









Research Article

Crack Detection Method of Sleeper Based on Cascade Convolutional Neural Network

Liming Li ^{1,2,3} Shubin Zheng ^{1,3} Chenxi Wang ¹ Shuguang Zhao ²
Xiaodong Chai ^{1,3} Lele Peng ^{1,3} Qianqian Tong ¹ and Ji Wang ¹

¹School of Urban Railway Transportation, Shanghai University of Engineering Science, Shanghai 201620, China

²School of Information Science and Technology, Donghua University, Shanghai 201620, China

³Shanghai Engineering Research Center of Vibration and Noise Control Technologies for Rail Transit, Shanghai University of Engineering Science, Shanghai 201620, China

Correspondence should be addressed to Shubin Zheng; shubin.zheng@sues.edu.cn

Received 30 July 2021; Accepted 16 December 2021; Published 11 January 2022

Academic Editor: Seyed Ali Ghahari

Copyright © 2022 Liming Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents a new method for sleeper crack identification based on cascade convolutional neural network (CNN) to address the problem of low efficiency and poor accuracy in the traditional detection method of sleeper crack identification. The proposed algorithm mainly includes improved You Only Look Once version 3 (YOLOv3) and the crack recognition network, where the crack recognition network includes two modules, the crack encoder-decoder network (CEDNet) and the crack residual refinement network (CRRNet). The improved YOLOv3 network is used to identify and locate cracks on sleepers and segment them after the sleeper on the ballast bed is extracted by using the gray projection method. The sleeper is inputted into CEDNet for crack feature extraction to predict the coarse crack saliency map. The prediction graph is inputted into CRRNet to improve its edge information and local region to achieve optimization. The accuracy of the crack identification model is improved by using a mixed loss function of binary cross-entropy (BCE), structural similarity index measure (SSIM), and intersection over union (IOU). Results show that this method can accurately detect the sleeper crack image. During object detection, the proposed method is compared with YOLOv3 in terms of directly locating sleeper cracks. It has an accuracy of 96.3%, a recall rate of 91.2%, a mean average precision (mAP) of 91.5%, and frames per second (FPS) of 76.6/s. In the crack extraction part, the F-weighted is 0.831, mean absolute error (MAE) is 0.0157, and area under the curve (AUC) is 0.9453. The proposed method has better recognition, higher efficiency, and robustness compared with the other network models.

1. Introduction

China's total railroad mileage is expected to exceed 128,000 km by the end of 2020, prompting researchers to improve maintenance techniques for railroad infrastructure [1]. In Figure 1, the sleeper is used to support the rail and transfer the huge impact brought by the train to the roadbed. Accordingly, the sleeper needs to have a certain degree of flexibility and can be slightly deformed to cushion the pressure. However, the cracks and other damage generated within it will undermine the integrity of the sleeper and diminish the support force provided by the sleeper to the train above when the load bending moment is greater than the cracking strength. This situation poses a safety hazard to

trains passing at a high speed. In recent years, nondestructive testing techniques, such as those in the literature [2], have been widely used in the maintenance of track facilities. This method of sleeper cracking can be quick and efficient in preventing accidents.

At present, the main method of sleeper crack detection has shifted from manual identification to a series of physical detection means, such as ultrasonic, eddy current detection, and ray detection. Although this method has been developed, it still has the limitations of the use methods and the common problem of poor crack detection. The efficiency and accuracy of crack detection have been enhanced with the development of the computer vision technology. The main methods applied to this field are as follows: image

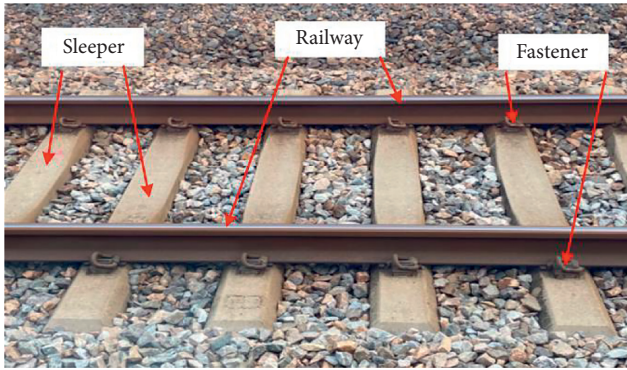


FIGURE 1: Railway track line.

processing-based methods [3], machine learning-based methods [4], and deep convolutional neural network (DCNN)-based methods [5]. The methods represented by DCNNs are subdivided into methods based on image classification [5], object detection [6], and pixel-level segmentation [7], depending on the way the crack detection problem is handled. The network used to detect cracks in sleepers in this cascade is based on the latter two types of methods.

The main crack detection methods based on object detection include Faster R-CNN [8], single-shot multibox detector (SSD) [9], and You Only Look Once (YOLO) [10] to determine the location of cracks in the input image and localize them with bounding boxes. Cha et al. [11] proposed a concrete crack detection method based on Faster R-CNN. The network is improved to quickly detect and locate multiple types of cracks in real time, allowing for more accurate detection results. Mandal et al. [12] proposed an automated detection method based on DCNNs for road concrete cracks. However, the achieved detection accuracy is low. Li et al. [13] proposed an improved YOLO network to improve the detection accuracy of track plate cracks. However, the method is less versatile due to the single background information of the track plate. Bao et al. [14] proposed a triplet graph reasoning network for the problem of insufficient samples of metal surface defects.

Crack detection methods based on pixel-level segmentation mainly include fully convolutional networks (FCNs) [15], U-Net [16], and Seg-Net [17]. Labels can be assigned to crack pixel points to determine the presence of cracks and to obtain important features, such as the location, size, and shape of cracks. Cheng et al. [18] proposed an automatic U-Net-based road crack detection method and tested it in a crack dataset to obtain a high pixel-level segmentation accuracy. Islam and Kim [19] proposed a full CNN-based concrete crack detection method. This network consisting of encoder and decoder patterns is tested and exhibits good detection results on publicly available crack datasets. Dung [20] designed a full CNN with Visual Geometry Group-16 (VGG-16) based on a codec framework. This network further improves the accuracy of crack detection. Literature [21] compared three U-Net algorithms of different depths for automatic pavement crack detection systems. The objective is to verify whether a model architecture with greater

depth necessarily results in better detection accuracy. Experiments prove that choosing a network architecture with the right depth can guarantee the detection accuracy and improve the detection speed.

Although great progress has been made in the field of crack detection based on DCNNs, how to obtain more detailed crack features still needs to be explored. For the sleeper crack detection, the crack is small, similar to the background of the sleeper, the boundary is unclear, and the regional information is incomplete. This paper proposes a new cascade network for crack detection. YOLOv3 is used as one of the mainstream frameworks for object detection. The YOLO series is improved on the basis of YOLOv3. Given that YOLOv3 uses a residual network in the feature extraction part, three feature layers of different depths are simultaneously extracted, and a stacked stitching approach is used to obtain the prediction results [22]. The aforementioned method can be used to detect cracks of different sizes. However, the crack detection effect is unsatisfactory for the complex background of the rail sleeper. Accordingly, we add the squeeze and excitation (SE) module at the end of the YOLOv3 backbone network to improve the crack region extraction accuracy. Further quantitative parameter detection of cracks is needed to complete high-precision crack identification and provide more scientific detection data. Crack encoder-decoder network (CEDNet) and crack residual refinement network (CRRNet) are used to extract and optimize the features of rail sleeper cracks. The shallow information of the crack image can be passed to the corresponding decoding process after the feature extraction of the input rail cracks by the coding part of CEDNet. Consequently, the low-level detail features are fused with the high-level complex semantics to improve the network feature extraction performance. CRRNet is added because the coarse saliency map obtained in the previous step has deficiencies, such as blurred crack boundaries and missing important regions. CRRNet can be optimized by learning the residuals between the coarse saliency map and the ground truth.

