scientific reports

OPEN

Studying the performance of YOLOv11 incorporating DHSA BRA and PPA modules in railway track fasteners defect detection

Chengwei Zhang[1]✉, Jiawei Zhu[1], Yihao Ma[2] & Qingmei Huang[1]

With the development of railway transportation and the advancement of deep learning, object detection algorithms are increasingly replacing manual inspection of track fasteners. However, current algorithms struggle with low accuracy in complex weather conditions or low-contrast backgrounds. To address this, we propose a track fastener defect detection algorithm based on YOLOv11 (You Only Look Once).First, we incorporate the DHSA (Dynamic-range Histogram Self-Attention) module into the backbone network of YOLOv11 to enhance noise robustness. Second, we introduce the BRA (Bi-Level Routing Attention) sparse attention mechanism into the neck network for improved efficiency. Finally, we add the PPA (Parallelized Patch-Aware Attention) module to the original neck network to enhance multi-scale feature extraction, specifically for small object detection.To validate the model, we created a dataset and conducted experiments. The experimental results show that YOLO-DRPA achieves a mAP@0.5 of 94.6% and a mAP@0.5:0.95 of 80.7%, marking improvements of 1.8% and 4.0% over YOLOv11n, respectively. The model also demonstrates competitive performance compared to other popular object detection algorithms, highlighting its potential to improve both detection accuracy and efficiency.

Keywords Defect detection, Deep learning, YOLO11, Feature fusion, Attention mechanism

In recent decades, railway transportation has received extensive attention and experienced significant growth, aiming to further boost industrial production efficiency and accelerate socioeconomic development. However, this rapid growth has also led to an increase in potential safety hazards, which pose serious risks to industrial production and social stability. Track fasteners, essential components that connect railway tracks to sleepers, play a crucial role in maintaining track stability. The health of these fasteners directly affects the stability and efficiency of railway transportation. Unfortunately, track fasteners are prone to defects due to factors such as fatigue from prolonged use, uneven stress on the tracks, and adverse weather conditions. Common defects include breaks, lack of fastening, rotation, among others[1], which severely hinder the normal operation of railway systems.

Currently, there are various methods for detecting defects in railway track fasteners. Traditional manual inspection relies heavily on the experience and skills of inspectors[2], but it is inefficient and subject to human error, leading to missed or false detections. As a result, image-processing-based machine learning methods have emerged as alternatives to manual detection. These methods often use hand-crafted feature extraction algorithms and classifiers such as SIFT[3] (Scale-Invariant Feature Transform), SURF[4] (Speeded-Up Robust Features), and HOG[5] (Histogram of Oriented Gradients). While these approaches offer improved efficiency and stability compared to manual inspection, their performance is still limited by the accuracy and robustness of the algorithms. Consequently, deep-learning-based methods, particularly those employing CNNs[6] (Convolutional Neural Networks) and RCNNs[7] (Region-based Convolutional Neural Networks), have gained widespread application.

Deep learning-based object detection primarily falls into two categories: two-stage and one-stage algorithms. Two-stage algorithms first generate candidate regions with target probabilities and then process these regions through ROI (Region of Interest) Pooling. Notable examples of this approach include RCNN and Faster RCNN[8] (Faster Region-based Convolutional Neural Network). In contrast, one-stage algorithms treat detection as a

[1]College of Engineering, Zhejiang Normal University, Yingbin Avenue, Jinhua 321005, China. [2]College of Physics and Electronic Information Engineering, Zhejiang Normal University, Yingbin Avenue, Jinhua 321005, China. ✉email: ZhangCW233666@163.com

regression task, directly predicting target categories and bounding boxes from images. Examples of one-stage algorithms include YOLO[9] and SSD[10] (Single Shot MultiBox Detector). Feng Guo et al. developed the YOLOv4-hybrid[11] model, which builds upon YOLOv4 to enable portable, high-speed track detection. While two-stage algorithms excel in accuracy, one-stage algorithms are typically faster. Additionally, Feng Guo et al. proposed RailFormer[12], a Transformer-based model designed for precise pixel-level RSD (Rail Surface Defects) detection in the railway industry. Furthermore, multi-sensor data fusion[13], such as the use of millimeter-wave radar, has been adopted to enhance detection performance, although this approach comes with higher hardware costs.

In general, defect detection of railway track fasteners involves a variety of methods, with deep learning-based approaches showing promising comprehensive performance and application potential. Considering specific challenges in railway track fastener defect detection, such as the unavoidable impact of weather conditions (e.g., rainy days, nighttime, tunnel environments), the low proportion of fastener targets in input detection images causing inefficient computation in traditional undifferentiated attention mechanisms, and the small size of fastener targets leading to the loss of important details during traditional feature extraction and downsampling, we propose an innovative model, YOLO11-DRPA, based on YOLO11. By incorporating the DHSA module[14], BRA module[15], and PPA module[16], the model's ability to address these challenges is improved. Specifically, the DHSA module added to the backbone network groups input images by gray levels and applies BHR (Bin-wise Histogram Reshaping) and FHR (Frequency-wise Histogram Reshaping) to enhance sensitivity to both global and local features, reducing weather noise interference and improving adaptability to tunnel environments. The BRA module in the neck network performs spatial division and sparse attention processing on fastener images, filtering out irrelevant information and helping optimize computing power usage. The PPA module, also in the neck network, uses pointwise convolution to partition input data into distinct feature groups, helping mitigate feature loss during downsampling and enhancing the model's sensitivity to fasteners.

Methods

Dataset

Currently, datasets related to track fastener defects are relatively limited, and due to confidentiality concerns, most railway data are not publicly accessible. As a result, the track fastener dataset used in this study was sourced from Jinhua Railway Station in Zhejiang. It took approximately three months to capture 1,489 fastener images with a resolution of 4032 × 3024 from different angles and heights, under various weather conditions, using a manual camera. These images cover different fastener categories, including normal, rotation, lack, and break, as illustrated in Fig. 1. Additionally, real-world scenarios such as rainy days and low-light tunnel conditions were simulated to account for various lighting and occlusion challenges typically encountered in track detection tasks.

