DOI: 10.1111/mice.13109

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Intelligent recognition of defects in high-speed railway slab track with limited dataset

Xiaopei Cai[1] | **Xueyang Tang**[1] | **Shuo Pan**[2] | **Yi Wang**[1] | **Hai Yan**[2] | **Yuheng Ren**[3] | **Ning Chen**[2] | **Yue Hou**[4]

[1]School of Civil Engineering, Beijing Jiaotong University, Beijing, China

[2]Beijing Key laboratory of Traffic Engineering, Beijing University of Technology, Beijing, China

[3]ARUP Group, Cardiff, UK

[4]Department of Civil Engineering, Faculty of Science and Engineering, Swansea University, Swansea, UK

Correspondence
Hai Yan, Beijing Key Laboratory of Traffic Engineering, Beijing University of Technology, Beijing, China.
Email: yhai@bjut.edu.cn

Yue Hou, Department of Civil Engineering, Faculty of Science and Engineering, Swansea University, Swansea, UK.
Email: yue.hou@swansea.ac.uk

Funding information
National Natural Science Foundation of China, Grant/Award Number: 52178405; Project of Science and Technology Research and Development Program of China State Railway Group Co., Ltd., Grant/Award Number: SY2022T002; Open Fund of National Key Laboratory of High-speed Railway Track Technology, Grant/Award Number: 2021YJ053

Abstract

During the regular service life of high-speed railway (HSR), there might be serious defects in the concrete slabs of the infrastructure systems, which may further significantly affect public transportation safety. To address these serious issues and fulfill the regular functions of HSR, the traditional methods for railway engineers involve carrying out regular on-site inspections manually or by semi-automatic inspection vehicles, and conducting timely corresponding repairing approaches and maintenance, where these methods are time-consuming and dangerous. In recent years, machine learning methods have been widely applied to the intelligent and automatic detection of severe defects in HSR. Currently, one of the most serious problems is the lack of sufficient high-quality data for model training, resulting in low recognition accuracy in HSR defects. To solve this problem, this paper proposed an intelligent recognition of defects in concrete slabs of HSR based on a few-shot learning model, that is, an artificial intelligence model based on limited data size, which recognizes three service conditions of concrete slabs in HSR: cracks, track board gaps, and unbroken state. Lightweight few-shot learning models specifically designed for HSR detection were proposed. Experiments were conducted to compare the performances of different lightweight-designed models, including accuracy, parameter quantity, and testing time. Results showed that the optimum model can fast and satisfactorily recognize the defects in HSR with a very limited data size of 10 samples for each training category, with a satisfactory accuracy of 73.9% in the test dataset with 20 samples for each category, parameter amounts of 2.8 million, and a testing time of 2.2 s per image. This study provides a reference for the automatic recognition of defects in HSR by railway engineers with insufficient samples.

1 | INTRODUCTION

Compared with the traditional railway, high-speed railway (HSR) has the advantages of high speed, and high comfort/convenience, which is one of the key components of the modern transport system (L. Gao, 2015; Zhai, 2020). However, in the current practice, HSR is facing huge challenges in operational safety and long-term

service performance because of the short construction period, large construction scale, and lack of operation experience (Zhai & Zhao, 2016). For example, there exist significant defects in the civil infrastructures, including track slab cracks, track board gaps, falling blocks, and some other defects (Ai et al., 2018) that may greatly affect the HSR service life and service safety. Life-cycle engineering is an important element in the railway (X. Li et al., 2023; Val et al., 2000; Wilson et al., 1997). Therefore, timely and accurate identification of HSR defects is of great significance for railway engineers.

The traditional methods of identifying HSR defect include manual inspection, using semi-automatic track inspection vehicles, and so forth. In the current engineering practices, Germany, France, Japan, China, and some other countries have established a series of operation safety systems based on regular inspection and maintenance operations on HSR infrastructure. In recent years, advanced characterization methods, including ultrasonic (K. Wang et al., 2020), ground-penetrating radar (Clark et al., 2003; Huang et al., 2023; Nurmikolu, 2012), acoustic emission (J. Wang et al., 2018), unmanned aerial vehicles (UAVs) (Bang et al., 2022; Kang & Cha, 2018), infrared imaging, and so forth, have been widely used in the field of HSR defect identification (Park et al., 2016). However, there are two major shortcomings of these approaches: (1) The detection speeds of these advanced characterization methods are slow. (2) The analysis of inspection data is mainly reliant on manual recognition by railway engineers, which is time-consuming and labor-intensive.

With the rapid development of machine learning-based methods (Hinton & Salakhutdinov, 2006; Rafiei & Adeli, 2017a) and computing power (Adeli & Kamal, 1989), artificial intelligence (AI) has been widely used in the civil engineering area (Rafiei & Adeli, 2016, 2018; Rafiei et al., 2017), especially in the automatic recognition of defects in civil infrastructures (Tsiflakos & Owen, 1993; Yao et al., 2023; Yoon et al., 2016). Compared to traditional recognition methods, AI models have better performance and faster computation speed (Adeli, 2001; Adeli & Park, 1995; Shajihan et al., 2022). Therefore, it is possible to employ AI models for the automatic recognition of defects and service conditions in railways. Trosino et al. (2002) used machine learning-based vision technology to detect the track state. Marino et al. (2007) proposed a visual inspection system for railway, which used a wavelet transform to extract features and applied a neural network to detect if hexagonal bolt fasteners were missing. Gibert et al. (2015, 2017) extracted a histogram of oriented gradient features from track images and then fed them into multiple support vector machine classifiers. Chun et al. (2017) proposed a method for identifying cracks and mold marks using convolutional neural networks (CNNs). F. C. Chen et al. (2018) proposed a deep learning network based on CNN and a naive Bayesian (NB) data fusion scheme, namely, NB-CNN, which effectively discarded false alarm information and improved crack detection effects in images through NB decision. Z. Liu et al. (2018) used an improved faster region-CNN to detect locator support. J. Chen et al. (2018) proposed a cascade network to identify catenary accessories. T. Ye et al. (2018) and J. Li et al. (2018) modified the convolution type based on a single-shot multi-box detector (SSD) and replaced the ordinary convolution with a depthwise separable convolution to save computational cost and realize the target detection. Zhong et al. (2019) improved the performance versus accuracy network (PVANET) through a feature fusion mechanism and proposed the PVANET++ algorithm to detect cotter pins. B. Guo et al. (2019) proposed a method for generating images of railway intrusion targets based on improved conditional deep convolutional generative adversarial networks. Chun et al. have made remarkable contributions to the field of damage detection (Chun & Hayashi, 2021; Chun et al., 2023). They completed multiple non-destructive testing and evaluation of internal damage in reinforced concrete based on the random forest algorithm (Chun et al., 2020) and completed the recording of bridge damage areas and automatic generation of image captions based on deep learning technology (Chun et al., 2021; Yamane et al., 2023). T. Ye et al. (2020) designed an automatic object detection system capable of detecting track obstacles and curves. Y. Li et al. (2020) used the SSD to implement railway detection. F. Guo et al. (2021) proposed a real-time pixel-level rail components detection framework that realized fast object detection and highly accurate instance segmentation. Tong et al. (2023) proposed a novel anchor-adaptive railway track detection network realizing full-angle railway track detection for the UAV aerial images taken from arbitrary viewing angles. W. Ye et al. (2022, 2023) realized rapid detection of apparent cracks in rail concrete based on deep learning and developed a systematic pixel-level crack segmentation–quantification method suited for nighttime detection of slab tracks. Q. Wang et al. (2023) enabled rapid, low-cost, and continuous monitoring for rail wear on a running band by a depth-plus-region fusion network. Rosso, Aloisio, et al. (2023) analyzed the exertion of various deep learning models in order to increase the productivity of classifying ground penetrating radar images for monitoring structural health purposes, especially focusing on road tunnel linings evaluations. In addition, they implemented intelligent detection of road tunnel defects based on ground penetrating radar and employed state-of-the-art neural visual transformer architecture and migration learning methods in the existing defect classification framework (Rosso,