The main contributions of this paper are summarized as follows:

- (1) A two-level cascade network based on DCNN is proposed. This network fuses CEDNet and CRRNet, which can play the role of crack feature extraction and optimization in one step. Its F-weighted is 0.831, mean absolute error (MAE) is 0.0157, and area under the curve (AUC) is 0.9453.
- (2) An improved YOLOv3 network is proposed to localize the cracks, and the attention mechanism, SE module, is added at the end of the backbone network. The mean average precision (mAP) is improved by 6.9% compared with YOLOv3.
- (3) The optimization effects of loss functions binary cross-entropy (BCE), intersection over union (IOU), and structural similarity index measure (SSIM) on crack recognition are superimposed to propose a new hybrid loss function for the crack recognition. Particularly, our method improves F_{weighted} by

68.4%, 74.8%, 84.1%, and 99.0% on $l_{bce} + l_{iou}$, l_{bce} , l_{iou} , and l_{ssim} , respectively.

The rest of this paper is organized as follows: Section 2 introduces the method overview, including the overall steps and the specific theory for each step. Section 3 shows some experimental results of our method and compares them with other methods. Section 4 gives the conclusion and outlook.

2. Method Overview

In the acquired image of rail sleeper cracks, the edge of ballast can interfere with the recognition of rail sleeper cracks because the imaging of ballast and concrete rail sleeper is similar. Given that the edge of the rail sleeper has obvious features, a strict size regulation, and differs from the grayscale of the ballast, the rail sleeper area can be first segmented. The cracks on the rail sleeper can then be located and identified by using the network. The proposed crack detection algorithm is divided into two parts: crack localization and crack identification. The crack recognition part incorporates a feature extraction network and a boundary refinement network. The overall methodological flow is shown in Figure 2. In the first step, we choose the gray projection method to extract the sleeper area first because the large amount of ballast in the background of the sleeper affects the crack detection. In the second step, a modified YOLOv3 is used to locate and segment the cracks on the basis of the extraction of the rail sleeper area. In the third step, further quantitative parameter detection of cracks is needed to complete high-precision crack identification and provide more scientific detection data; hence, CEDNet is used for feature extraction. A boundary refinement network is designed for further optimization because the extracted cracks have partial boundary and region information incompleteness:

- (1) The location of the sleeper is extracted by using the gray projection method [23] combined with the empirical value of the sleeper pixels, and then, SE [24] and spatial pyramid pooling (SPP) [25] are added at the end of the YOLOv3 backbone network to locate the sleeper cracks
- (2) CEDNet, a crack coarse saliency feature extraction network, is used to obtain more detailed saliency information by fusing low-level features and high-level features of crack images through the network structure of codec patterns
- (3) CRRNet, a crack boundary refinement network, is used to learn the residuals between the original and ground truth maps of the crack for optimization purposes by fusing the outputs of the network feature layers

2.1. Crack Location Module. The dimensions are strictly defined, and they differ from the ballast grayscale because the sleeper edge features are obvious. The gray projection method combined with the empirical values of the sleeper

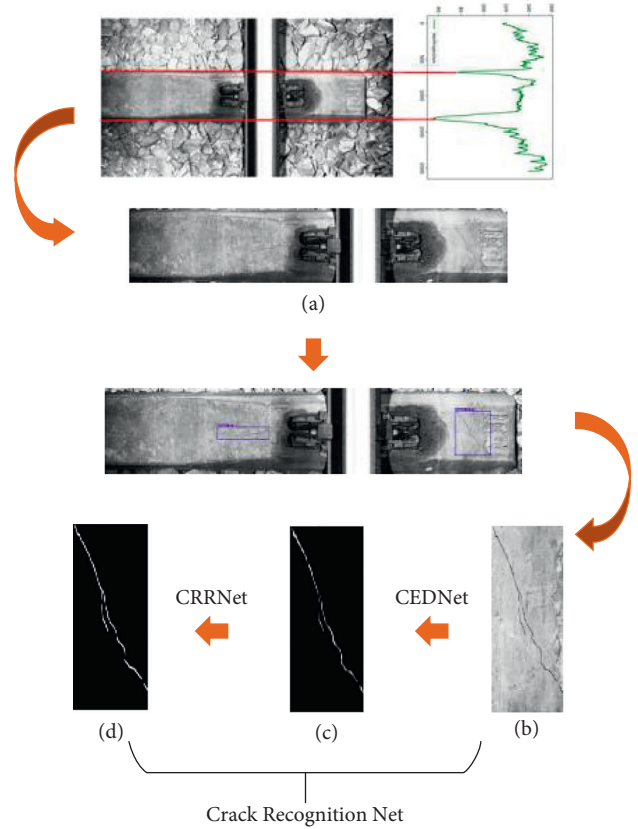


FIGURE 2: Process of the proposed method. (a) Gray projection. (b) Improved YOLOv3. (c) Feature extraction. (d) Edge refinement.

pixels can be used to locate the position of the sleeper. The gray projection method has better results for object edge detection with complex backgrounds, relying mainly on the peaks and valleys in the gray projection curve to determine the coordinates of the object edge position. Assuming that the image is represented as $f(x, y)$, the gray projection function in the x -direction is $f_x(x)$, the coordinates of the pixel points in the image are (x, y) , and the value of the gray projection function in the horizontal direction is

$$f_x(x) = \sum_y f(x, y). \quad (1)$$

The edge coordinates of the horizontal direction of the sleeper can be obtained in accordance with the gray projection method. The pixel width of the edge of the sleeper is relatively fixed in the captured roadbed images. Figure 3(a) shows the original drawing of the ballasted roadbed. The valley of the horizontal projection in Figure 3(b) depicts the contact edge between the sleeper and the ballast. Figure 3(c) presents the segmentation results.

The prediction results are obtained by stacking and splicing after simultaneously extracting three feature layers with different depths because YOLOv3 uses a residual network in the feature extraction part. Therefore, this network can be used to detect cracks of different sizes. However, in the complex background of the sleeper, the crack detection effect is poor. Inspired by the literature [24–26], the SE module

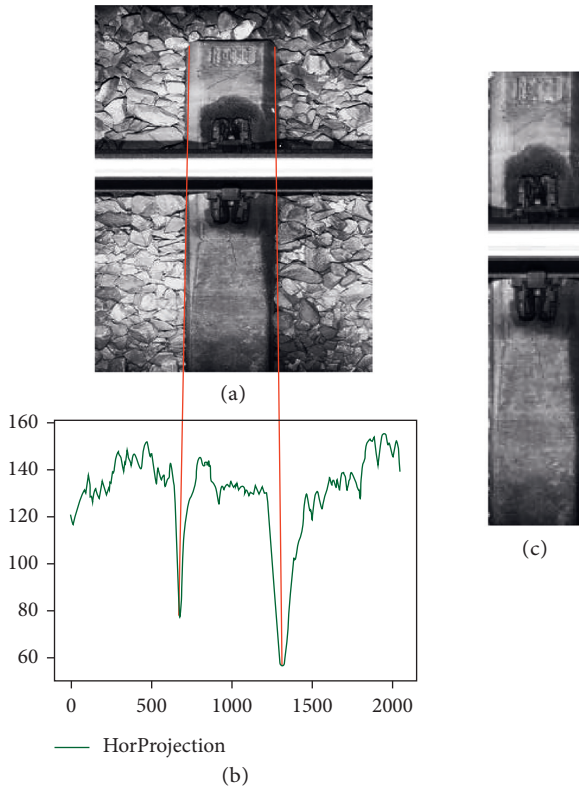


FIGURE 3: Gray projection experimental results. (a) Original image. (b) Horizontal projection. (c) Segmentation results.

suppresses the interference of background and other noises, and the SPP module can improve the operation efficiency by relieving the network of the size requirement for input images while ensuring that the images are not distorted. The end-to-end semisupervised object detection method, the object detection head with unified awareness from the attention perspective, and Composite Backbone Network Version2 (CBNetV2), which eliminates the pretraining process, can avoid the more complex multistage training approach in the literature [27–29]. However, the algorithms in the above documents still have some shortcomings, such as slow detection speed, large consumption of network resources, low accuracy and recall rate, and poor detection accuracy. Therefore, we choose to add SE and SPP modules at the end of the backbone network to make the model simpler in the training process and to improve the accuracy of crack region extraction while minimizing additional overhead. An improved algorithm based on YOLOv3 is designed in this paper, and its overall structure is shown in Figure 4.