To increase the dataset size and ensure the sample distribution reflects real-world conditions, we applied several data augmentation techniques, including image position transformation, noise addition, and adjustments

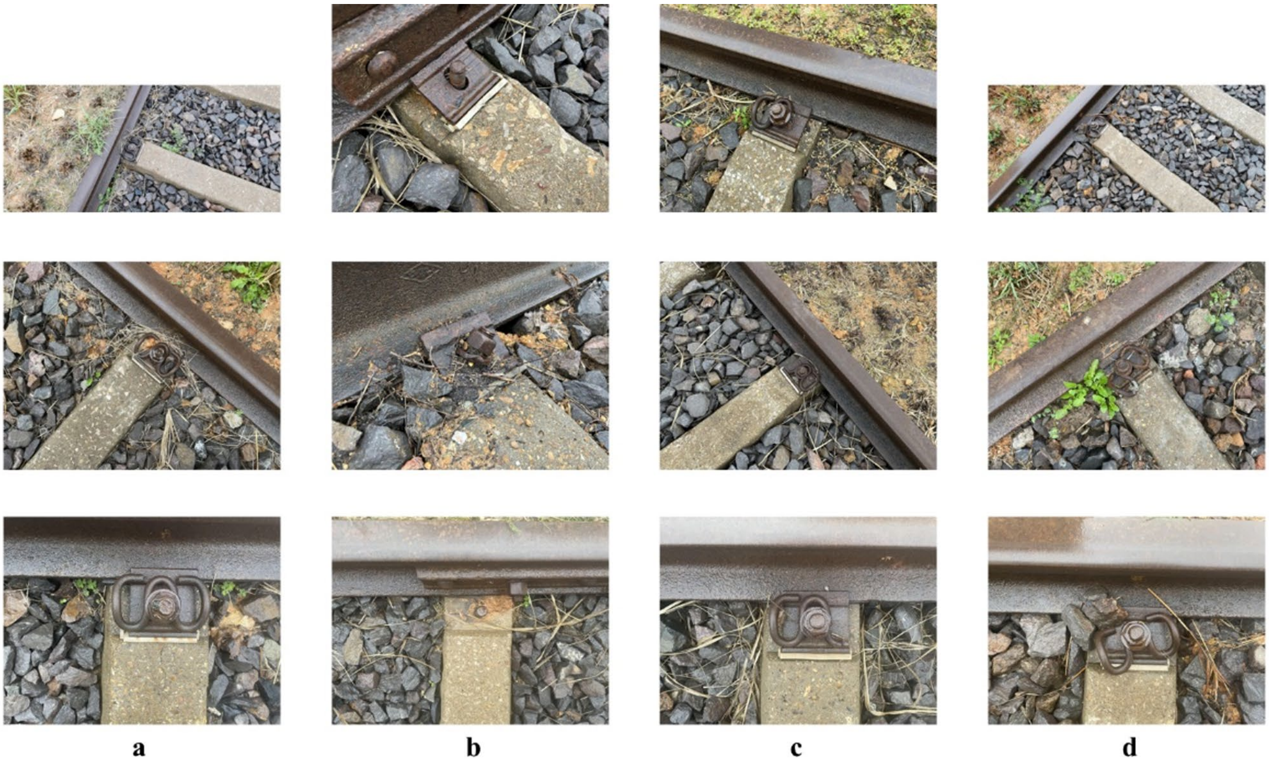

Fig. 1. Dataset category examples. In the figure, (**a**–**d**) are respectively examples of image dataset for the categories of Normal, Lack, Break, and Rotation.

to brightness and contrast. Subsequent experiments have demonstrated that variations in the sample size do not significantly affect the model's detection performance. After screening and labeling, the dataset was split into training, validation, and test sets following an 8:1:1 ratio.

Experimental setup

In this study, we used Ubuntu 20.04 as the operating system and the PyTorch framework for model training. The software setup included PyTorch 2.0.1, CUDA 11.8, and CUDNN 8.7.0. The hardware configuration consisted of an AMD Ryzen 9 5900HX with Radeon Graphics x 16 and an NVIDIA GeForce RTX 3060 with 6GB of memory. For data augmentation, we utilized Albumentations[17]. New images were generated by stitching four different images together, along with applying transformations such as rotation, translation, and cropping. The experimental setup was configured with the following parameters: no pre-trained weights, a total of 300 epochs, a batch size of 16, an input image resolution of 640 × 640, and a cosine annealing learning rate strategy[18].

YOLO-DRPA

YOLO-DRPA striction

The network structure of YOLO11 is divided into three parts: the backbone network, the neck network, and the output head. The backbone network is composed of four modules, namely Conv, C3k2, SPPF, and C2PSA. It generates rich feature tensors by gradually reducing the spatial resolution of the image and increasing the number of channels, and is mainly responsible for extracting features of the input image and capturing image information. The neck network is located between the backbone network and the output head, and its main part is the PAN (Path Aggregation Network), which is responsible for further enhancing and fusing the feature tensors output by the backbone network to provide better input for the output head. The output head is responsible for receiving the data output by the neck network and generating the final prediction results. It consists of three independent output heads, which respectively process feature maps of large, medium, and small sizes to meet the object detection requirements of different sizes. Finally, the network outputs a high-dimensional tensor containing category, confidence, and position coordinate information.

On this basis, we added the DHSA module to the backbone network and the BRA module and PPA module to the neck network. The specific network structure is shown in Fig. 2.

YOLO-DRPA backbone

In order to enhance the robustness of the model against the degradation factors caused by the inevitable weather environments (such as rainy days, nights, railway track tunnels, etc.) during the detection of railway track fasteners, including image noises like occlusions and changes in brightness[19], we have introduced the DHSA module into the backbone network. The network structure of the DHSA module is shown in Fig. 3. It improves the model's robustness to environmental noises through dynamic range convolution and a dual-path histogram self-attention mechanism.

In the dynamic range convolution, this module conducts convolution within a dynamic range on the input image. For a dynamic input feature $\mathrm{F} \in \mathrm{R}^{W \times H \times C}$, where $W \times H$ represents the spatial scale and C represents the number of channels, it is equally divided along the channel dimension into intermediate feature tensors $\mathrm{F}_1 \in \mathrm{R}^{W \times H \times \frac{C}{2}}$ and $\mathrm{F}_2 \in \mathrm{R}^{W \times H \times \frac{C}{2}}$. The grayscale values of the data in the F_1 branch are sorted horizontally and vertically, and then connected with F_2 to obtain the adaptively processed feature F', as shown in Eq. (1):

$$\begin{aligned} F_1, F_2 &= Split\left(F\right) \ , \ F_1 = Sort_v\left(Sort_h\left(F_1\right)\right), \\ F' &= Conv_{3 \times 3}^{d}\left(Conv_{1 \times 1}\left(Concat\left(F_1, F_2\right)\right)\right) \end{aligned} \tag{1}$$

where, $\mathrm{Conv}_{1\times1}$ denotes the 1×1 pointwise convolution operation; $\mathrm{Conv}^{\mathrm{d}}_{3\times3}$ represents the3×3 depthwise convolution operation; Concat refers to the operation of concatenating along the channels; Split stands for the operation of splitting along the channels; and $\mathrm{Sort}_{\mathrm{i}\in(\mathrm{h},\ \mathrm{v})}$ indicates the operation of arranging in the horizontal (h) or vertical (v) direction.

Subsequently, through depthwise separable convolution, the convolution operation is enabled to compute across dynamic ranges. This process organizes pixels of varying intensities into regular patterns, allowing convolution kernels to focus on preserving clean information and restoring degraded features respectively, thereby reducing noise interference.