Marasco, et al., 2023). Their research impressively increases the capabilities of the AI-based road tunnels automatic defects classification paradigm for road tunnel structural health monitoring. These deep learning algorithms have made remarkable progress in terms of model accuracy, scalability, and robustness, resulting in great success in defect identification areas (He et al., 2016; Krizhevsky et al., 2017).

Previously, most of these AI models have relied heavily on large amounts of training data. In real civil engineering practices, obtaining very large-scale datasets is sometimes unachievable. For example, transportation engineers often face problems such as small and unbalanced datasets (N. Chen et al., 2022). In this case, when the data sizes of specific categories are insufficient and/or unbalanced, the deep neural network model tends to overfit (Zhou et al., 2017) or underfit.

At present, there has been significant progress in the AI models with small sample sizes (Gao & Mosalam, 2018; Grabczewski & Jankowski, 2007; Lake et al., 2013), such as few-shot learning models, which can improve the anti-inference ability and generalization performance. As one of the most widely used few-shot learning models, the Siamese network (SN) (Bromley et al., 1993) is robust to category imbalance problems and easy to do integration learning. Chopra et al. (2005) used SN for face recognition to increase the number of sample cycles for training without changing the original number of training samples, preventing the overfitting problem caused by too few samples. Yuan et al. (2017) creatively applied SN to solve the problem of sparsely labeled samples and unbalanced category distribution, enhancing the application value of SN in solving small-sample data learning. Zhang et al. (2019) proposed an SN model for rolling bearing fault diagnosis with limited data. In addition, it can also be enhanced by Ganerative adversarial network (GAN) and other methods to overcome the problems caused by the limited amount of data (Marano et al., 2023).

Meanwhile, in computer vision, the research on image retrieval (Triantafillou et al., 2017), object tracking (Bertinetto et al., 2016), radar signal recognition (Luo et al., 2022), and other image recognition (W. Wang et al., 2022; Zheng et al., 2023) based on small sample learning is developing rapidly. Xue et al. (2023) proposed an adaptive cross-scenario few-shot learning framework for structural damage detection. Pan et al. (2023) applied few-shot learning to the identification of pavement textures in extremely limited sample scenarios. These studies make it applicable to realize automatic HSR defect recognition using deep learning techniques designed for practices with limited data size.

In general, deep learning models contain too many parameters and require complex computational processes, consuming significant resources in terms of both storage and computation. In railway engineering, especially for HSR infrastructure, it is more important to quickly recognize the defects with reasonable accuracy. Therefore, it is necessary to conduct lightweight design of AI models. Deep learning model compression algorithms usually involve tensor low-rank decomposition, parameter clipping, training adjustment, network structure optimization, and so forth (Choudhary et al., 2020; Deng et al., 2020). Compared with the original networks, lightweight models have the advantages of a light structure, simple calculations, and strong portability (Xu et al., 2018; Yang et al., 2019). At present, lightweight models have been widely used in natural language processing (Jelodar et al., 2020), data mining (Feng et al., 2019), medical image processing (Fan et al., 2020), and semantic segmentation (Chaurasia & Culurciello et al., 2017; Paszke et al., 2016; Sun et al., 2020). At the same time, the lightweight network is also widely used in the identification and detection of infrastructure injuries and damages (Hu et al., 2021). Z. Liu et al. (2023) evaluated different types of pavement damage recognition based on the lightweight design of GANs. Cao et al. (2022) presented a more effective detection method based on a lightweight neural network for the problem of foreign object intrusion in railways. Y. Li et al. (2022) and Ouyang et al. (2023) have offered more efficient and accurate approaches for detecting crack diseases in various structures, such as dams and tunnels. Based on the abovementioned techniques, this paper proposed a method for intelligent recognition of defects in slabs of HSR based on lightweight-designed few-shot learning models. SN designed specifically for automatic detection of HSR defect is created. The computational steps are as follows: First, various lightweight-designed SNs were utilized for HSR slab defect classification. Second, the results on classification accuracy, parameter quantity, and testing time between the original SN and the improved SN were compared under the same conditions. Finally, the optimum lightweight was determined. The model's performance metrics encompass three aspects: accuracy, complexity, and computational cost. Accuracy refers to the proportion of correctly classified samples to the total number of samples. Complexity is represented by the number of parameters, while computational cost is measured by testing time. The flowchart of this method is shown in Figure 1.

The advantages of this method include: (1) A lightweight few-shot learning model for HSR slab defect classification based on SN was established, which significantly reduces the parameter quantity and testing time. (2) The classification accuracies of the lightweight SN were further improved by designing deeper and wider networks, making it more suitable for practical railway engineering applications.