The SE module belongs to one of the more classical algorithms of the attention mechanism. The accuracy of crack detection can be significantly improved by designing special parameters capable of removing the invalid information extracted by the YOLOv3 network [25]. This module compresses the sleeper crack image to a size of $1 \times 1 \times 1024$ after a global averaging pooling layer. The activation is performed by two modules in fully connected layers and activation functions. The crack feature channels are weighted uniformly. The designed residual module ensures

effective training so that the network extracts more accurate information about crack features and suppresses interference from other noises in the sleeper images.

When performing prediction of the a priori frame on three scales of the crack image, YOLOv3 requires consistent size of the crack feature maps outputted by the backbone feature extraction network. The cropping or shape change of the image tends to cause partial loss of information, resulting in biased crack detection results. Accordingly, the SPP module is added after the SE module to remove the limitation of the fixed size of the input image [26]. The sleeper crack images outputted from the backbone network of this module are simultaneously pooled at three scales after one convolution operation. The output crack features are fused and inputted to the fully connected layer. We can obtain a fixed size crack image output without losing the original information for any size and scale of the crack image input.

2.2. Crack Recognition Module. After locating and segmenting the cracked area of the rail sleeper, this paper proposes a crack identification module to obtain more detailed crack characteristics. The module uses a crack boundary refinement network to optimize the predicted saliency map because the extracted crack information is incomplete. The final crack saliency map is obtained by fusing the crack boundary refinement network with the feature extraction network, and the general block diagram of this module is shown in Figure 5.

2.2.1. Feature Extraction Module. The backbone network used for feature extraction is the crack coarse saliency feature extraction network CEDNet, which is a codec network focusing on crack regions and boundaries. The network is built on the basis of ResNet-34 (Residual Network with 34 parameter layers) [30] using a codec form. After feature extraction of the input sleeper cracks in the encoding part, the resulting image features are further optimized and processed by the decoding part. The shallow information of the cracked image is passed to the corresponding decoding process, which enables the fusion of low-level detailed features with high-level complex semantics as a method to improve the network feature extraction performance. The structure is shown in Figure 6.

The specific structure and operational steps of the network are as follows:

- (1) The coding part consists of an input convolutional layer and six stages consisting of basic residual blocks, with a modified ResNet-34 structure for the input convolutional layer and the first four convolutional stages. The improvements mainly include the use of a 3×3 convolution filter and a convolution kernel with a stride of 1. The pooling operation is removed after the input convolutional layer to guarantee that the feature map in the first stage has the same spatial resolution as the input image. By contrast, the first feature map in the original ResNet has only one-quarter of the resolution of the input map. This change allows the network to obtain higher resolution feature maps in

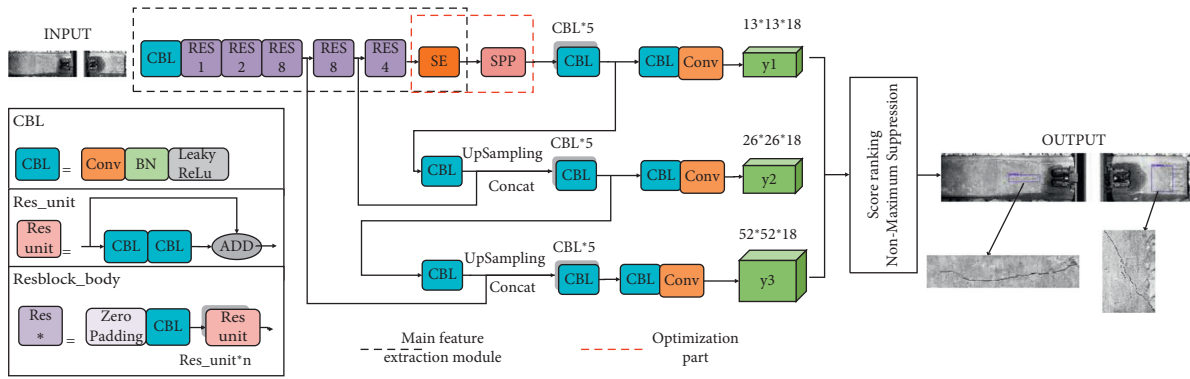


FIGURE 4: Improved YOLOV3 structure process diagram.

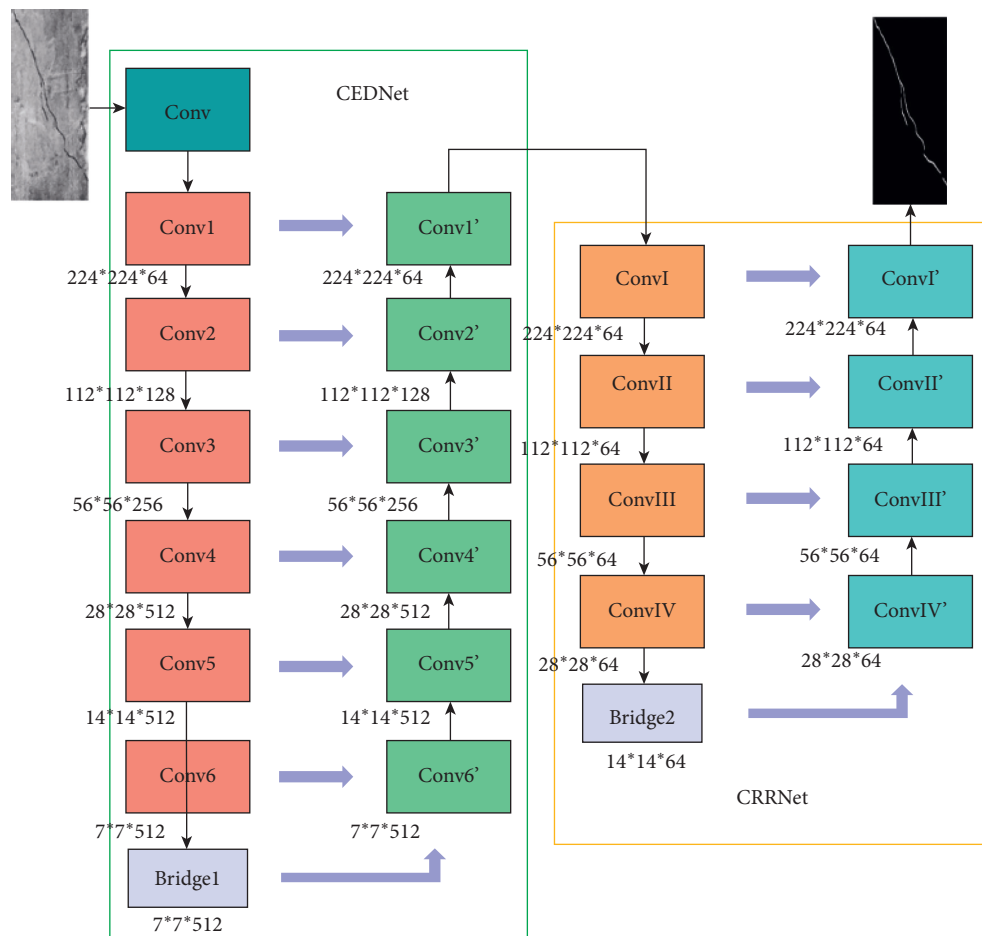


FIGURE 5: Crack recognition module.

previous layers although reducing the overall receptive field. Consequently, Conv5 and Conv6, which are two convolutional stages consisting of 512 filters and three basic residual blocks, are added to obtain a greater extent of the object detection region on the original map and achieve the same receptive field as the original ResNet.

- (2) A bridge connection structure is used to further obtain the global information of cracks. The bridge connection structure contains three modules

consisting of a Conv layer, a batch normalization (BN) layer [31], and a rectified linear unit (ReLU) activation function [32], where each convolutional layer consists of 512 3×3 dilated convolutions [33].