In the dual-path histogram self-attention mechanism, for the output of the dynamic range convolution, the query-key pairs are first sorted and then passed to the two branches based on the index arrangement, which is used to extract local and global features simultaneously, as shown in Eq. (2).

$$\begin{aligned} V,\ d &= Sort\left(R_{H \times W \times C}^{HW \times C}\left(V\right)\right), \\ Q_1, K_1 &= Split\left(Gather\left(R_{H \times W \times C}^{HW \times C}\left(QK_1\right), d\right)\right) \\ Q_2, K_2 &= Split\left(Gather\left(R_{H \times W \times C}^{HW \times C}\left(QK_2\right), d\right)\right) \end{aligned} \tag{2}$$

where, $\mathrm{R}_{H \times W \times C}^{HW \times C}$ represents the reshaping of the feature tensor from $\mathrm{R}^{H \times W \times C}$ to $\mathrm{R}^{HW \times C}$, d is the index of the permutation value, and Gather denotes the retrieval of tensor elements from the given indices.

Finally, the dual-branch architecture employs BHR and FHR for feature reshaping. In the BHR branch, the number of bins is denoted as B_{b}, with each bin containing $\frac{\mathrm{HW}}{\mathrm{B}}$ features. Large-scale information is extracted from bins containing a large number of dynamically positioned pixels, covering a broader intensity range

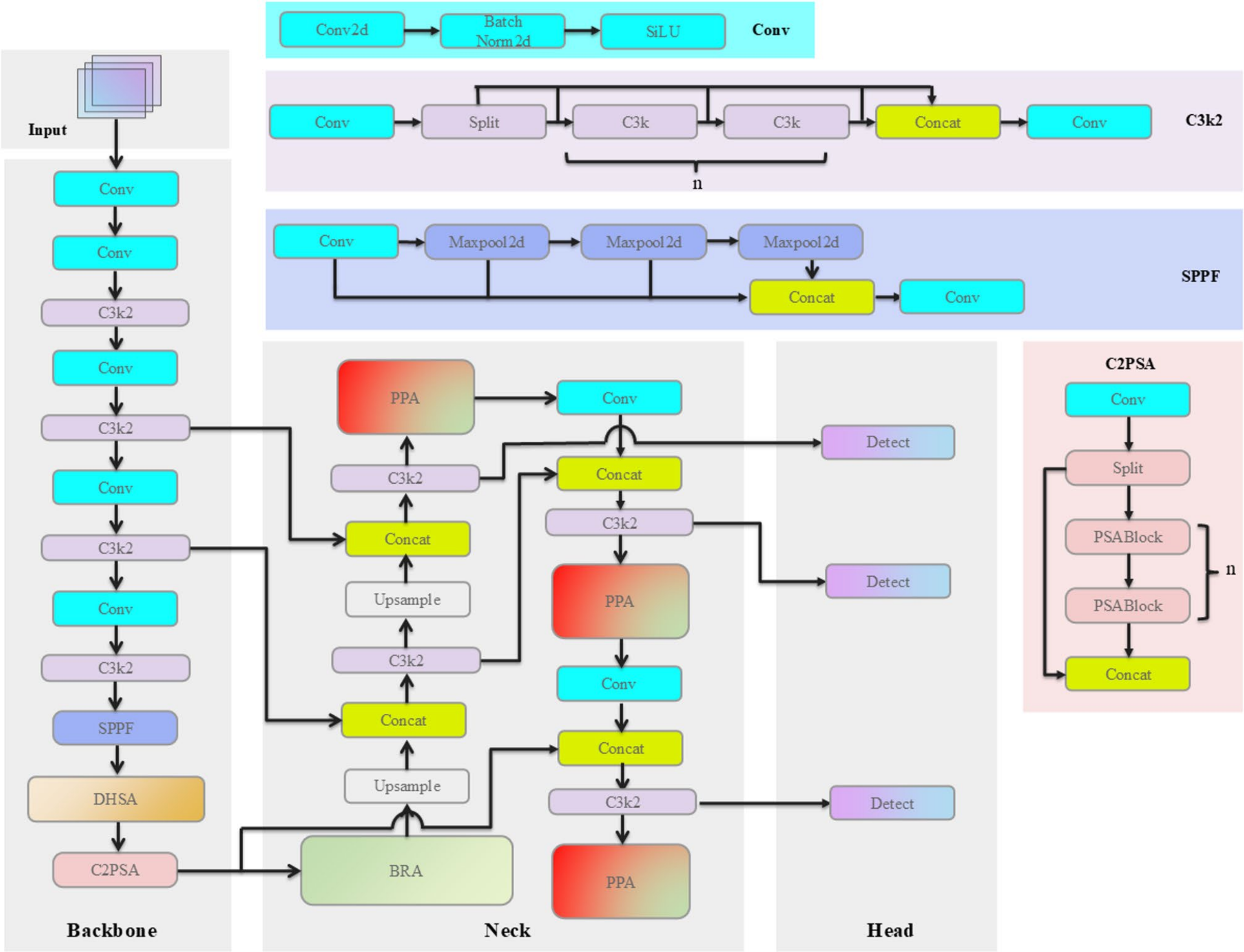

Fig. 2. The network structure of YOLO-DRPA. The structure of YOLO11-DRPA is shown in the figure, with the details of the DHSA, BRA, and PPA modules provided in the section on YOLO-DRPA.

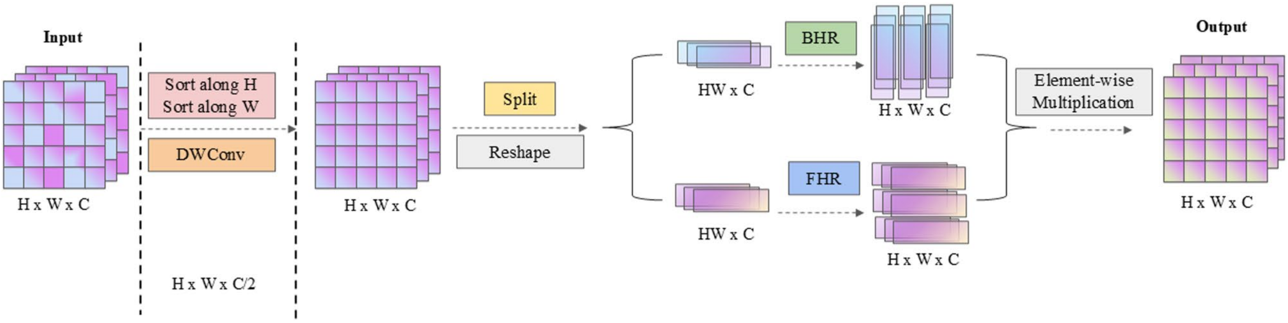

Fig. 3. The network structure of DHSA module. The legend depicts the main structure of the DHSA. The process of processing dynamic image data is encapsulated in the first step. Subsequently, reshaping operations of BHR and FHR are carried out to obtain the complete output features.

to integrate global features and prevent the model from being misled by local noise. In the FHR branch, the frequency per bin is denoted as B_f, and the number of bins is set to $\frac{HW}{B}$. Fine-grained information is extracted from bins with fewer spatially adjacent pixels, focusing on a limited number of pixels to enhance the perception of detailed features and help identify details obscured by noise. The query-key features from both branches are reshaped accordingly and passed through a self-attention mechanism to capture long-range spatial dependencies and distinguish background from noise. The resulting attention maps A_b and A_f are multiplied element-wise to obtain the final attention map A, as shown in Eq. (3).

$$A_{i\ \in\ (b,f)}\ = Softmax\left(\frac{R_i\ (\ Q_1)\ R_i(K_1)^T}{\sqrt{k}}\right) R_i\ (V)\ , \\ A\ =\ A_b\ \odot\ A_f \tag{3}$$

Here, k represents the number of self - attention heads. $\mathrm{R}_{\mathrm{i}\in(\mathrm{b,f})}$ stands for the reshaping methods BHR or FHR. $\mathrm{A}_{\mathrm{i}\in(\mathrm{b,f})}$ denotes the attention maps obtained from the branches. A is the fusion result of the branch attention maps, and $\odot$ represents element - wise multiplication.