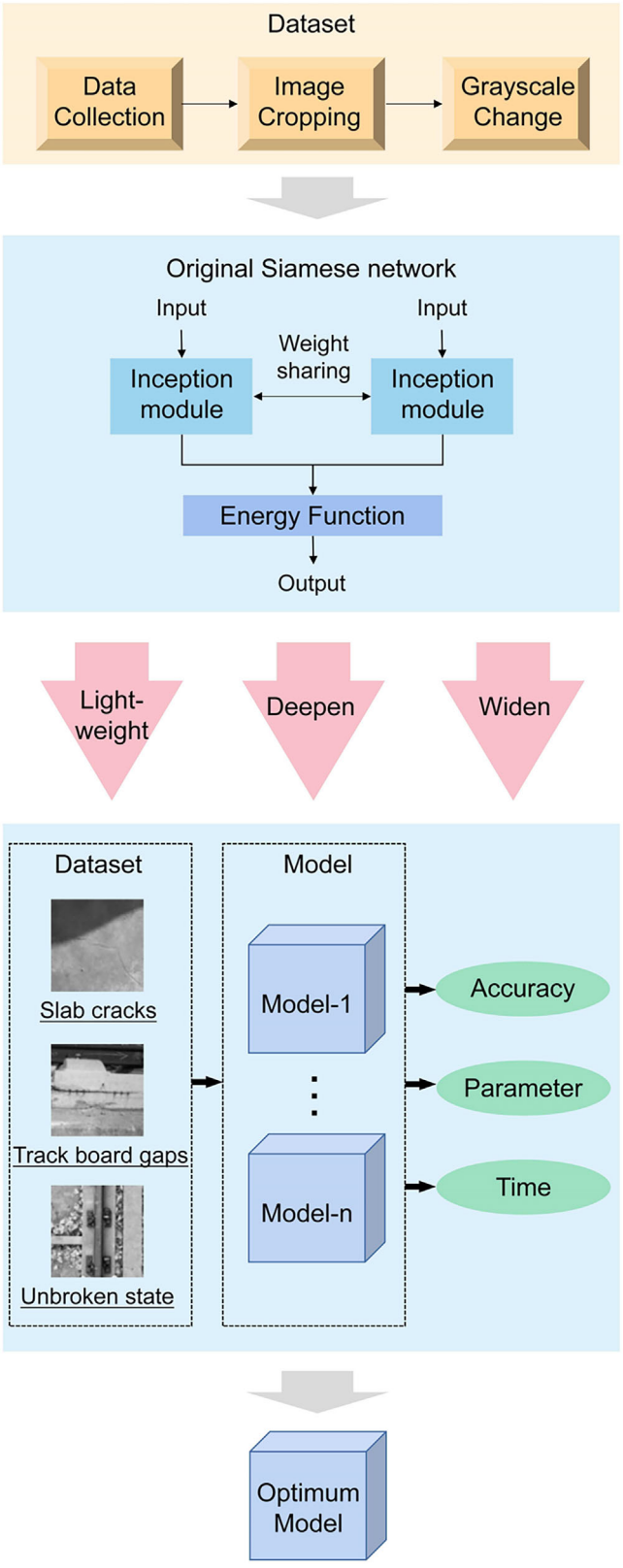

FIGURE 1 Flowchart of high-speed railway (HSR) defects recognition method based on lightweight few shots learning (Siamese network).

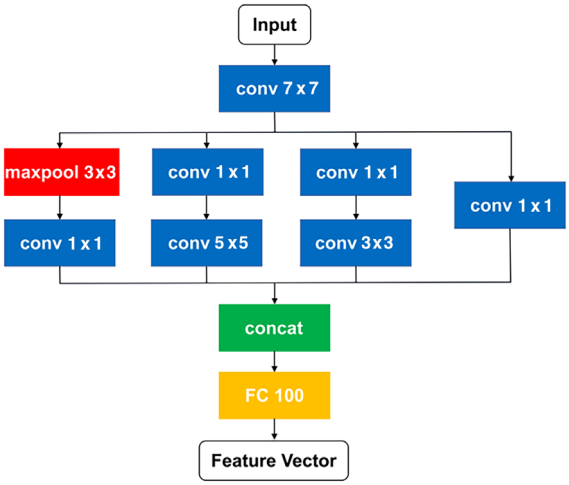

FIGURE 2 The original structure of the SN sub-network.

2 | METHODOLOGY

2.1 | SN

SN is a type of neural network consisting of two identical sub-networks, which share the same architecture and weights (Bromley et al., 1993). When the dataset has very limited samples, SN outperforms conventional deep learning methods for classification tasks, as the SN minimizes the distance between samples from the same category and maximizes the distance between samples from different categories during training, thus making full use of the samples and introducing more computation.

The key to achieving a satisfactory classification result for SN is that its sub-network can effectively extract the characteristics of samples, from which the similarities and differences between different HSR defects can be learned. In this study, the inception module was adopted as the backbone of the sub-networks in the SN, which has been demonstrated to achieve excellent performance in classification tasks and has the advantage of being lightweight (Szegedy et al., 2015). The inception module uses multi-scale filters to perform parallel convolutions on the input data and concatenate the outputs. In this way, the inception module can capture features of different scales and complexities in the input data and improve computational efficiency.

The original structure of the SN sub-network proposed in this paper using the inception module is shown in Figure 2. The blue blocks represent convolutional layers, the red blocks represent pooling layers, the yellow blocks represent fully connected (FC) layers, and the green blocks represent dimension concatenation. The

same pattern applies to Figures 2 through 12. The input image first goes through a 7 × 7 convolution layer with 64 channels, followed by parallel operations including: (1) passing through a max pooling layer with a pooling kernel size of 3 × 3, then through a 1 × 1 convolution layer; (2) passing through a 1 × 1 convolution layer with 32 channels, then through a 5 × 5 convolution layer; (3) passing through a 1 × 1 convolution layer with 32 channels, then through a 3 × 3 convolution layer; and (4) passing only through a 1 × 1 convolution layer. Each feature map of the parallel operations has 128 channels. Then, the feature maps are concatenated and converted into a one-dimensional vector, which is then passed through an FC layer with 100 output nodes. The output vector of the FC layer is the feature vector extracted by the SN sub-network.

In this study, to realize the target of both lightweight computation and satisfactory recognition accuracy, a total of 10 improved models were designed and tested: initially two lightweight-designed models, six deepened-designed models, and two widened-designed models.

2.2 | Lightweight network design

To realize a rapid detection of HSR defects at a reasonable computation cost, this study proposed two methods to lightweight the original network structures: (1) narrowing and deepening the FC layer and (2) adding a 1 × 1 convolution layer.

The reason why the original sub-network structure of the SN is not lightweight enough is that there are too many input and output nodes in the FC layer, resulting in too many parameters. Therefore, the two abovementioned approaches were aimed at reducing the output nodes and the input nodes. However, as fewer output nodes may lead to a decrease in the feature extraction effect, this study further used the method of narrowing and deepening the FC layer. The output nodes of the FC layer were reduced to 10, and two FC layers with 10 output nodes were added after this layer, as shown in Figure 3, thus increasing the feature extraction capability while lightweighting the network. The input nodes were derived from the one-dimensional feature vector formed by transforming the feature map extracted by the inception module. Simply, reducing the input nodes means compressing the feature map output by the inception module. Accordingly, this study proposed adding a 1 × 1 convolution layer with 128 channels after the feature map output by the inception module to achieve lightweighting as shown in Figure 4.

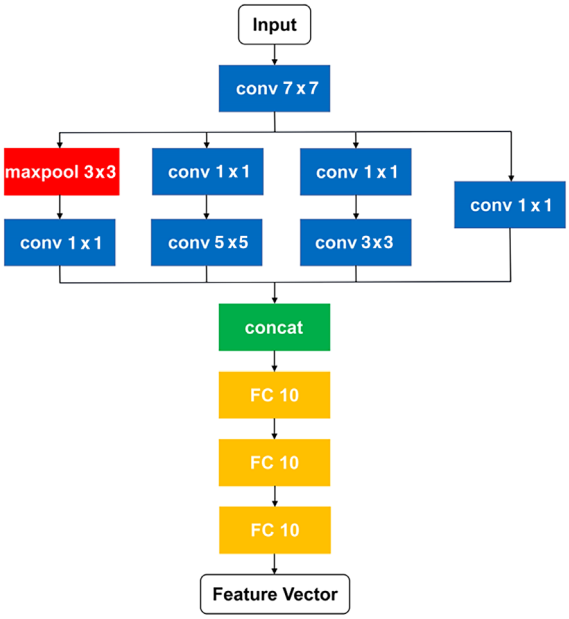

FIGURE 3 Narrowing and deepening the fully connected (FC) layer for the SN sub-network.