- (3) The input of each level of the decoding section is cascaded from the previous level and the pooled output of the corresponding level in the encoding section. A sigmoid function is added to each layer after using bilinear up-sampling for mapping the predicted values to $[0, 1]$. Seven saliency mappings are generated in this

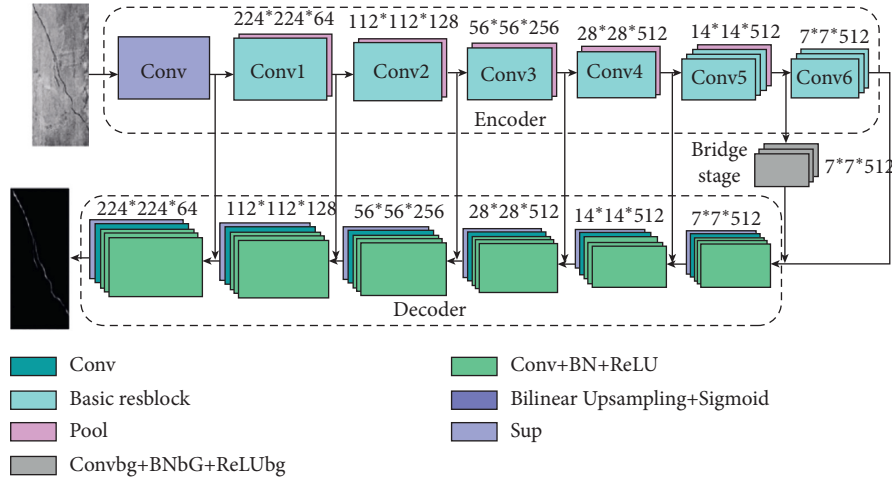


FIGURE 6: CEDNet.

module, containing six postcascade feature mappings and the final output feature mapping. However, only the last feature map with the highest accuracy can be inputted into the CRRNet. The supervision of the ground truth map is supervised at the last layer of each decoding stage to reduce overfitting, as in holistically nested edge detection [34].

2.2.2. Edge Refinement Module. After the object detection and feature extraction, the predicted crack coarse saliency map can be obtained for the sleeper cracks. Figure 7 shows the original map of cracks, the ground truth map, and the coarse saliency map after the CEDNet extraction.

In the coarse saliency map, the crack boundary is blurred, some salient regions are missing, and the background is incorrectly marked as the object and inaccurately located. Therefore, the boundary information and local details of the extracted crack feature map are incomplete. Therefore, the extracted feature map is fed into CRRNet for further optimization.

The network is built in codec form and achieves optimization by learning the residuals between the original and the ground truth maps, using two 1D filters (i.e., 3×1 and 1×3 convolutional layers) rather than of 3×3 in size, which can improve the network optimization performance while avoiding a large computational effort [35]. Coarse feature maps of the input and stacked outputs are fused by using residual module propagation with identity mapping branches to facilitate training, and iterations are conducted to optimize coarse saliency map accuracy. The boundary refinement map under the sigmoid function mapping is used as the final output of the network, as shown in Figure 8.

The network structure consists of three parts: encoder, decoder, and bridge connection.

The coding section consists of four stages with two 1D filters and a maximum pooling layer for down-sampling and reduced computational effort. The order of the built convolutional layers is 3×1 in front and 1×3 convolution in the back. Only one ReLU layer is added after the former, and a BN layer and a ReLU layer are placed after the convolutional layer

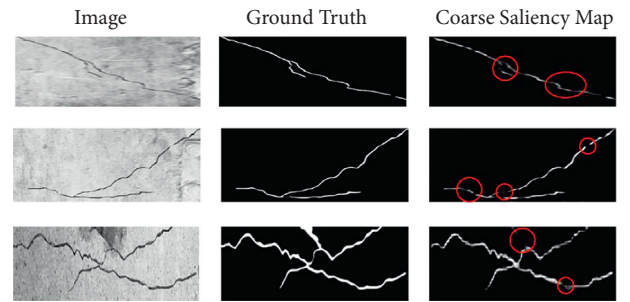


FIGURE 7: Coarse saliency map of crack.

of the latter [36]. This design allows the network to be built to a deeper level with less degradation in performance and mitigates to a certain extent the effect of gradient diffusion on network training, balancing network optimization performance and computational efficiency.

The decoding part is composed of a bilinear interpolation unit for up-sampling to match the feature dimensions and two 1D filters identical to the encoding part. The 1D filter is built in the reverse order of the coding part. This part also consists of four stages, and the codec pattern is reflected in the decoding part, where the 1×3 convolution in each stage is cascaded with the 3×1 convolution in the corresponding stage of the coding part.

The bridge connection part contains a Conv layer, a BN layer, and a ReLU layer. The convolutional layer in the structure has 64 filters and a convolutional size of 3×3 .

2.3. Hybrid Loss Function. The training loss function in this paper is defined as the sum of the outputs of all saliency feature mappings:

$$L = \sum_{k=1}^k \alpha_k I^{(k)}, \quad (2)$$

where $I^{(k)}$ is the loss of the k th lateral output and α_k is the weight of each loss. k is taken as 8, indicating the presence of 8 outputs of the supervised sleeper crack detection network, 7 of which are from CEDNet and the rest from CRRNet. A

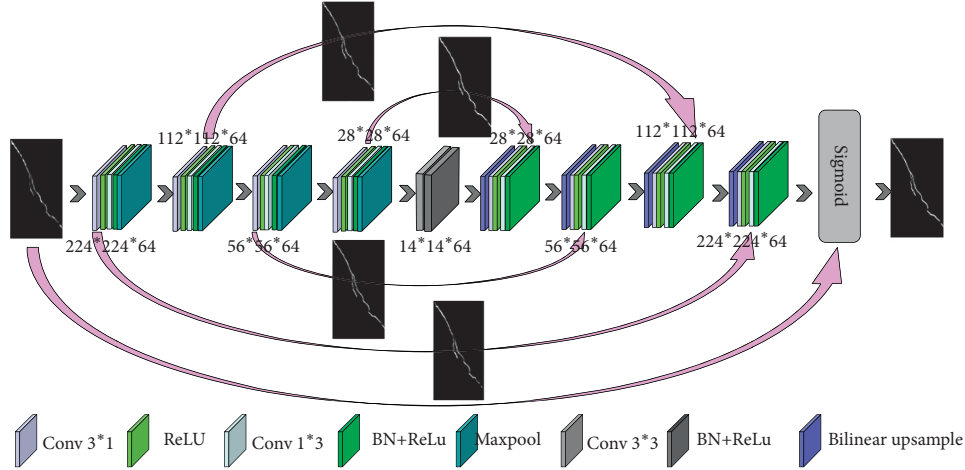


FIGURE 8: CRRNet.

hybrid loss function $l^{(k)}$ that mixes three losses of BCE, SSIM, and IOU is used to obtain a high-quality detection object with complete information:

$$l^{(k)} = l_{\text{bce}}^{(k)} + l_{\text{ssim}}^{(k)} + l_{\text{iou}}^{(k)}, \quad (3)$$

where $l_{\text{bce}}^{(k)}$, $l_{\text{ssim}}^{(k)}$, and $l_{\text{iou}}^{(k)}$ denote the BCE [37], SSIM [38], and IOU losses [39], respectively.

BCE is used as a loss function in this network to supervise the training accuracy of object detection from the pixel level, which can be performed pixel by pixel. The pixel points of foreground and background pixel points are considered equally important and ignore the labeling of the neighboring regions. Accordingly, all pixel points can be converged. BCE is mainly applied to binary classification and segmentation tasks. The definitions are as follows:

$$l_{\text{bce}} = - \sum_{(r,c)} [G(r,c) \log(S(r,c)) + (1 - G(r,c)) \cdot \log(1 - S(r,c))], \quad (4)$$

where $G(r,c) \in \{0, 1\}$ is the ground truth label of the pixel (r,c) and $S(r,c)$ is the predicted probability of the saliency object.

SSIM is used as a loss function for supervised object detection from the local domain level to evaluate the image quality. This loss function assigns a higher weight to the boundary making the loss near the boundary higher, that is, focusing on the attention to the foreground and background boundaries. Progressively more important background losses come into play as the prediction of background pixel points approaches the ground truth, making the boundaries of cracks in the background prediction clearer. SSIM captures structural information in the image; therefore, it is integrated into the blend function to learn the structural information of the saliency object. The definition is as follows:

$$l_{\text{ssim}} = 1 - \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (5)$$

where $x \in \{x_j; j = 1, \dots, N^2\}$ and $y \in \{y_j; j = 1, \dots, N^2\}$ are the pixel values of two corresponding patches cropped

from the predicted probability map S and the binary ground truth mask G , respectively, μ_x, μ_y and σ_x, σ_y are the mean and standard deviations of x and y , respectively, and σ_{xy} is their covariance. $C_1 = 0.01^2$ and $C_2 = 0.03^2$ to avoid dividing by zero.