YOLO11-DRPA nect

The original neck network of YOLO11n uses C3k2 and Conv modules (structure in Fig. 2), enabling bidirectional feature transmission[20]. However, its simple summation - based fusion treats all image data equally, mixing valid and invalid information. In track fastener detection, this wastes computational resources and reduces efficiency due to large - sized input images. We thus introduce the BRA module with a sparse attention mechanism to boost the model's real - time performance.

The working process of the BRA module, as depicted in Fig. 4, mainly encompasses region division and input projection, region-to-region routing, and token-to-token attention. For an input image $\mathrm{F} \in \mathrm{R}^{H\ \times\ W\ \times\ C}$, region division splits it into $\mathrm{S} \times \mathrm{S}$ non-overlapping regions. Different spatial scales such as 4×4, 7×7, and 14×14 are commonly used for region division. Smaller scales provide more detailed local information, but an excessive number of regions increases routing overhead and may neglect long-range dependencies. Larger scales reduce computational load, yet information redundancy within regions may weaken feature representation. Therefore, a 7×7 size is typically chosen to balance local details and global context. Following this, the image data of each region $\mathrm{F}'\ = \mathrm{R}^{\frac{\mathrm{H}}{\mathrm{S}} \times \frac{\mathrm{W}}{\mathrm{S}} \times \mathrm{C}}$ undergoes input projection using the projection matrices $\omega_0^{\mathrm{V}}\ {}_{S\ \times\ S}$, $\omega_0^{\mathrm{K}}\ {}_{S \times S}$, and $\omega_0^{\mathrm{Q}}\ {}_{S\ \times S}$ to generate value, key, and query tensors $\mathrm{V}\ \in \mathrm{R}^{S\ \times\ S}$, $\mathrm{K}\ \in \mathrm{R}^{S\ \times\ S}$, and $\mathrm{Q}\ \in \mathrm{R}^{S\ \times\ S}$. Subsequently, for region-to-region routing, a region affinity matrix is calculated and pruned to retain only the top k connections of each region, adaptively determining the regions of interest and routing indices, which significantly cuts down computation compared to traditional routing methods. Finally, in the token-to-token attention stage, key-value pairs from regions with higher affinity are collected and fused, yielding tensors $\mathrm{V}^{\mathrm{g}} \in \mathrm{R}^{\frac{\kappa \mathrm{H}\ \mathrm{W}}{\mathrm{S}^2}}$, $\mathrm{K}^{\mathrm{g}} \in \mathrm{R}^{\frac{\kappa \mathrm{H}\ \mathrm{W}}{\mathrm{S}^2}}$, and $\mathrm{Q}\ \in \mathrm{R}^{S\ \times\ S}$ to effectively tackle the issues of low-resolution targets and spatial ambiguity in track fastener detection.

By collecting key-value pairs from regions with high affinity and filtering out most irrelevant ones at the coarse-grained region level, fine-grained token-to-token attention calculation is performed only within a few relevant regions. This enables the model to precisely focus on semantically relevant key-value pairs, effectively addressing the challenges of low target resolution ratios and spatial ambiguity in fastener detection engineering.

To address the issue that key information of small fastener targets is likely to be lost during multiple downsampling operations[21], we further introduce the PPA module into the original neck network. The PPA

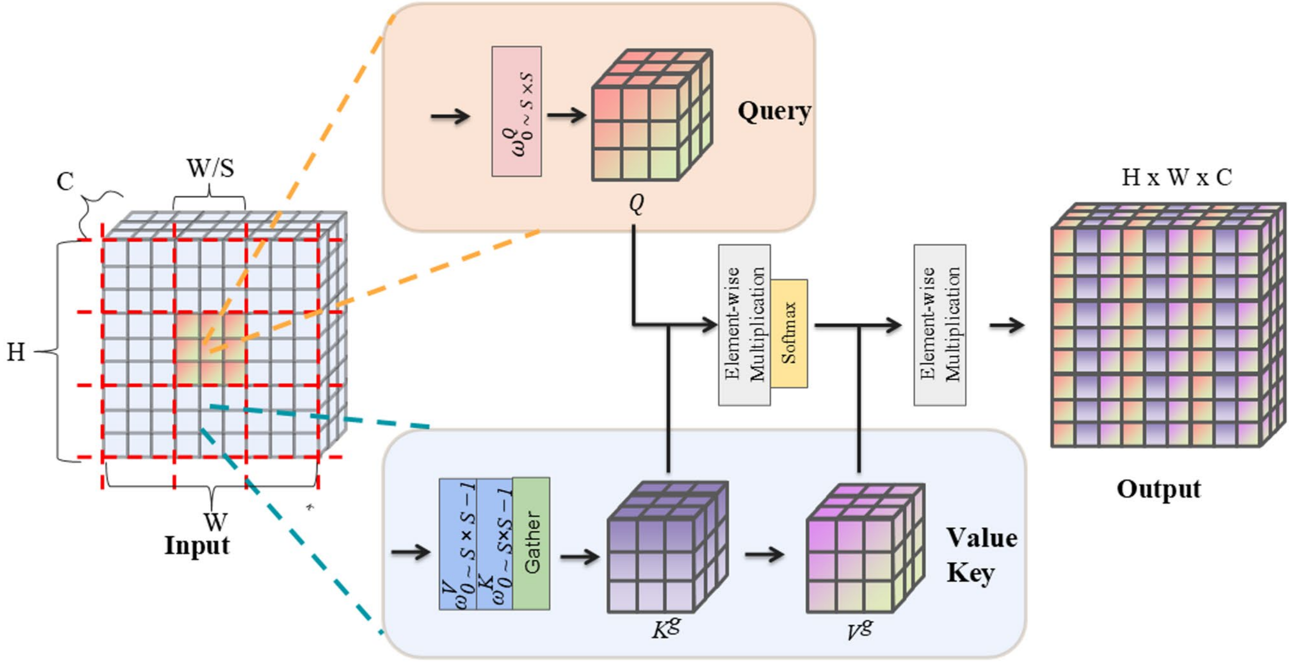
Query
$\omega_0^Q \sim S \times S$
Q
C
W/S
H
W
Input
Element-wise Multiplication
Softmax
Element-wise Multiplication
H x W x C
Output
$\omega_0^V \sim S \times S-1$
$\omega_0^K \sim S \times S-1$
Gather
Value
Key
K^g
V^g

Fig. 4. The working process of the BRA module. The figure shows the specific details of the BRA module, which retains relevant regions, filters out irrelevant regions, and incorporates an attention mechanism.

module employs a multi-branch parallel feature extraction method at different levels, and the complete network is shown in Fig. 5.