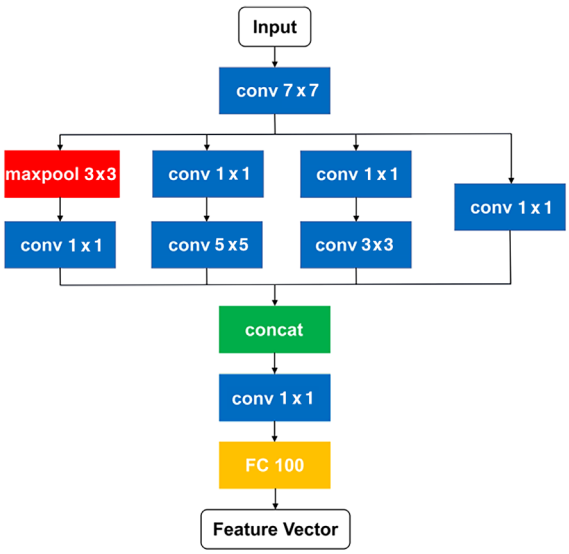

FIGURE 4 Adding a 1 × 1 convolution layer for the SN sub-network.

2.3 | Deeper network design

To achieve higher recognition accuracy in HSR defects with a very limited dataset, designing deeper networks to enhance feature extraction capability is crucial. This study used six methods to increase the depth of the sub-network of the SN. First, four methods were employed based on the lightweight design method of adding a 1 × 1 convolution layer: (1) adding 1 × 1 convolution layers; (2) adding 3 × 3 convolution layers; (3) adding 3 × 3 convolution layers and

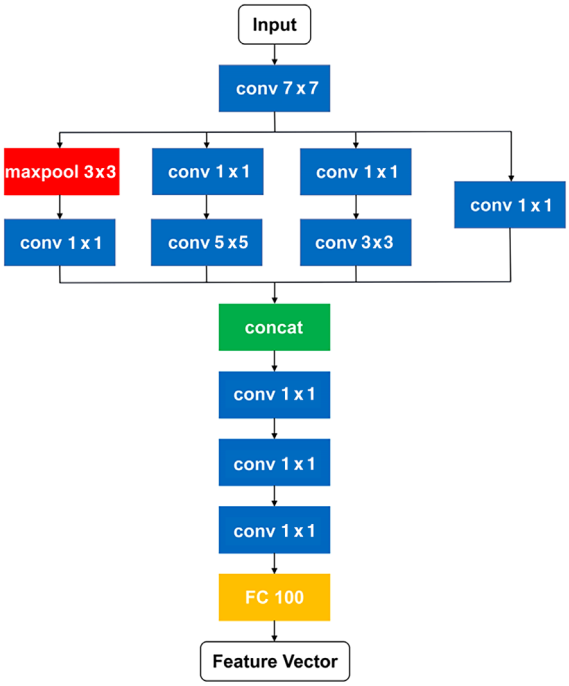

FIGURE 5 Adding 1 × 1 convolution layers for the SN sub-network.

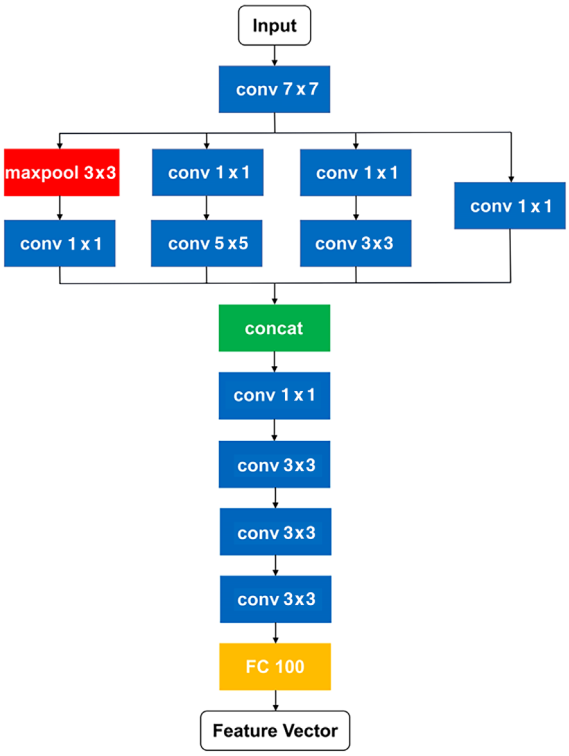

FIGURE 6 Adding three 3 × 3 convolution layers for the SN sub-network.

pooling layers; (4) adding more 3 × 3 convolution layers and pooling layers. Second, an FC layer with 100 nodes was added to the original structure of the sub-network of the SN. Third, double inception modules were used to deepen the network.

Normally, to increase the depth of the network, the simplest method is to add 1 × 1 convolution layers, as shown in Figure 5, which only have a few parameters and computations. On the contrary, adding 3 × 3 convolution layers introduces more parameters and computations, as shown in Figure 6, while having a higher feature extraction capability than 1 × 1 convolution layers. To avoid excessive complexity when adding 3 × 3 convolution layers, it is reasonable to set a 3 × 3 pooling layer with a stride of 2 × 2 after the 3 × 3 convolution layer, which can reduce the dimensions of the feature map and avoid excessive computations. To investigate whether the deeper network has better performance, two structures of adding three and six 3 × 3 convolution layers were designed. For the structure of adding three convolution layers, a maximum pooling layer was set after each convolution layer, as shown in Figure 7, while for the structure of adding six convolution layers, a maximum pooling layer was set after the first, second, and sixth 3 × 3 convolution layers as shown in Figure 8. In addition, adding an FC layer with 100 nodes directly is also feasible as shown in Figure 9.

At the same time, to compare the effectiveness of these methods with directly stacking the inception modules, a structure with two inception modules was also included in

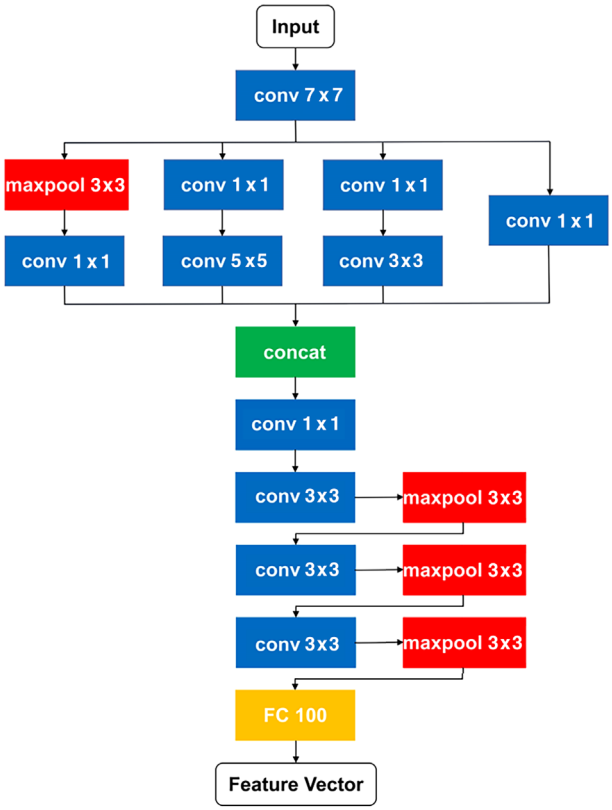

FIGURE 7 Adding three 3 × 3 convolution layers while adding three pooling layers for the SN sub-network.