IOU is originally used to calculate the similarity between two sets and extended to a standard method for evaluating the effectiveness of object detection and segmentation. After the foreground loss is reduced to zero combined with the three loss functions, the BCE can be used to maintain all pixel point gradients and make the IOU focus more on the foreground as the prediction confidence of the foreground network gradually increases. At the feature map level, the following formula is used to oversee the training of object detection and ensure its differentiability in the training loss function.

$$l_{\text{iou}} = 1 - \frac{\sum_{r=1}^H \sum_{c=1}^W S(r,c)G(r,c)}{\sum_{r=1}^H \sum_{c=1}^W [S(r,c) + G(r,c) + S(r,c)G(r,c)]}, \quad (6)$$

where $G(r,c) \in \{0, 1\}$ is the ground truth label of the pixel (r,c) and $S(r,c)$ is the predicted probability of the saliency object.

3. Experiment and Results

3.1. Dataset. The image acquisition device used in the paper is mainly composed of industrial high-speed line matrix camera and camera lens used in accordance with the field design requirements. As shown in Figure 9, the image acquisition system consists of an industrial computer and the LQ-H3X module, where the LQ-H3X module mainly consists of a laser light source and a line array camera. The main parameters of the LQ-H3X module are shown in Table 1.

3.2. Experimental Setup. The model in this paper runs under a Win10 operating system, with dual CPU Intel Xeon Silver 4214 2.2 GHz and NVIDIA RTX 2080Ti 11 GB graphics card. The three networks of object localization, coarse saliency feature extraction, and boundary refinement are built and run under the integrated development environment of PyTorch framework and PyCharm.

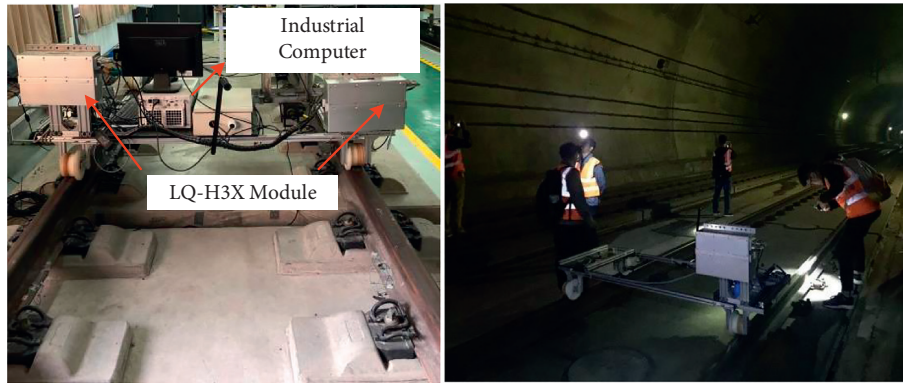


FIGURE 9: Image acquisition. (a) Special rail inspection vehicle. (b) Picture of image acquisition in high-speed railway line.

TABLE 1: LQ-H3X module parameters.

Characteristics	Parameters
Camera resolution	2048/4096 pixels
Scanning frequency	2000 kHz (CL)
Laser power	15 W/25 W
Laser center wavelength	808/915 nm

3.3. *Hyperparameter Configuration.* For the saliency detection part, several parameters with deeper influence, such as initial learning rate, batch size, and epochs, are adjusted during model training. The initial learning rate is closely related to the update of the weight parameters. If it is extremely large, the loss value increases, and the network model is infinitely divergent. If it is extremely small, the loss value decreases extremely slowly, and the parameters are updated extremely slowly. Choosing minibatch stochastic gradient descent and appropriate epochs can improve the running speed of neural network, and let the model converge properly. The actual situation with different combinations of important parameters is compared through several experiments to improve the model training speed, and the results are shown in Table 2.

Initially, with the batch size and epochs unchanged, the loss value decreases faster and faster with the downward adjustment of lr . On the basis of determining the lr of 0.001, the batch size of 4 is selected first in accordance with the performance of the device graphics card and GPU memory size. The epochs are chosen to be adjusted downward from 300 to 100 for the case that the rail crack dataset does not have data diversity. The parameter combination of the lowest loss of 0.046 is established. In consideration of improving the running speed of the neural network, the epochs are increased from 200 to 300 to achieve the same accuracy when the batch size was adjusted to 5. The loss value does not drop as fast as the former in the whole process.

In summary, the optimal combination of parameters selected for the crack recognition module in this paper is as

TABLE 2: Hyperparameter configuration.

Initial learning rate	Batch size	Epochs	Loss
0.001	4	200	0.046
0.001	4	100	0.054
0.001	4	300	0.050
0.001	5	200	0.052
0.001	5	300	0.049
0.002	4	200	0.053
0.002	5	300	0.051
0.005	4	200	0.057
0.005	5	300	0.055

TABLE 3: Combination of parameters.

Parameters	Value
Input size	224×224
Initial learning rate	0.001
Batch size	4
Epochs	200

follows: initial learning rate, batch size, and epochs are set to 0.001, 4, and 200, respectively, and the results are shown in Table 3.

3.4. *Evaluation Metric.* The selected evaluation metrics include F -measure, mAP, F -weighted [40], MAE [41], and AUC [42]. The F -measure is a comprehensive index for the evaluation of the final obtained crack detection results. mAP is used as the average accuracy rate to measure the recognition accuracy, with larger values indicating higher accuracy rates. F -weighted is calculated from the corresponding PR value. The weight of the PR value is the percentage of samples in the total number of samples. The larger the value, the stronger the network performance. MAE is used to measure the error of the test results. The AUC value indicates the high or low performance of the network in classifying the crack and rail background. The closer to 1, the better the network classification.

Its calculation formula is as follows:

$$F_\lambda = \frac{(1 + \lambda^2)P * R}{\lambda^2 * P + R}, \quad (7)$$

$$P = \frac{T_{P_{-1}}}{T_{P_{-1}} + F_{P_{-1}}} * \omega_{-1} + \frac{T_{P_0}}{T_{P_0} + F_{P_0}} * \omega_0 + \frac{T_{P_1}}{T_{P_1} + F_{P_1}} * \omega_1, \quad (8)$$

$$MAE = \frac{1}{W \times H} \sum_{x=1}^W \sum_{y=1}^H |\bar{S}(x, y) - \bar{G}(x, y)|,$$

where P denotes the precision, R denotes the recall, and λ^2 is 0.3, similar to those in reference [40]; ω_{-1} , ω_0 , and ω_1 are the weight ratios of each precision. After the recall is calculated, the F -weighted is obtained from Equation (7). W and H are used to represent the length and width of the input sleeper crack image to be processed.

3.5. Hybrid Loss Function. This work compares and verifies the performance of the proposed hybrid loss function l with single and multiple forms of loss function combined with the network model. As shown in Figure 10, the saliency map predicted by the proposed algorithm is the closest to the ground truth. The integrity of the cracked part of the region with the clarity of the boundary is shown to be the best situation compared with the others.

The quantitative analysis is shown in Table 1. After the comparison experiments for individual loss functions, the more effective l_{bce} and l_{iou} are then selected for the combined analysis. Table 4 shows that the network performance can be optimized only when all three loss functions are simultaneously used. Particularly, our method improves $F_{weighted}$ by 68.4%, 74.8%, 84.1%, and 99.0% on $l_{bce} + l_{iou}$, l_{bce} , l_{iou} , and l_{ssim} , respectively.

3.6. Object Detection. In this experiment, for the comparison of YOLOv3, YOLOv4, and YOLOv5, we conduct the corresponding experiments. The settings of our experimental parameters are shown in Table 5. The initial parameter values for input size, initial learning rate, class, batch size, and epochs for the training of rail crack images are provided.