The overall workflow is as follows: For the input feature tensor $\mathrm{F} \in \mathrm{R}^{H\times W\times C}$, three parallel branches are constructed via point convolution adjustment to capture local features, global features, and perform multi-level convolution, respectively. In the local and global branches, feature extraction at different scales is achieved by controlling the patch size parameter p, involving operations such as partitioning F, channel averaging, linear computation, activation function processing, and feature selection. The serial convolution branch employs three $3\ \times 3$ convolutional layers instead of traditional convolutions to generate outputs. The resulting three-branch feature tensors $\mathrm{F}_{\mathrm{local}} \in \mathrm{R}^{H\times W\times C}$, $\mathrm{F}_{\mathrm{global}} \in \mathrm{R}^{H\times W\times C}$, and $\mathrm{F}_{\mathrm{conv}} \in \mathrm{R}^{H\times W\times C}$ are summed to form the intermediate feature tensor $\mathrm{F}'\ \in \mathrm{R}^{H\times W\times C}$. Finally, unlike the traditional FPN(Feature Pyramid Network) structure that simply fuses features across layers, the PPA module enhances the summed result with dedicated attention mechanisms for adaptive feature enhancement, emphasizing features relevant to small targets. The attention mechanism operates as follows: F' first passes through the Channel Attention Module (CAM) to generate a one-dimensional channel attention map, denoted as $\mathrm{M}_{\mathrm{c}}\left(\mathrm{F}'\right)$. This map is then multiplied element-wise with F' to produce the channel attention tensor F'', as shown in Eq. (4). Subsequently, F'' serves as the input to the Spatial Attention Module (SAM). After processing through SAM, a two-dimensional spatial attention map $\mathrm{M}_{\mathrm{s}}\left(\mathrm{F}''\right)$ is obtained. This map is multiplied element-wise with F'' to yield the final output feature F_{o}, as expressed in Eq. (5).

The PPA module captures multi-scale features of objects through a multi-branch feature extraction strategy and enhances these features using dedicated attention mechanisms. This ensures that the information of small fastener objects is better preserved and enhanced within the network, thereby improving the accuracy of defect detection for track fasteners.

$$F''' = M_c\left(F''\right) \otimes F'' \tag{4}$$

$$F_o = M_s\left(F'''\right) \otimes F''' \tag{5}$$

$$M_c\left(F''\right) = \sigma\left(MLP\left(AvgPool\left(F''\right)\right)\right) + MLP\left(MaxPool\left(F''\right)\right) \tag{6}$$

$$M_s\left(F'''\right) = \sigma\left(\int^{7\times 7}\left(\left[AvgPool\left(F'''\right); MaxPool\left(F'''\right)\right]\right)\right) \tag{7}$$

Here, F'' represents the image data output by the addition of parallel branches. $\mathrm{M}_{\mathrm{c}} \in \mathrm{R}^{1\times 1\times \mathrm{C}'}$ represents the operation of generating a one - dimensional channel attention map through the CAM[22]. $\mathrm{M}_{\mathrm{s}} \in \mathrm{R}^{\mathrm{H}'\times \mathrm{W}'\times 1}$ represents the operation of generating a two - dimensional spatial attention map through the SAM[23]. $\otimes$ represents the element - wise multiplication calculation.

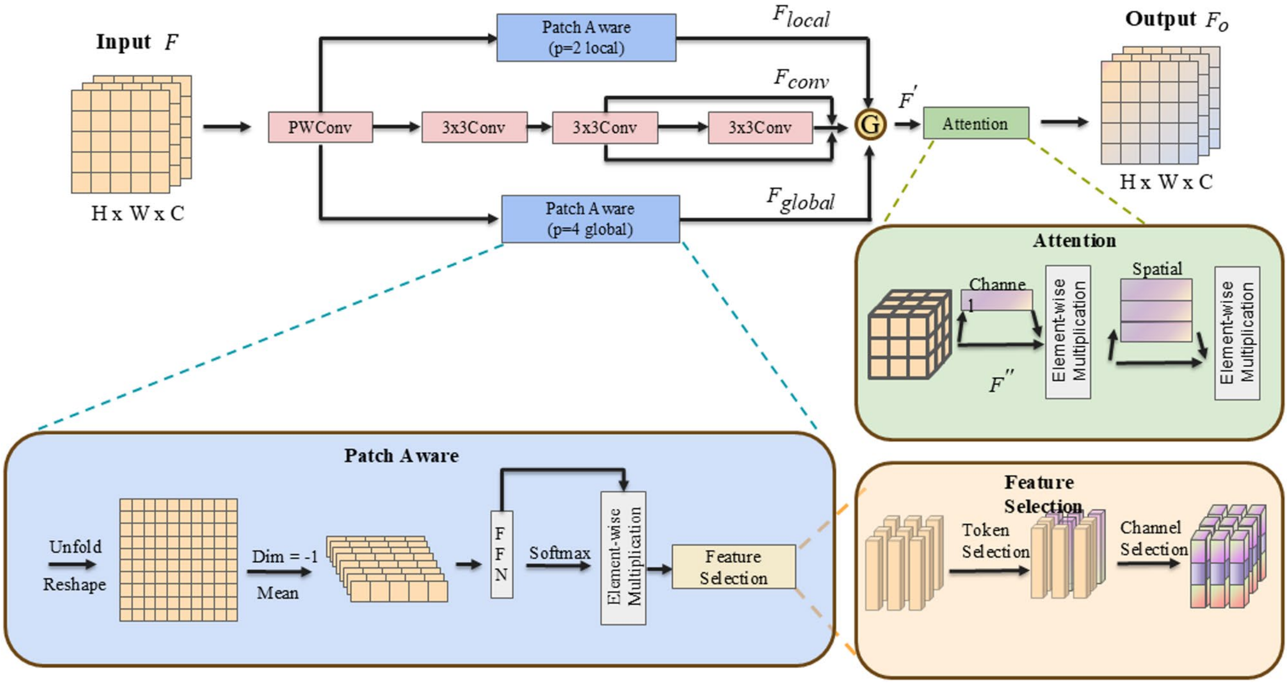

Fig. 5. The complete network structure of PPA module. The figure illustrates the detailed workflow of the PPA module, where “G” represents “Gather” and “FFN” represents “MLP + LayNorm + MLP”.

Loss function

YOLO11 - DRPA adopts a comprehensive loss function[24], which includes bounding box regression loss, classification loss, and confidence loss. As shown in Eq. (8).

$$Loss = Box_{loss} + Cls_{loss} + Dfl_{loss} \tag{8}$$

Here, Loss represents the total loss function. $\mathrm{Box_{loss}}$ denotes the bounding box regression loss, $\mathrm{Cls_{loss}}$ represents the classification loss, and $\mathrm{Dfl_{loss}}$ indicates the confidence loss.