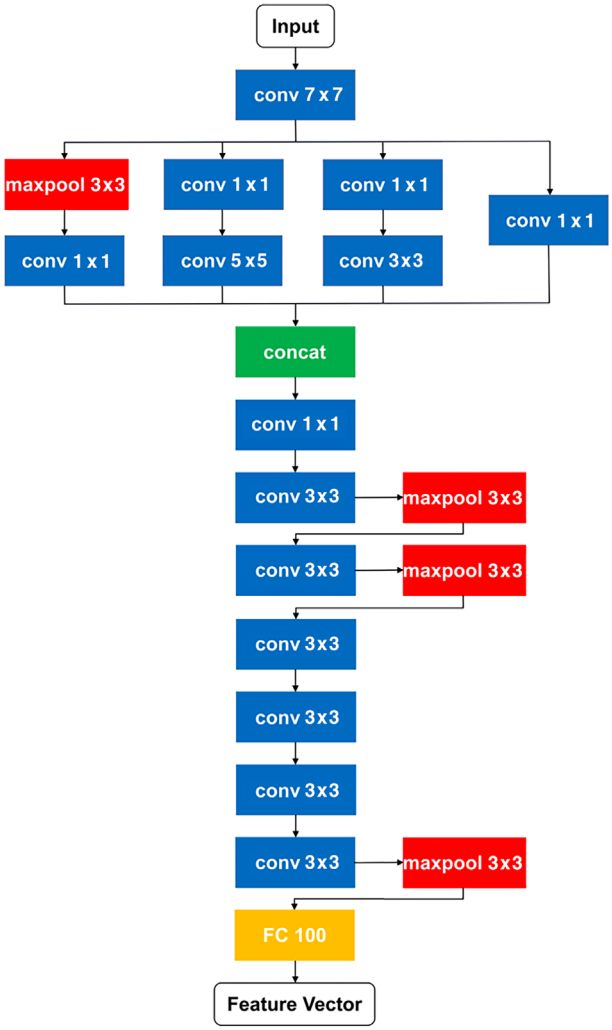

FIGURE 8 Adding six 3 × 3 convolution layers and three pooling layers for the SN sub-network.

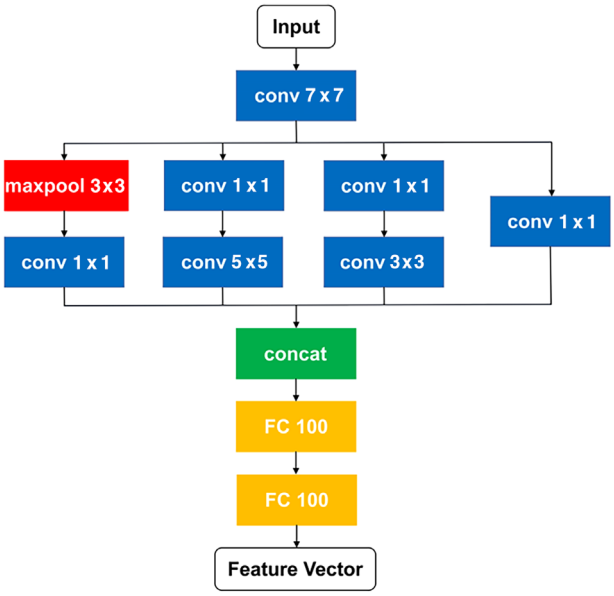

FIGURE 9 Adding one FC layer for the SN sub-network.

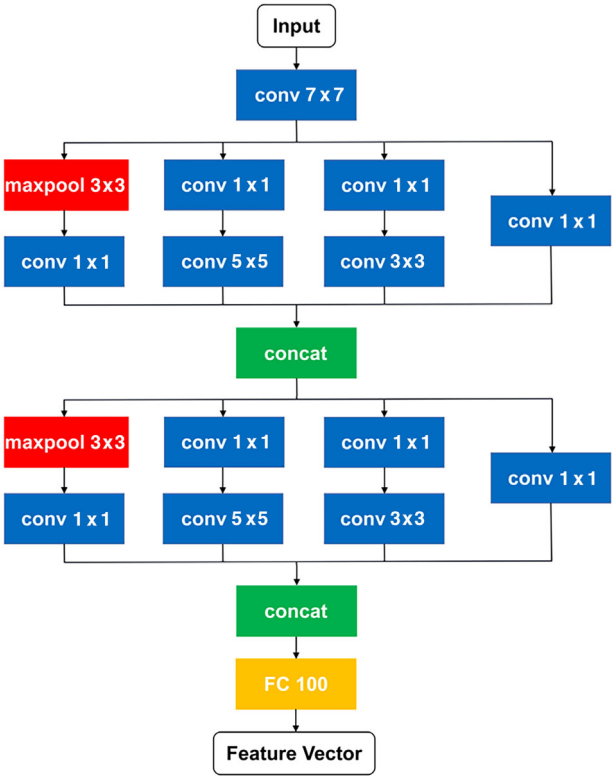

FIGURE 10 Double inception modules for the SN sub-network.

the experiment, where the concatenated feature map of the first inception module was used as the input of the second inception module as shown in Figure 10. The structure of the second inception module was the same as the first but with different weights, and the output of the FC layer of the second inception module was the feature vector of the SN sub-network.

2.4 | Wider network design

Another method to improve the accuracy of the few-shot learning model for HSR defect detection was to increase the width of the inception module in the sub-network of SN. At the same time, to reduce the computational cost, 1 × 1 convolution layers were used while increasing the width of the inception module.

The original inception module has three scales of convolution kernels: 1 × 1, 3 × 3, and 5 × 5. To enhance its capability of extracting larger-scale features, a 7 × 7 convolution kernel was added. Correspondingly, a branch was added to the parallel computing of the inception module, where the feature map first passed through a 1 × 1 convolution layer with 32 channels and then through a 7 × 7 convolution layer with 128 channels. After that, the five sets of 128-channel feature maps were concatenated,

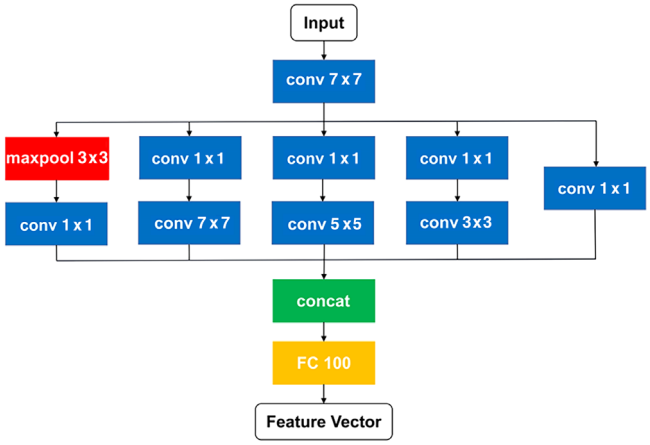

FIGURE 11 Adding a branch for the SN sub-network.