On the basis of this experimental condition, tests are performed for Tiny YOLOv3, YOLOv3, YOLOv4, and YOLOv5. The model accuracy is verified in terms of the three metrics: precision, recall, and MAP, and the model speed is verified in terms of frames per second (FPS), as shown in Table 6.

YOLOv3 has a higher recognition accuracy than Tiny YOLOv3 and a faster recognition speed than YOLOv4 and YOLOv5. The recognition accuracy can be optimized with the help of SE module and SPP module. In accordance with the experimental results, YOLOv3 can reach the same or even exceed the level of YOLOv4 and YOLOv5.

Therefore, a preliminary conclusion is that YOLOv3 is a more ideal target for optimization. This conclusion can be verified in the final optimized test results.

The prediction frame when the network locates cracks is more accurate compared with the original YOLOv3 by using the improved YOLOv3 network to complete the detection of cracks in the sleeper due to the added attention mechanism to improve the ability to capture the location of cracks. The detection effect is shown in Figure 11.

YOLOv3 and the proposed algorithm are used to detect cracks of the overall roadbed image and the segmented sleeper image by using gray projection method. The comparison of experimental results is shown in Table 7. The comparison of the two inputs of the overall roadbed and sleeper areas shows that the mAP of crack detection is improved by 35.4% and 38.8% on YOLOv3 and improved YOLOv3 after rail sleeper area extraction, respectively, proving the necessity of sleeper area extraction for crack detection. The data entered in the sleeper region column show that the improved YOLOv3 improves the mAP by 6.9% compared with the original network, proving the significant superiority of the present algorithm for sleeper crack detection.

3.7. Feature Extraction. With regard to the sleeper crack dataset constructed in this work, the results of sleeper crack saliency detection obtained using the method of this work are compared with those of several other network models. The models include BAS [43], R2Net [44], SOD100k [45], EDR [46], PFA [47], HED [34], and POOLNet [48]. Figure 12 shows that the proposed algorithm has a good detection of cracks in a variety of situations, including low contrast (1st, 4th, and 6th columns), small target (4th and 6th columns), and complex background (2nd, 3rd, 5th, and 7th columns).

The above evaluation metrics are applied to make a quantitative analysis of all network performance, as shown in Figures 13 and 14. In terms of AUC, the proposed algorithm improves by 6.0%, 0.2%, 1.2%, 2.8%, 3.8%, 10.4%, 15.5%, and 50.9% compared with CEDNet, EDR, BAS, POOLNet, R2Net, PFA, SOD 100 k, and HED, respectively. This result indicates that the proposed algorithm has better classification prediction performance. The MAE value of this work is 0.015, verifying that the algorithm has a small error and high accuracy rate compared with the other networks. The closer the curve composed of precision and recall to the upper-right corner, the better the network classification, and the larger the area enclosed by the F curve and the horizontal axis, the stronger the performance of the network.

The proposed algorithm has better crack integrity and clarity than other algorithms and depends on the form of cascade network used herein. A more complete crack feature can be obtained after cascading the residual networks of codec modes (i.e., CEDNet and CRRNet). In comparison with EDR, the pooling operation after the input convolutional layer is removed in the feature extraction stage to improve the image resolution in this work, and Conv5 and Conv6 are designed to restore the network receptive field. The crack information obtained in this stage is more detailed. By contrast with BAS, a 1D filter is used in

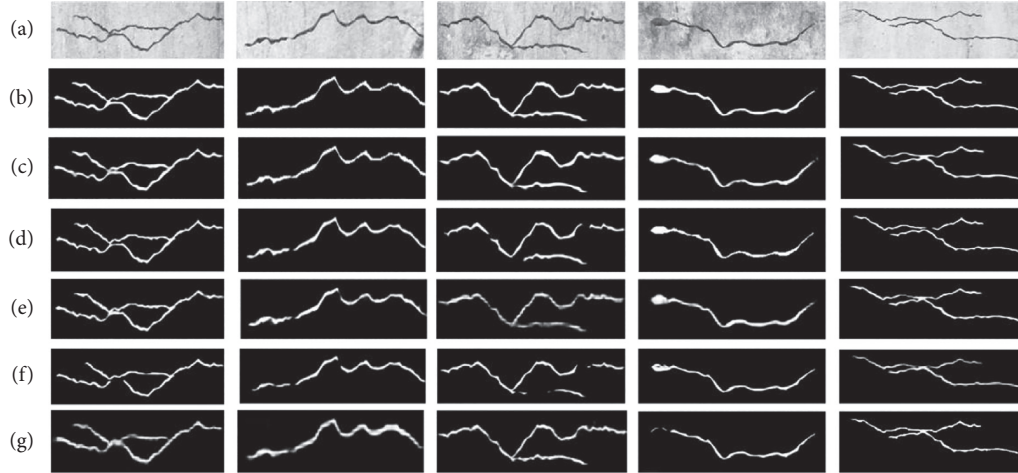


FIGURE 10: Saliency maps under different loss functions. (a) Image. (b) Ground truth. (c) l . (d) $l_{bce} + l_{iou}$. (e) l_{bce} . (f) l_{iou} . (g) l_{ssim} .

TABLE 4: Performance comparison of different loss functions.

Evaluation metrics	F-weighted \uparrow	MAE \downarrow
CEDNet + CRRNet + l	0.805	0.015
CEDNet + CRRNet + $l_{bce} + l_{iou}$	0.254	0.038
CEDNet + CRRNet + l_{bce}	0.203	0.039
CEDNet + CRRNet + l_{iou}	0.128	0.040
CEDNet + CRRNet + l_{ssim}	0.008	0.043

TABLE 5: Setting of initial parameter values.

Parameters	Value
Input size	128×608
Initial learning rate	0.1
Class	1
Batch size	6
Epochs	200

TABLE 6: Comparisons of experimental results.

Models	Precision	Recall	MAP	FPS
Tiny YOLOv3	0.361	0.452	0.392	146.35
YOLOv3	0.794	0.877	0.856	81.7
YOLOv4	0.866	0.924	0.884	26.23
YOLOv5x	0.932	0.905	0.911	32.52
Ours	0.963	0.912	0.915	76.6

the optimization part to balance the refinement performance and computational efficiency. In FPN-based U-Net structures, such as POOLNet and R2Net, the high-level semantic features are continuously diluted because of their

structural limitations when fusing with low-level image features, and the different receptive fields in each layer of the network lead to the loss of local information in the crack saliency map.

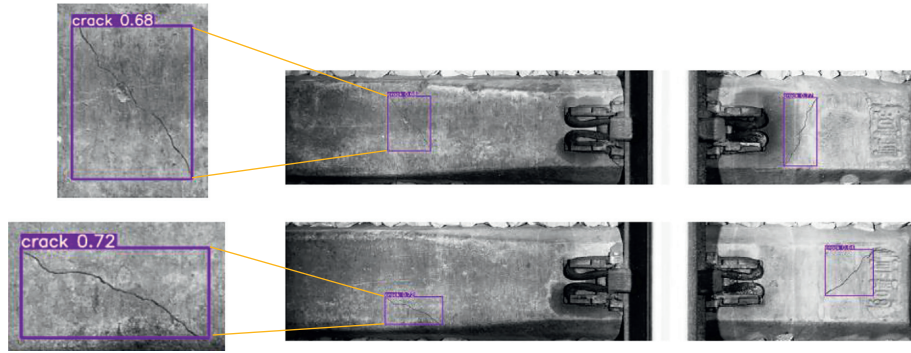


FIGURE 11: Sleeper crack location.

TABLE 7: Comparison of crack detection results.