Bounding box regression loss optimizes the difference between predicted and ground - truth bounding boxes. It has two components: center point coordinate loss and width - height loss, with the principle in Eq. (9).

$$\begin{aligned} Box_{loss} = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(\left(x_i - x_i' \right)^2 + \left(y_i - y_i' \right)^2 \right) \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(\left(\sqrt{w_i} - \sqrt{w_i'} \right)^2 + \left(\sqrt{h_i} - \sqrt{h_i'} \right)^2 \right) \end{aligned} \tag{9}$$

where, S is the grid size, B is the number of bounding boxes per grid cell. $1_{\mathrm{ij}}^{\mathrm{obj}}$ indicates if the j - th box in the i - th cell predicts an object. x and y are are box center coords, w and h are its width and height, and λ_{coord} is a weight coefficient for balancing losses.

The classification loss is typically calculated using the cross - entropy loss. It is used to measure the difference between the probability distribution predicted by the model and the true labels, aiming to determine which category the target belongs to. The specific form is shown in Eq. (10).

$$Cls_{loss} = \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} \left(p_i(c) - p_i'(c) \right)^2 \tag{10}$$

$$Dfl_{loss} = - \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \left(\alpha \left(1 - p_{ic} \right)^{\gamma} \log \left(p_{ic} \right) + \left(1 - \alpha \right) p_{ic}^{\gamma} \log \left(1 - p_{ic} \right) \right) \tag{11}$$

Here, $\mathrm{p_i(c)}$ is the probability that the model predicts the target in the *i* - th grid cell belongs to class *c*, and $\mathrm{p_i'(c)}$ is the true label, indicating whether the target in the *i* - th grid cell belongs to class *c*. $\mathrm{y_{ic}}$ is the true label of the *i* - th sample, and $\mathrm{p_{ic}}$ is the probability that the *i* - th sample belongs to class *c*. α is the balancing factor used to adjust the weights between positive and negative samples, and γ is the focusing parameter used to control the degree of attention to difficult samples.

The confidence loss function is often combined with traditional cross - entropy loss to boost the model's learning of hard samples. Its core is to weight category prediction probabilities, making the model focus more on misclassifiable samples. The specific formula is in Eq. (11).

Results

Evaluation method

In our study, to rigorously evaluate the model's performance, we sampled 50 datasets for each experiment and conducted statistical analysis after removing the highest and lowest values. The metrics employed include Precision (P), Recall (R), Average Precision (AP), mean Average Precision (mAP)[25], Standard Deviation (SD), model size parameters, and Frames Per Second (FPS). P represents the accuracy of target recognition, R denotes the recall rate, AP is the average precision calculated as the area under the Precision-Recall curve for each class, and mAP is the mean of individual APs across all classes. SD measures the standard deviation of mAP values. Additionally, we used a representative GPU device, the NVIDIA GeForce RTX 3060 GPU (NVIDIA, Santa Clara, CA, USA), which aligns with the computational capabilities of modern edge devices, to assess model size and FPS, thereby quantifying the model's lightweight characteristics for downstream applications such as embedded systems. The mathematical formulations of P, R, AP, and mAP are presented in Equations (12) to (15).

$$P = \frac{TP}{TP + FP} \tag{12}$$

$$R = \frac{TP}{TP + FN} \tag{13}$$

$$AP = \int_0^1 P(R)\, dR \tag{14}$$

$$mAP = \frac{\sum_{n=1}^{C} AP_n}{C} \tag{15}$$

Here, TP represents the correctly predicted true positive instances, FP represents the false positive instances with inaccurate predictions, FN represents the falsely predicted false negative instances, and C represents the classes in the dataset.

Ablation and comparative experiment

In this stage, we first carried out a series of ablation experiments by combining various modules differently to verify the collaborative optimization among them. The experimental results show that whether added individually or in combination, each module can improve the model's performance in different dimensions to some extent. For example, the PPA module optimizes the problem of information loss during traditional multiple downsampling through parallel multi-branch feature extraction. The BRA module introduces a sparse attention mechanism, constructs a region-level affinity graph, and performs pruning to improve the efficiency of attention calculation. The DHSA module convolves pixel features of images within a dynamic range to extract large-scale and fine-grained information and fuses the outputs. All these modules enable the model to extract and retain rich spatial features, effectively improving the model's detection accuracy. However, there are still some deficiencies in the engineering of fastener detection. The detailed experimental data are presented in Table 1.

Combination of single modules. The PPA module has the greatest improvement on the model precision P, increasing by 2.6% to reach 97.0%. However, the recognition accuracy for the target objects of Rotation and Lack decreases to 97.6% and 73.2%, respectively. In contrast, the BRA module has almost no improvement on the model precision P, but it enhances the recognition accuracy for various categories. Specifically, the recognition accuracy for the Lack category increases to 76.8%. Additionally, the DHSA module reduces the overall precision P, but it significantly improves the mean Average Precision (mAP) in the range of 0.5 to 0.95 for target recognition, reaching 80.1%, thus enhancing the confidence in the detection of fastener targets. In conclusion, the three modules all have a certain promoting effect on YOLO11, but they also have their own limitations.

For the fusion of dual modules, the combination of the PPA module and the BRA module further improves the model precision to 98.6%. It also enhances the recognition accuracy for Lack instances, reaching 76.5%. However, the average confidence of the fastener detection results decreases. The combinations of the other two groups also yield results similar to the above, indicating that the simple stacking of modules does not always improve the model performance, and an organic combination is required.

Finally, the joint effect of the three modules achieves the overall best performance. Although the standard deviation of each group of data is similar, and the improvement effect of our model in terms of overall recognition accuracy is relatively low, it reaches the best upper level in the recognition accuracy of each category. Moreover, it has the highest improvement in mAP0.5 and mAP0.5:0.95, increasing by 1.8% and 4.0% respectively compared with the original YOLO11 network, reaching 94.6% and 80.7%. While increasing the model size slightly, it improves the comprehensive performance of the model. The specific detection results are shown in Fig. 6.

Although the improved model has a decrease of 9.7 in terms of FPS, it reaches 38.5 and still meets the work requirements in the fastener detection project[26]. Moreover, in the fastener detection project, higher precision can better identify potential risks. In addition, with the enhancement of the computing power of edge devices, it is sufficient to optimize the calculation speed of the model.

To verify the performance and generalization of YOLO11-DRPA, we tested it on the training dataset and an unseen one, comparing it with Rtdetr[27], PP-YOLOE-S[28], and YOLOX variants, which are current mainstream object detection models. On the training dataset, YOLO11-DRPA achieved 95.4% accuracy and 93.5% regression rate, with a notable boost in regression rate over other models. It excelled in detecting rotation, normal, and damaged fasteners, and despite slightly lower performance in missing fastener detection than YOLO10n, it outperformed YOLO10n in regression rate and mAP0.5:0.95 by 3.2% and 3.4%, respectively, indicating higher stability. In the external dataset, its mAP0.5 reached 86.5%, the highest among all models, demonstrating strong generalization. The detailed experimental data are presented in Table 2.