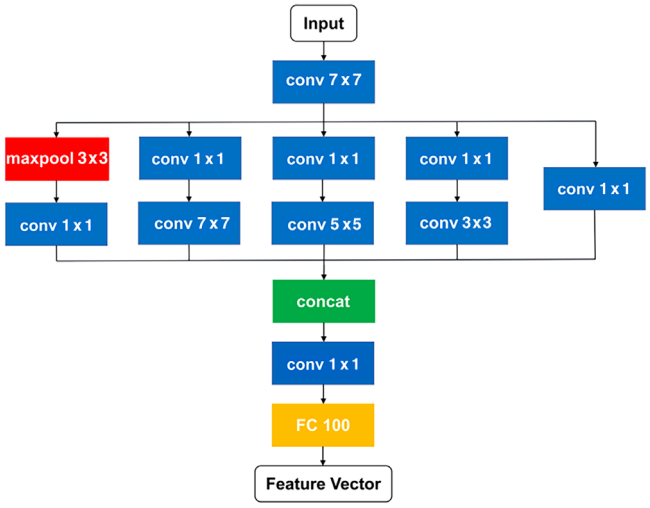

FIGURE 12 Adding a branch and a 1×1 convolution layer for the SN sub-network.

converted to one dimension, and passed through an FC layer to obtain a feature vector as shown in Figure 11. Due to the excessive number of channels after concatenating the five sets of 128-channel feature maps, which led to a high amount of parameters and computations, a 1×1 convolution layer with 128 channels was applied to reduce the number of channels. Then, it was converted to one dimension and passed through an FC layer to obtain the feature vector as shown in Figure 12.

3 | EXPERIMENT DESIGN

3.1 | Dataset

The image data were sourced from on-site photographs of the Beijing–Shanghai HSR lines taken by railway engineers. Due to limitations of field safety, Figure 13 is used as a schematic view to show the detection of HSR defects for the collection of image dataset on the test loop of the China Academy of Railway Sciences. To facilitate image processing and minimize computational cost, all images were cropped to a size of 112×112 pixels and converted to single-channel grayscale images. As a preliminary work on HSR defect recognition based on a very limited dataset, this paper focuses on the classification of three conditions of HSR infrastructures: slab cracks, track board gaps, and unbroken states, corresponding to the images shown in Figure 14.

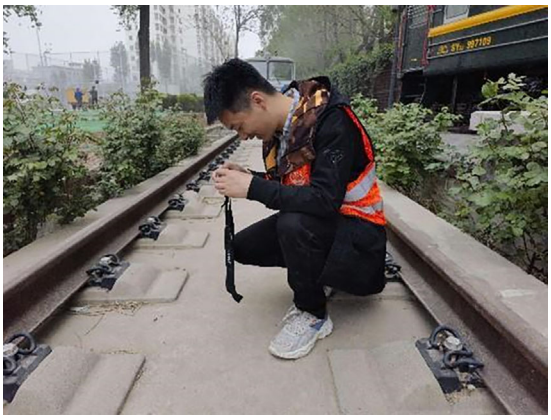

FIGURE 13 Schematic view of HSR defect detection (at China Academy of Railway Sciences test loop).

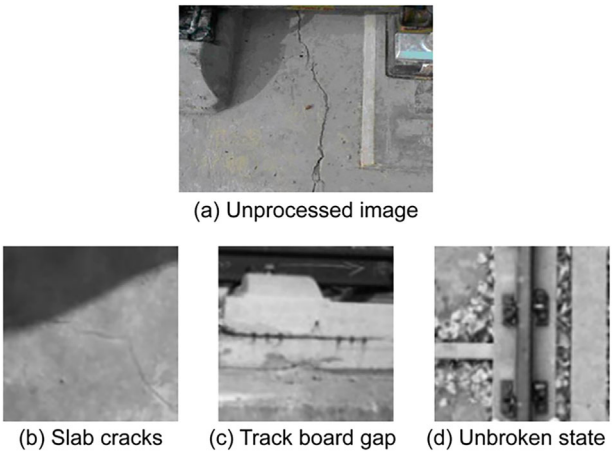

FIGURE 14 Examples of HSR typical defects.

The ballastless track of HSR is a reinforced concrete structure (Ahmadkhanlou & Adeli, 2005; Aldwaik & Adeli, 2016). The slab cracks in Figure 14b refer to cracks or fissures on the surface of track boards. As these cracks gradually expand, the internal steel reinforcements may corrode, resulting in a reduction in the load-bearing capacity and durability of the track. If the cracks continue to develop, the track boards may experience through-cracks, seriously affecting the stability and safety of the

TABLE 1 Sample composition of the dataset.

	Slab cracks	Track board gaps	Unbroken state
Sample size of training set	10	10	10
Sample size of test set	20	20	20

track. The track board gaps in Figure 14c are gaps that occur between the structural layers of the track when subjected to complex operating conditions. These gaps can lead to severe track irregularities, thereby affecting train safety. Both of these typical defects frequently occur during the operation of high-speed rail and can be visually observed in the images (Zhai et al., 2014; Zhu & Cai, 2012).

Considering that it might be very difficult to collect the on-site HSR defect data and to provide rail engineers with a method to intelligently recognize the HSR defects even with very few samples, this study conducted a classification task based on a very small dataset. Only 10 and 20 samples were used for each category as the training and test set, respectively, as shown in Table 1. The whole research was named as a three-way 10-shot few-shot Learning task.

3.2 | Comparative experiments

To investigate the accuracy of 10 improved models, comparative experiments were designed. As illustrated in the Introduction, the improvements of SN were mainly focused on three aspects: lightweight, deepening, and widening effects. For the original SN and each improved SN, they were trained under the same conditions. To reduce the negative influences of errors caused by randomness, each training SN experiment was repeated three times, and the models obtained from the three experiments were used for HSR defect classification. The test time and accuracy of each classification experiment were recorded, and the average model results of the original SN and each improved SN were calculated. Finally, by comparing the parameter amount, average test time, and average accuracy of different networks, the optimal improved SN was determined.

As shown in Table 2, the original SN and a total of 10 improved SNs were developed. Test-1 corresponds to the original SN composed of a 7×7 convolution layer, an inception module, and an FC layer with 100 output nodes. Test-2 corresponds to the lightweight SN-1, which changes the output nodes of the FC layer to 10 and adds two FC layers with output nodes of 10 based on the original SN. Test-3 corresponds to the lightweight SN-2, which adds a 1×1 convolution layer based on the original SN. Test-4 corresponds to the deepened SN-1, which adds two 1×1 convolution layers based on the lightweight SN-2. Test-5 corresponds to the deepened SN-2, which adds three 3×3 convolution layers based on the lightweight SN-2. Test-6 corresponds to the deepened SN-3, which adds three 3×3 maximum pooling layers based on the deepened SN-2. Test-7 corresponds to the deepened SN-4, which adds three 3×3 convolution layers based on the deepened SN-3. Test-8 corresponds to the deepened SN-5, which adds an FC layer with 100 output nodes based on the original SN. Test-9 corresponds to the deepened SN-6, which adds an inception module based on the original SN. Test-10 corresponds to the widened SN-1, which adds a parallel branch of 1×1 and 7×7 convolution layers based on the original SN. Test-11 corresponds to the widened SN-2, which adds a 1×1 convolution layer based on the widened SN-1. The structure of each test and its corresponding network are shown in Table 2.