Algorithm model	Enter the overall roadbed area			Input sleeper area		
	Precision	Recall	mAP	Precision	Recall	mAP
YOLOv3	0.444	0.736	0.632	0.794	0.877	0.856
Ours	0.469	0.792	0.659	0.963	0.912	0.915

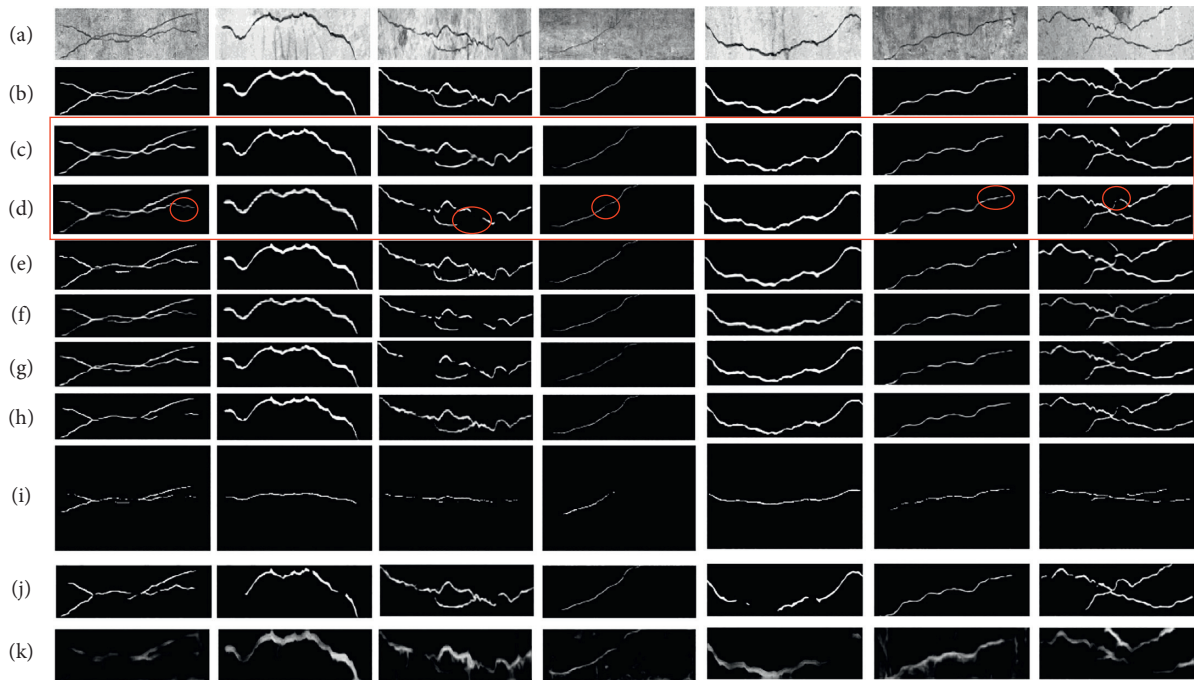


FIGURE 12: Comparison of saliency maps. (a) Image. (b) Ground truth. (c) Ours. (d) CEDNet. (e) EDR. (f) BAS. (g) POOLNet. (h) R2Net. (i) PFA. (j) SOD 100k. (k) HED.

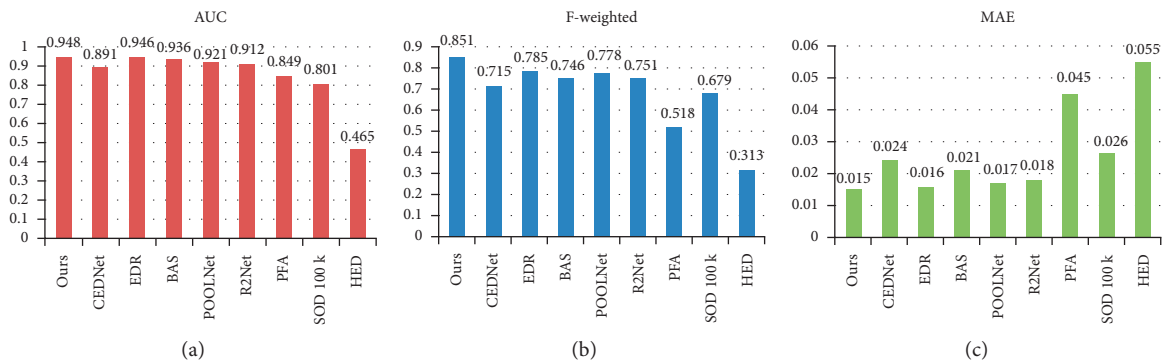


FIGURE 13: Performance comparison of each algorithm. (a) AUC. (b) *F*-weighted. (c) MAE.

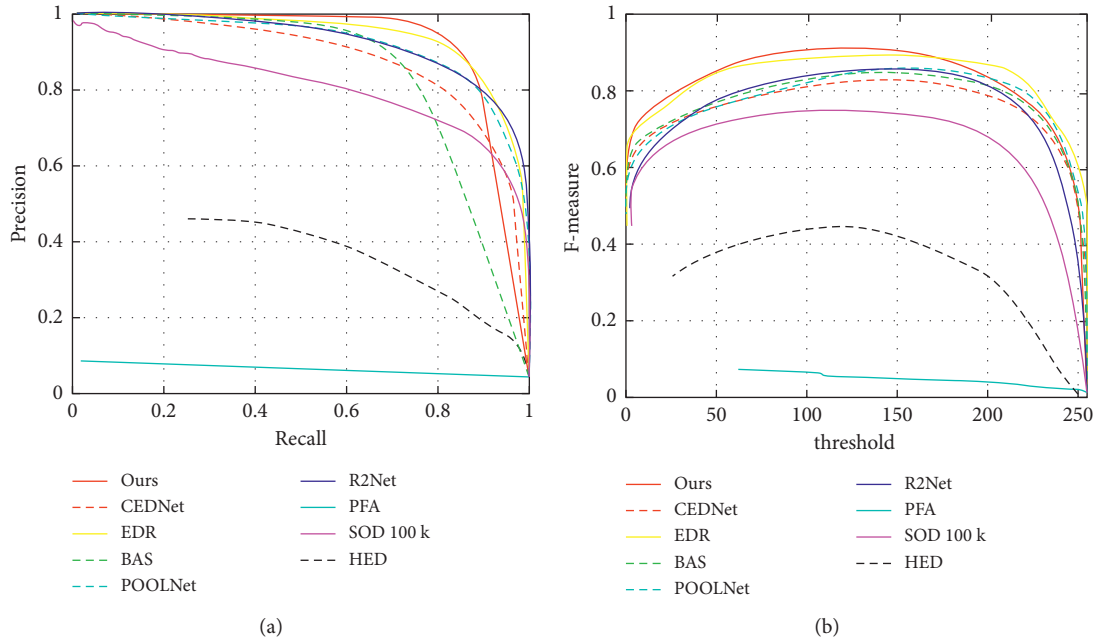


FIGURE 14: Performance comparison diagram of each algorithm. (a) Precision-recall curves. (b) F -measure curves.

4. Conclusion and Expectations

We propose a method for detecting cracks in rail sleepers based on DCNN to address the lack of accuracy in crack detection in crack recognition. The CNN used consists of a modified YOLOv3 network for localization and CEDNet and CRRNet for extracting and optimizing the rail sleeper crack features, respectively. In locating the rail sleeper crack region, the crack on the concrete rail sleeper has some similarity with the ballast edge in the captured images due to the lighting and other causes. However, a grayscale difference can be observed between the rail sleeper and the ballast. Hence, the rail sleeper area is first segmented for the next step. The attention module SE is added at the end of the original YOLOv3 network to extract the cracked areas, thereby improving the accuracy of the rail sleeper crack detection while preserving the network computation speed. CEDNet is constructed to extract more crack information by fusing the high- and low-level features of crack images. The crack boundary refinement network CRRNet is added to optimize the cracks, and the stacked output of the crack coarse saliency feature map and the network can be optimized by learning the residuals from the ground truth. A cascade approach is adopted for the above two networks to obtain a crack saliency map with more complete boundary and region information. The conclusions of this work are as follows:

- (1) A new crack detection method is designed. A cascade network combining CEDNet and CRRNet is used to improve the integrity of crack detection. Its F -weighted is 0.831, MAE is 0.0157, and AUC is 0.9453.
- (2) An improved YOLOv3 network is proposed to localize the cracks, and the attention mechanism SE

module is added at the end of the backbone network. The mAP is improved by 6.9% compared with that of YOLOv3.