Notwithstanding a marginal increment in model size and computational parameters, YOLO11-DRPA appears to yield discernible improvements in detection accuracy. Collectively, the results of this study indicate that the proposed YOLO11-DRPA model could potentially offer enhanced precision and comprehensive detection performance for railway track fastener inspection. It is hoped that this research can serve as a meaningful contribution to the ongoing efforts in advancing the methodologies within this specialized domain.

Conclusions

In this study, we propose the YOLO11-DRPA detection algorithm, an enhancement based on YOLO11n. By introducing the DHSA module into the backbone network, the model effectively fuses multi-scale image features and local-global information, improving its robustness against adverse weather conditions and environmental noise. The BRA module is integrated before the upsampling operation in the neck network, utilizing spatial scale segmentation and a regional self-attention mechanism to enhance operational efficiency. The PPA module is gradually added within the neck network. By leveraging weighted image attention in parallel branches, this module strengthens the model's ability to detect small targets. Ablation experiments show that the synergistic integration of these modules achieves an optimal balance in network structure, leading to outstanding performance in the evaluation metrics of mAP@0.5 and mAP@0.5:0.95. Comparative experiments with leading detection models demonstrate that, despite a moderate increase in model size, the proposed approach results in significant improvements in detection accuracy and stability. These findings suggest that the model strikes a promising balance between complexity and recognition performance when compared to similar algorithms.

Experimental results on the railway track fastener dataset show that the YOLO11-DRPA algorithm effectively addresses the challenges of improving both defect detection efficiency[29] and accuracy[30]. While the model size increases moderately, future research will focus on optimizing model parameters to reduce complexity, computational costs, and enhance operational efficiency, all while maintaining detection accuracy and stability.

Model	*P* (%)	*R* (%)	AP					mAP0.5:E0.95 (%)	Model size (M)	FPS	SDmAP 0.5 (10^{-16})	SDmAP 0.5:0.95 (10^{-16})
			Normal (%)	Rotation (%)	Lack (%)	Break (%)	mAP0.5 (%)					
YOLO11n	94.4	90.5	98.9	98.3	76.4	97.9	92.8	76.7	10.6	48.2	1.09	1.10
YOLO11 + PPA	97.0	91.1	99.4	97.6	73.2	99.5	92.5	78.6	13.5	31.2	1.11	1.12
YOLO11 + BRA	94.4	93.9	99.2	99.5	76.8	99.5	93.5	79.5	11.7	53.4	1.09	1.11
YOLO11 + DHSA	87.5	91.5	99.4	99.5	69.6	98.4	91.7	80.1	14.2	47.6	1.13	1.11
YOLO11 + PPA + BRA	98.6	81.7	99.0	98.3	76.5	96.0	92.2	76.7	14.6	36.0	1.00	1.08
YOLO11 + PPA + DHSA	96.9	92.3	99.3	99.5	77.9	99.5	94.1	78.0	17.2	37.3	1.18	1.18
YOLO11 + BRA + DHSA	96.2	93.5	99.3	99.5	79.0	98.4	93.6	78.3	15.3	50.5	1.13	1.11
YOLO-DRPA	95.4	93.5	99.4	99.5	79.0	99.5	94.6	80.7	18.2	38.5	1.03	1.02

Table 1. Ablation experiments on DHSA, BRA, and PPA modules.

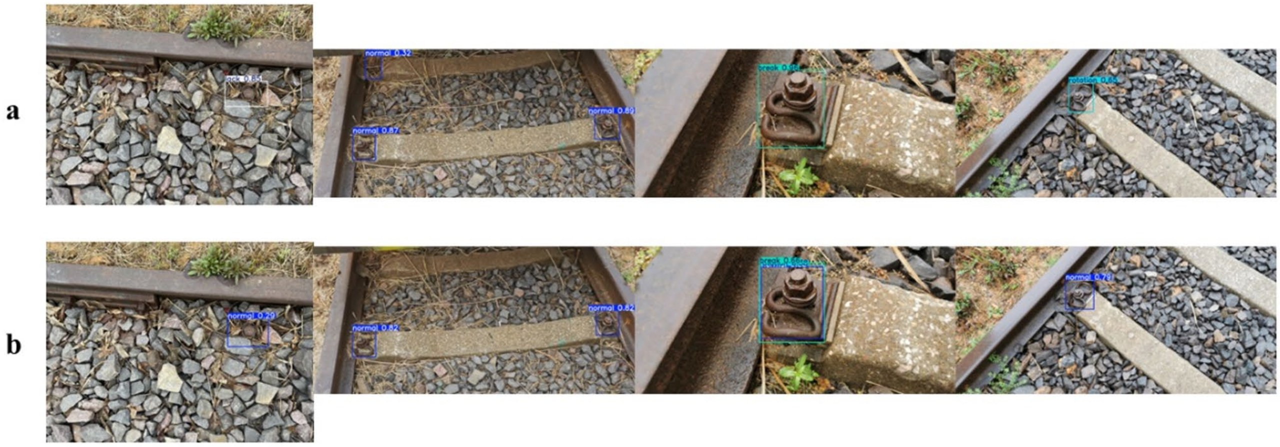

Fig. 6. Verification of the model's effectiveness. Group (**a**) is YOLO11-GRPA, and group (**b**) is YOLO11. The actual targets from left to right are Lack, Normal, Break, and Rotation.

These improvements are aimed at enabling the efficient deployment of the algorithm on resource-constrained embedded devices.

