L. (2024). A subtle defect recognition method for catenary fastener in high-speed railroad using destruction and reconstruction learning. *Advanced Engineering Informatics*, *60*, 102393.

Mu, Z., Qin, Y., Yu, C., Wu, Y., Wang, Z., Yang, H., & Huang, Y. (2023). Adaptive cropping shallow attention network for defect detection of bridge girder steel using unmanned aerial vehicle images. *Journal of Zhejiang University-SCIENCE A*, *24*(3), 243–256.

Murray, N., & Perronnin, F. (2014). Generalized max pooling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH (pp. 2473–2480).

Nettis, A., Massimi, V., Nutricato, R., Nitti, D. O., Samarelli, S., & Uva, G. (2023). Satellite-based interferometry for monitoring structural deformations of bridge portfolios. *Automation in Construction*, *147*, 104707.

Pan, X., & Yang, T. (2024). Bolt loosening assessment using ensemble vision models for automatic localization and feature extraction with target-free perspective adaptation. Computer-Aided Civil and Infrastructure Engineering. Advance online publication. https://doi.org/10.1111/mice.13355

Pan, X., Yang, T., Xiao, Y., Yao, H., & Adeli, H. (2023). Vision-based real-time structural vibration measurement through deep-learning-based detection and tracking methods. *Engineering Structures*, *281*, 115676.

Pezeshki, H., Adeli, H., Pavlou, D., & Siriwardane, S. C. (2023). State of the art in structural health monitoring of offshore and marine structures. *Proceedings of the Institution of Civil Engineers—Maritime Engineering*, *176*(2), 89–108.

Pezeshki, H., Pavlou, D., Adeli, H., & Siriwardane, S. C. (2023). Modal analysis of offshore monopile wind turbine: An analytical solution. *Journal of Offshore Mechanics and Arctic Engineering*, *145*(1), 010907.

Qi, H., Kong, X., Jin, Z., Zhang, J., & Wang, Z. (2024). A vision-transformer-based convex variational network for bridge pavement defect segmentation. *IEEE Transactions on Intelligent Transportation Systems*, *25*(10), 13820–13832.

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, *26*(18), e1400.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2), 237.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. Wells, & A. Frangi (Eds.), *Lecture notes in computer science: Vol. 9351. Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference* (pp. 234–241) Springer.

Sacks, R., Kedar, A., Borrmann, A., Ma, L., Brilakis, I., Hüthwohl, P., Daum, S., Kattel, U., Yosef, R., Liebich, T., Barutcu, B. E., & Muhic, S. (2018). SeeBridge as next generation bridge inspection: Overview, information delivery manual and model view definition. *Automation in Construction*, *90*, 134–145.

Tan, M., & Le, Q. (2019), Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks, International Conference on Machine Learning, PMLR (pp. 6105–6114).

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., & Li, Y. (2022). MaxViT: Multi-axis vision transformer. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & Tt. Hassner (Eds.), *Lecture notes in computer science: Vol. 13684. European conference on computer vision* (pp. 459–479). Springer.

Vivekananthan, V., Vignesh, R., Vasanthaseelan, S., Joel, E., & Kumar, K. S. (2023). Concrete bridge crack detection by image processing technique by using the improved Otsu method. *Materials Today: Proceedings*, *74*, 1002–1007.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada (pp. 7464–7475).

Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, WA (pp. 390–391).

Wang, C.-Y., Liao, H.-Y. M., & Yeh, I.-H. (2022). *Designing network design strategies through gradient path analysis*. arXiv preprint arXiv:2211.04800.

Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2024). *YOLOv9: Learning what you want to learn using programmable gradient information*. arXiv 2024. arXiv preprint arXiv:2402.13616.

Wang, F., Zou, Y., del Rey Castillo, E., Ding, Y., Xu, Z., Zhao, H., & Lim, J. B. (2024). Automated UAV path-planning for high-quality photogrammetric 3D bridge reconstruction. *Structure and Infrastructure Engineering*, *20*(10), 1595–1614.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: Efficient channel attention for deep convolutional neural networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA (pp. 11534–11542).