- (3) The optimization effects of loss functions BCE, IOU, and SSIM on crack recognition are superimposed to propose a new hybrid loss function for the crack recognition. Particularly, our method improves F_{weighted} by 68.4%, 74.8%, 84.1%, and 99.0% on $l_{\text{bce}} + l_{\text{iou}}$, l_{bce} , l_{iou} , and l_{ssim} , respectively.
- (4) A comprehensive evaluation of the proposed methodology is conducted. Our method has strong robustness and high level of crack detection efficiency compared with the seven state-of-the-art methods.

The proposed crack recognition module consists of two parts. In the optimization stage, we perform the crack boundary refinement process directly on the basis of the first output. Compared with end-to-end learning, this approach requiring secondary adjustment of model parameters increases the time cost and requires more manual processing. Therefore, if the optimization part can be encapsulated into a plug-and-play module, it will greatly improve the efficiency of model operation, which is the next optimization intention of this paper. This paper effectively improves the accuracy of the identification of cracks in the rail sleeper but does not measure the geometric parameters. How to calculate the actual size of the cracks on the basis of existing data is a direction for our future efforts, which is extremely helpful for practical engineering applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 51975347 and 51907117).

References

- [1] W. Zhu, G. Fan, X. Meng et al., "Ultrasound SAFT imaging for HSR ballastless track using the multilayer sound velocity model," *Insight*, vol. 63, no. 4, pp. 199–208, 2021.
- [2] L. Peng, S. Zheng, P. Li, Y. Wang, and Q. Zhong, "A comprehensive detection system for track Geometry using fused vision and inertia," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.
- [3] A. Mohan and S. Poobal, "Crack detection using image processing: a critical review and analysis," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 787–798, 2018.
- [4] P. Kannadaguli and V. Bhat, "Microwave imaging based automatic crack detection system using machine learning for columns," in *Proceedings of the IEEE 9th International Conference on Communication Systems and network technologies (CSNT)*, vol. 5–8, April 2020.
- [5] Y. A. Hsieh and Y. J. Tsai, "Machine learning for crack detection review and model performance comparison," *Journal of Computing in Civil Engineering*, vol. 34, no. 5, 2020.
- [6] M. Nie and C. Wang, "Pavement crack detection based on yolo v3," in *Proceedings of the 2nd International Conference on Safety Produce Informatization (IICSPI)*, pp. 327–330, Chongqing, China, November 2019.
- [7] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "Deepcrack: learning hierarchical convolutional features for crack detection," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498–1512, 2018.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [9] W. Liu, A. Dragomir, and E. Dumitru, "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, vol. v 9905 LNCS, pp. 21–37, Amsterdam, The Netherlands, September 2016.
- [10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only Look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–786, IEEE, Las Vegas, NV, USA, June 2016.
- [11] Y. Cha, W. Choi, and O. Buyukozturk, "Deep learning based crack damage detection using CNNs," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 316–378, 2017.
- [12] V. Mandal, L. Uong, and Y. Adu-Gysmfi, "Automated road crack detection using deep convolutional neural networks," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, pp. 5212–5215, IEEE, Seattle, USA, December 2018.
- [13] W. Li, Z. Shen, and P. Li, "Crack detection of track plate based on YOLO," in *Proceedings of the 12th International Symposium on Computational Intelligence and Design*, pp. 15–18, Hangzhou, China, December 2019.
- [14] Y. Bao, K. Song, J. Liu et al., "Triplet-graph reasoning network for few-shot metal generic surface defect segmentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, Article ID 5011111, 2021.
- [15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.
- [16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Cham, October 2015.
- [17] V. B. Adrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder–decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [18] J. Cheng, W. Xiong, W. Chen, and Y. Gu, "Pixel-level crack detection using," in *Proceedings of the TENCON 2018-2018 IEEE Region10 Conference*, pp. 462–466, Jeju Island, Korea, October 2018.
- [19] M. M. M. Islam and J.-M. Kim, "Vision-based autonomous crack detection of concrete structures using a fully convolutional encoder-decoder network," *Sensors*, vol. 19, no. 19, p. 4251, 2019.
- [20] C. V. Dung, "Autonomous concrete crack detection using deep fully CNN," *Automation in Construction*, vol. 99, p. 5258, 2019.
- [21] U. Escalona, F. Arce, E. Zamora, and H. Sossa, "Fully convolutional networks for automatic pavement crack segmentation," *Computación Y Sistemas*, vol. 23, no. 2, pp. 451–460, 2019.
- [22] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, <https://arxiv.org/abs/1804.02767>.
- [23] S. Tian, "Grayscale projection image stabilization algorithm based on gray bit-plane for moving object," in *Proceedings of the Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pp. 610–613, China, July 2018.
- [24] J. Hu, S. Li, and G. Sun, "Squeeze-and-Excitation networks," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Jeju Island, South Korea, June 2018.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [26] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation' in fully convolutional networks," *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018*, vol. 11070, pp. 421–429, 2018.
- [27] M. Xu, Z. Zhang, H. Hu et al., "End-to-End semi-supervised object detection with soft teacher," 2021, <https://arxiv.org/pdf/2106.09018.pdf>.
- [28] X. Dai, Y. Chen, B. Xiao, and D. Chen, "Dynamic head: unifying object detection heads with attentions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, June 2021.
- [29] T. Liang, X. Chu, Y. Wang, Z. Tang, and W. Chu, "CBNetV2: a composite backbone network architecture for object detection," 2021, <https://arxiv.org/abs/2107.00420>.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference*

- on *Computer Vision and Pattern Recognition*, pp. 770–778, Honolulu, HI, USA, June 2016.
- [31] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Lille, France, July 2015.
- [32] R. H. R. Hahnloser, H. S. Seung, and J. J. Slotine, “Permitted and forbidden sets in symmetric threshold-linear networks,” *Advances in Neural Information Processing Systems*, vol. 15, pp. 217–223, 2001.
- [33] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” 2015, <https://arxiv.org/abs/1511.07122>.
- [34] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1395–1403, Santiago, Chile, December 2015.
- [35] C. Peng, X. Zhang, and G. Yu, *Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network*, pp. 4353–4361, CVPR, Honolulu, HI, USA, 2017.
- [36] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” *In AISTATS*, pp. 315–323, 2011.
- [37] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A tutorial on the cross-entropy method,” *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.
- [38] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *Proceedings of the IEEE Asilomar Conference on Signals*, vol. 2, pp. 1398–1402, Pacific Grove, CA, USA, December 2003.
- [39] G. Mattyus, W. Luo, and R. Urtasun, “DeepRoadMapper: extracting road topology from aerial images,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3458–3466, Venice, Italy, October 2017.
- [40] P. Tao, H. Yi, C. Wei, L. Ge, and L. Xu, “A method based on weighted F-score and SVM for feature selection,” in *Proceedings of the 25th Chinese Control and Decision Conference (CCDC)*, May 2013.
- [41] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, “Salient object detection: a benchmark,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.
- [42] F. Liu, T. Shen, and S. Lou, “Deep network saliency detection based on global model and local optimization,” *Acta Optica Sinica*, vol. 37, no. 12, Article ID 1215005, 2017.
- [43] X. Qin, Z. Zhang, C. Huang, and C. Gao, “BASNet: boundary-aware salient object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7471–7481, Long Beach, CA, USA, June 2019.
- [44] M. Z. Alom, M. Hasan, C. Yakopcic, and T. M. Taha, “Nuclei Segmentation with Recurrent Residual CNNs Based U-Net (R2U-Net),” in *Proceedings of the NAECON 2018 - IEEE National Aerospace and Electronics Conference*, pp. 228–233, Dayton, OH, USA, 2018.
- [45] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, “Salient object detection in the deep learning era: an in-depth survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, <https://arxiv.org/pdf/1904.09146>, 2021.
- [46] G. Song, K. Song, and Y. Yan, “EDRNet: encoder-decoder residual network for salient object detection of strip steel surface defects,” *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9709–9719, 2020.
- [47] T. Zhao and X. Wu, “Pyramid feature attention network for saliency detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3080–3089, Long Beach, CA, USA, June 2019.
- [48] J. Liu, Q. Hou, M. Cheng, and J. Feng, “A simple pooling-based design for real-time salient object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3912–3921, Long Beach, CA, USA, April 2019.