The input data processed by the SN are in the form of sample pairs. Therefore, before training the SN, the HSR defect data in the training set have been constructed into sample pairs. Positive sample pairs consisted of two images from the same category, while negative sample pairs consisted of two images from different categories. There were 135 positive sample pairs and 300 negative sample pairs. All sample pairs were randomly divided into two parts for training and validation of the SN, comprising 217 sample pairs and 218 sample pairs, respectively.

The SN in this study measured the distance between different feature vectors using the Euclidean distance function and then calculated the similarity of sample pairs. The distance between positive sample pairs was very small, whereas the distance between negative sample pairs was large. The contrastive loss was used to determine whether the output of the SN matched the labels of the sample pairs. The loss was small when they matched and large when they did not, resulting in a greater adjustment of network parameters.

The root mean square propagation optimizer was used to accelerate the optimization, with a learning rate of 10^{-5}. The model was trained for 10 epochs with a batch size of 16. All other parameters were kept at their default settings.

3.3 | Test environment

The tests were conducted on a mobile workstation with an Intel Core i9-9880H Central Processing Unit (CPU) @2.30 GHz, featuring 16 cores and 32 GB of RAM. Python was used to implement all the models, utilizing the TensorFlow library.

TABLE 2 Comparative tests and the corresponding network structures.

Test	Improved method	Model	Structure
1	/	Original Siamese network (SN)	conv(7) + Inception + dense(100)
2	Lightweight structure	Lightweight SN-1	conv(7) + Inception + 3*dense(10)
3		Lightweight SN-2	conv(7) + Inception + conv(1) + dense(100)
4	Deepen network	Deepened SN-1	conv(7) + Inception + 3*conv(1) + dense(100)
5		Deepened SN-2	conv(7) + Inception + conv(1) + 3*conv(3) + dense(100)
6		Deepened SN-3	conv(7) + Inception + conv(1) + 3*conv(3)maxpool(3) + dense(100)
7		Deepened SN-4	conv(7) + Inception + conv(1) + 2*conv(3)maxpool(3) + 3*conv(3) + conv(3)maxpool(3) + dense(100)
8		Deepened SN-5	conv(7) + Inception + 2*dense(100)
9		Deepened SN-6	conv(7) + 2*Inception + dense(100)
10	Widen network	Widened SN-1	conv(7) + Inception(&conv(1)conv(7)) + dense(100)
11		Widened SN-2	conv(7) + Inception(&conv(1)conv(7)) + conv(1) + dense(100)

4 | RESULTS AND DISCUSSION

4.1 | Analysis of the model training process

Analyzing the variation of the loss function during the model training process holds vital significance in assessing whether the model is underfitting or overfitting (Rosso, Marasco et al., 2023; Zhou et al., 2020). To ensure the absence of underfitting or overfitting issues in the proposed model, a thorough analysis of the loss function curves during the training processes for various SNs is conducted as depicted in Figure 15. The loss function of the original SN exhibits instability due to its generic nature, not being explicitly designed for HSR defect detection. Additionally, the loss function of deepened SN-6 remains unstable due to its excessive structural complexity. In contrast, the remaining nine SN models introduced in this paper exhibit stable convergence throughout the training epochs. Among these, lightweight SN-1, lightweight SN-2, deepened SN-1, deepened SN-2, widened SN-1, and widened SN-2 demonstrate rapid convergence rates, while deepened SN-3 and deepened SN-4 exhibit slower convergence rates due to the trade-off between stability enhancement and learning speed introduced by pooling layers. Overall, the training of the proposed models in this paper proves to be both stable and effective.

4.2 | Analysis of accuracy

The most crucial indicator for intelligent recognition of HSR defects is accuracy, which determines the effectiveness of the neural work and is therefore evaluated first. The output of the SN was the distance between sample pairs, which could be adjusted to obtain the classification results, and the classification accuracy was then calculated. The simplest adjustment method was to calculate the average distance between each test data and each class of training data by constructing sample pairs. The training data class with the smallest average distance was the category of the test data. Then, the accuracy of classification was judged based on the label. This method had the advantage of simple operation and no changes required to the structure of the SN. The training of each SN was repeated three times. The modeling results obtained from the three repetitions were tested separately. The average accuracy was calculated. The arithmetic mean of all three categories' accuracies was used to represent the average classification accuracy as shown in Table 3.

Based on the above results, the original SN achieved an average classification accuracy of 64.4% for HSR defects. Compared to the original network, all improved SNs showed increased classification accuracy, except for deepened SN-5, which may be due to having too many nodes in the FC layers, resulting in overfitting due to excessive model parameters and computational complexity. The highest performing improved SNs were deepened SN-3 and deepened SN-4, with both achieving an accuracy of 73.9%, indicating that deepening the network by adding 3×3 convolution layers and maximum pooling layers can improve classification accuracy for HSR defects, while adding too many 3×3 convolution layers will not further improve classification accuracy. The second-highest performing improved SN was the widened SN-2, which achieved 71.7% accuracy, demonstrating that increasing the width of the inception module can improve feature extraction capabilities. Interestingly, both lightweight SNs also surpassed the original SN in accuracy because they increased the network depth while still being lightweight designed. It is noted that the 1×1 convolution layer had a positive effect on improving the classification accuracy of