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. In: V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer vision—Lecture notes in computer science: Vol. 11211. ECCV 2018, The European conference on computer vision (ECCV)* (pp. 3–19). Springer.

Wu, Y., Chen, P., Qin, Y., Qian, Y., Xu, F., & Jia, L. (2023). Automatic railroad track components inspection using hybrid deep learning framework. *IEEE Transactions on Instrumentation and Measurement*, *72*, 5011415.

Wu, Y., Qin, Y., Qian, Y., Guo, F., Wang, Z., & Jia, L. (2022). Hybrid deep learning architecture for rail surface segmentation and surface defect detection. *Computer-Aided Civil and Infrastructure Engineering*, *37*(2), 227–244.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, *34*, 12077–12090.

Xu, J., Gui, C., & Han, Q. (2020). Recognition of rust grade and rust ratio of steel structures based on ensembled convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, *35*(10), 1160–1174.

Xu, Y., Zhang, Y., & Zhang, J. (2023). Bridge acceleration data cleaning based on two-stage classification model with multiple feature fusion. *Applied Sciences*, *13*(21), 12045.

Yamaguchi, T., & Mizutani, T. (2024). Road crack detection interpreting background images by convolutional neural networks and a self-organizing map. *Computer-Aided Civil and Infrastructure Engineering*, *39*(11), 1616–1640.

Yamane, T., Chun, P. J., Dang, J., & Honda, R. (2023). Recording of bridge damage areas by 3D integration of multiple images and reduction of the variability in detected results. *Computer-Aided Civil and Infrastructure Engineering*, *38*(17), 2391–2407.

Yamane, T., Chun, P.-J., & Honda, R. (2024). Detecting and localising damage based on image recognition and structure from motion, and reflecting it in a 3D bridge model. *Structure and Infrastructure Engineering*, *20*(4), 594–606.

Yin, Y., Yu, Q., Hu, B., Zhang, Y., Chen, W., Liu, X., & Ding, X. (2023). A vision monitoring system for multipoint deflection of large-span bridge based on camera networking. *Computer-Aided Civil and Infrastructure Engineering*, *38*(13), 1879–1891.

Zhang, H., Shen, Z., Lin, Z., Quan, L., & Sun, L. (2024). Deep learning-based automatic classification of three-level surface information in bridge inspection. *Computer-Aided Civil and Infrastructure Engineering*, *39*(10), 1431–1451.

Zhang, J., Qian, S., & Tan, C. (2022). Automated bridge surface crack detection and segmentation using computer vision-based deep learning model. *Engineering Applications of Artificial Intelligence*, *115*, 105225.

Zhang, J., Yang, X., Wang, W., Guan, J., Liu, W., Wang, H., Ding, L., & Lee, V. C. (2024). Cross-entropy-based adaptive fuzzy control for visual tracking of road cracks with unmanned mobile robot. *Computer-Aided Civil and Infrastructure Engineering*, *39*(6), 891–910.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI (pp. 2881–2890).

Zheng, Y., Gao, Y., Lu, S., & Mosalam, K. M. (2022). Multistage semisupervised active learning framework for crack identification, segmentation, and measurement of bridges. *Computer-Aided Civil and Infrastructure Engineering*, *37*(9), 1089–1108.

Zhu, L., Wang, X., Ke, Z., Zhang, W., & Lau, R. W. (2023). BiFormer: Vision transformer with bi-level routing attention. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, BC, Canada (pp. 10323–10333).

How to cite this article: Xu, T., Wu, Y., Qin, Y., Long, S., Yang, Z., & Guo, F. (2025). Automatic steel girder inspection system for high-speed railway bridge using hybrid learning framework. *Computer-Aided Civil and Infrastructure Engineering*, *40*, 1508–1527. https://doi.org/10.1111/mice.13409