Model	*P* (%)	*R* (%)	AP				mAP 0.5(%)	mAP 0.5:0.95 (%)	Model size(M)	FPS	SDmAP 0.5 (10^{-16})	SDmAP 0.5:0.95 (10^{-16})
			Normal (%)	Rotation (%)	Lack (%)	Break (%)						
Railway track fastening dataset												
PP-YOLOE-S	94.8	90.0	99.4	98.9	77.3	97.8	93.7	78.5	30.6	55.4	1.08	1.03
Rtdetr- resent50	83.7	66.4	98.2	62.0	63.9	47.6	67.9	48.4	171.3	27.5	1.12	1.13
Rtdetr-l	78.5	52.7	96.7	55.6	39.5	39.8	58.4	45.2	131.5	27.0	1.14	1.13
YOLO3n	94.6	88.6	99.4	98.8	78.4	99.5	93.9	79.6	414.9	41.0	1.03	1.07
YOLO5n	96.1	91.7	99.3	99.1	77.9	99.5	93.9	78.6	10.3	58.9	1.06	1.06
YOLO6n	96.0	88.7	99.3	97.6	77.4	96.4	92.7	75.6	17.1	64.5	1.03	1.05
YOLO8n	88.0	91.5	99.4	99.1	77.1	97.6	93.3	78.2	12.3	63.8	1.07	1.02
YOLO10n	96.5	90.3	99.2	98.2	79.3	99.5	94.1	77.3	9.3	55.1	1.05	1.01
YOLO11n	94.4	90.5	98.9	98.3	76.4	97.9	92.8	76.7	10.6	48.2	1.09	1.10
YOLO-DRPA	95.4	93.5	99.4	99.5	79.0	99.5	94.6	80.7	18.2	38.5	1.03	1.02
External railway track fastening dataset												
PP-YOLOE-S	92.8	89.9	98.0	70.1	77.2	96.1	85.8	71.2	30.6	55.4	1.07	1.06
Rtdetr-resent50	80.7	60.6	95.7	39.1	61.5	46.2	61.5	43.8	171.3	27.5	1.13	1.12
Rtdetr-l	73.9	39.9	93.0	25.7	29.5	31.2	44.9	31.7	131.5	27.0	1.14	1.15
YOLO3n	92.9	80.6	97.9	69.8	78.6	99.3	86.1	72.9	414.9	41.0	1.03	1.08
YOLO5n	93.8	83.8	98.4	70.7	77.9	99.4	86.2	71.7	10.3	58.9	1.05	1.06
YOLO6n	93.9	80.0	98.4	67.9	76.7	96.0	84.7	68.9	17.1	64.5	1.03	1.04
YOLO8n	90.5	78.4	97.9	69.3	77.3	95.8	85.3	70.8	12.3	63.8	1.08	1.02
YOLO10n	96.1	83.0	98.4	70.4	78.0	99.3	86.3	70.7	9.3	55.1	1.04	1.02
YOLO11n	92.1	83.3	98.2	70.6	76.7	95.9	85.4	71.4	10.6	48.2	1.09	1.11
YOLO-DRPA	95.5	87.6	98.4	74.3	78.2	99.4	86.5	73.2	18.2	38.5	1.05	1.02

Table 2. Comparative experiment on the performance of different deep learning algorithms.

Data availability

The data sets generated and analyzed during the current study are not publicly available. Due to the confidentiality agreements with our collaborating partners, sharing these data is restricted. However, further information or additional details may be provided upon reasonable request to the corresponding author, subject to approval from the collaborating partners and compliance with the terms of the confidentiality agreements.

Received: 16 April 2025; Accepted: 24 July 2025
Published online: 29 July 2025

References

1. Poveda, E., Rena, C. Y., Lancha, J. C. & Ruiz, G. A numerical study on the fatigue life design of concrete slabs for railway tracks. *Eng. Struct.* **100**, 455–467 (2015).
2. Resendiz, E., Hart, J. M. & Ahuja, N. Automated visual inspection of railroad tracks. *IEEE Trans. Intell. Transp. Syst.* **14** (2), 751–760 (2013).
3. Cheung, W. & Hamarneh, G. N-sift: N-dimensional scale invariant feature transform for matching medical images. In *2007 4th IEEE international symposium on biomedical imaging: from nano to macro* (pp. 720–723). IEEE. (2007), April.
4. Bay, H., Tuytelaars, T. & Van Gool, L. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I 9* (pp. 404–417). Springer Berlin Heidelberg. (2006).
5. Dalal, N. & Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886–893). Ieee. (2005), June.
6. O'shea, K. & Nash, R. An introduction to convolutional neural networks. arxiv preprint arxiv:1511.08458. (2015).
7. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **38** (1), 142–158 (2015).
8. Siradjuddin, I. A. & Muntasa, A. Faster region-based convolutional neural network for mask face detection. In *2021 5th international conference on informatics and computational sciences (ICICoS)* (pp. 282–286). IEEE. (2021).
9. Jiang, P., Ergu, D., Liu, F., Cai, Y. & Ma, B. A review of Yolo algorithm developments. *Proc. Comput. Sci.* **199**, 1066–1073 (2022).
10. Liu, W. et al. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21–37). Springer International Publishing. (2016).
11. Feng Guo, Y. & Qian Yuefeng Shi.Real-time railroad track components inspection based on the improved YOLOv4 framework. *Autom. Constr.* **125**, 10359 (2021).
12. Feng Guo, J., Liu, Y. & Qian Quanyi xie.rail surface defect detection using a transformer-based network. *J. Ind. Inform. Integr.* **38**, 100584 (2024).
13. Abdu, F. J., Zhang, Y., Fu, M., Li, Y. & Deng, Z. Application of deep learning on millimeter-wave radar signals: A review. *Sensors 21*(6), 1951. (2021).
14. Zhou, F., Fu, Z. & Zhang, D. High dynamic range imaging with context-aware transformer. In *2023 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE. (2023), June.
15. Zhu, L., Wang, X., Ke, Z., Zhang, W. & Lau, R. W. Biformer: Vision transformer with bi-level routing attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10323–10333). (2023).
16. Xu, S. et al. Hcf-net: Hierarchical context fusion network for infrared small object detection. In *2024 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1–6). IEEE. (2024), July.
17. Buslaev, A. et al. Albumentations: fast and flexible image augmentations. *Information* **11** (2), 125 (2020).
18. Zhang, C. et al. The WuC-Adam algorithm based on joint improvement of warmup and cosine annealing algorithms. *Math. Biosci. Eng.* **21**, 1270–1285 (2024).
19. Gupta, H., Kotlyar, O., Andreasson, H. & Lilienthal, A. J. Robust object detection in challenging weather conditions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 7523–7532). (2024).
20. Lavin, A. On the Efficiency of Convolutional Neural Networks. arxiv preprint arxiv:2404.03617. (2024).
21. Dumitrescu, D. & Boiangiu, C. A. A study of image upsampling and downsampling filters. *Computers* **8** (2), 30 (2019).
22. Wang, Q. et al. ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11534–11542). (2020).
23. Zhu, X., Cheng, D., Zhang, Z., Lin, S. & Dai, J. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6688–6697). (2019).
24. He, Y. et al. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 2888–2897). (2019).
25. Kyriakis, J. M. et al. Raf-1 activates MAP kinase-kinase. *Nature* **358** (6385), 417–421 (1992).
26. Song, Q., Guo, Y., Jiang, J., Liu, C. & Hu, M. High-speed Railway Fastener Detection and Localization Method based on convolutional neural network. arxiv preprint arxiv:1907.01141. (2019).
27. Wang, S. et al. Phsi - rtdetr: A lightweight infrared small target detection algorithm based on UAV aerial photography. *Drones* **8** (6), 240 (2024).
28. Xu, S. et al. PP-YOLOE: An evolved version of YOLO. arxiv preprint arxiv:2203.16250. (2022).
29. Feng, H. et al. Automatic fastener classification and defect detection in vision-based railway inspection systems. *IEEE Trans. Instrum. Meas.* **63** (4), 877–888 (2013).
30. Wei, X. et al. Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. *Eng. Appl. Artif. Intell.* **80**, 66–81 (2019).

Author contributions

Chengwei Zhang completed the vast majority of the research. Jiawei Zhu and Yihao Ma created the experimental dataset. Qingmei Huang finished the layout work of the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to C.Z.

Reprints and permissions information is available at www.nature.com/reprints.