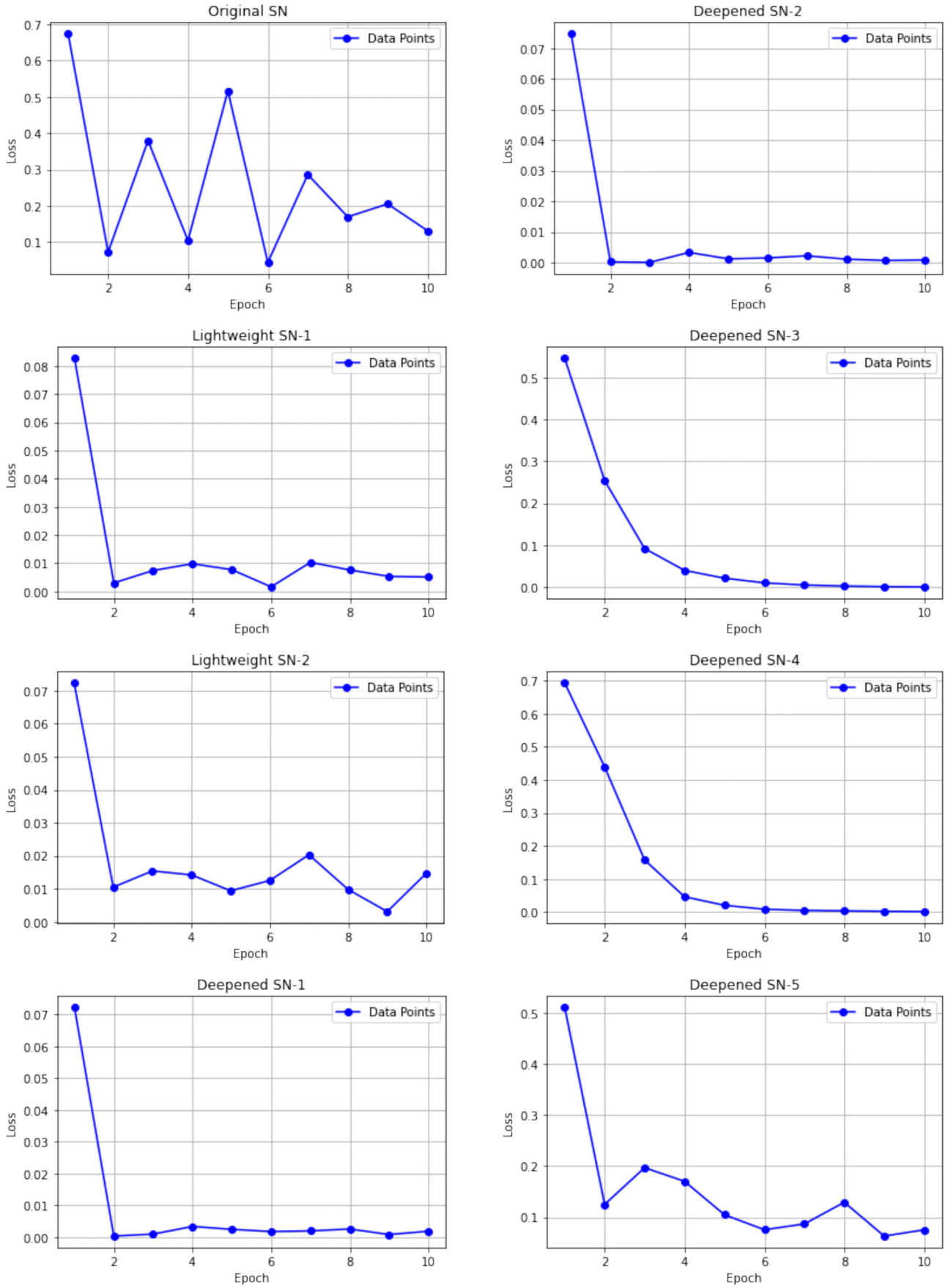

FIGURE 15 Training loss curves of various SNs.

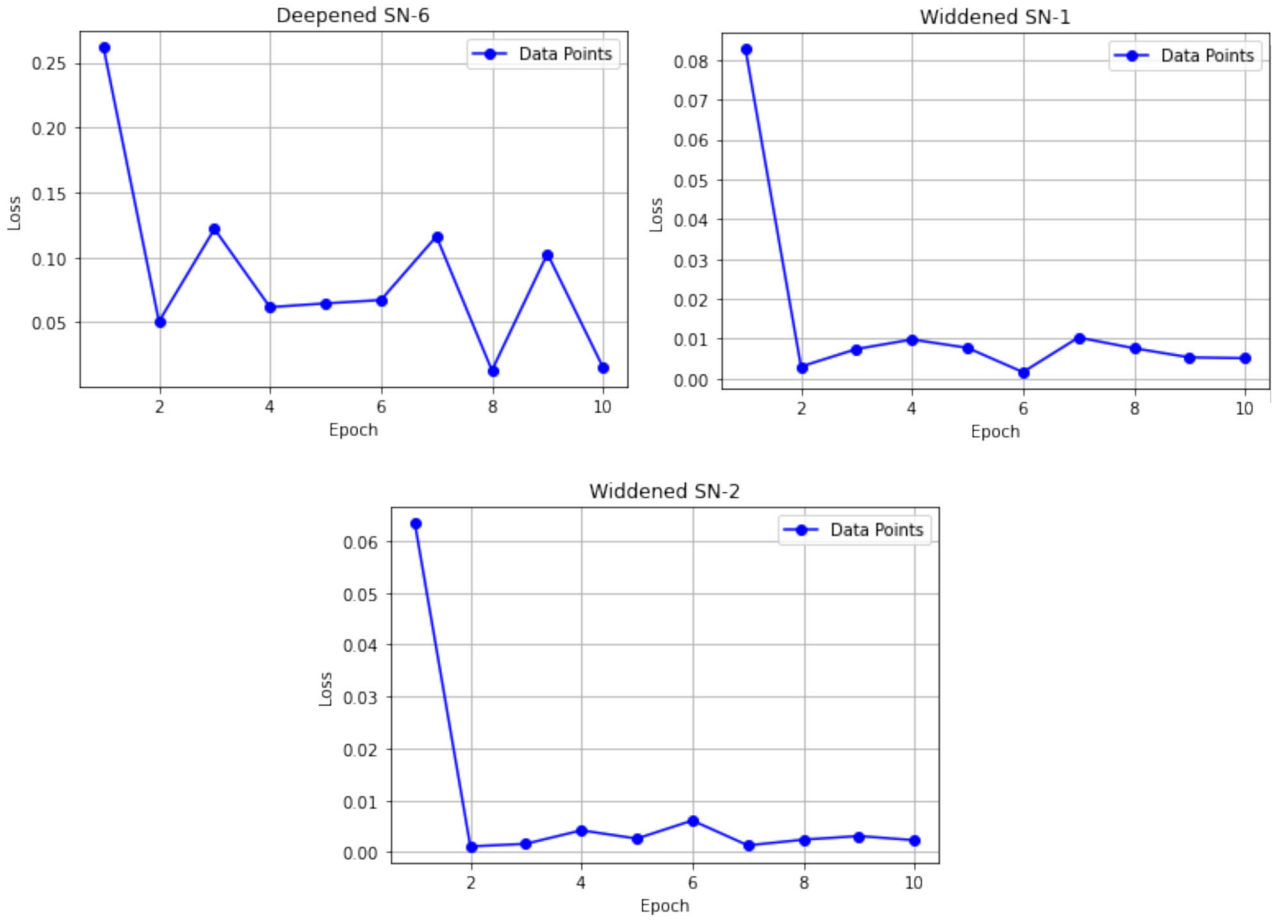

FIGURE 15 Continued

TABLE 3 The classification accuracy of high-speed railway defects.

Test	Model	Accuracy			
		Slab cracks	Track board gaps	Unbroken state	Average
1	Original SN	0.7	0.633	0.6	0.644
2	Lightweight SN-1	0.633	0.717	0.65	0.667
3	Lightweight SN-2	0.683	0.7	0.667	0.683
4	Deepened SN-1	0.683	0.65	0.767	0.7
5	Deepened SN-2	0.65	0.667	0.75	0.689
6	Deepened SN-3	0.65	0.583	0.983	0.739
7	Deepened SN-4	0.633	0.583	1	0.739
8	Deepened SN-5	0.667	0.667	0.567	0.634
9	Deepened SN-6	0.7	0.617	0.767	0.695
10	Widened SN-1	0.683	0.667	0.7	0.683
11	Widened SN-2	0.683	0.65	0.817	0.717

the SN for HSR defects, as evidenced by the comparisons among Test-1, Test-3, and Test-4, which only differed in whether or not they added 1×1 convolution layers and how many were added. Additionally, deepened SN-2 and deepened SN-6 also showed improvements in classification accuracy for HSR defects.

The confusion matrices of the top three improved SNs with the highest classification accuracy were analyzed and shown in Figure 16. Deepened SN-3 and deepened SN-4 can accurately distinguish images from unbroken state but have a higher probability of misclassifying images with slab cracks or track board gaps. Widened SN-2 had similar

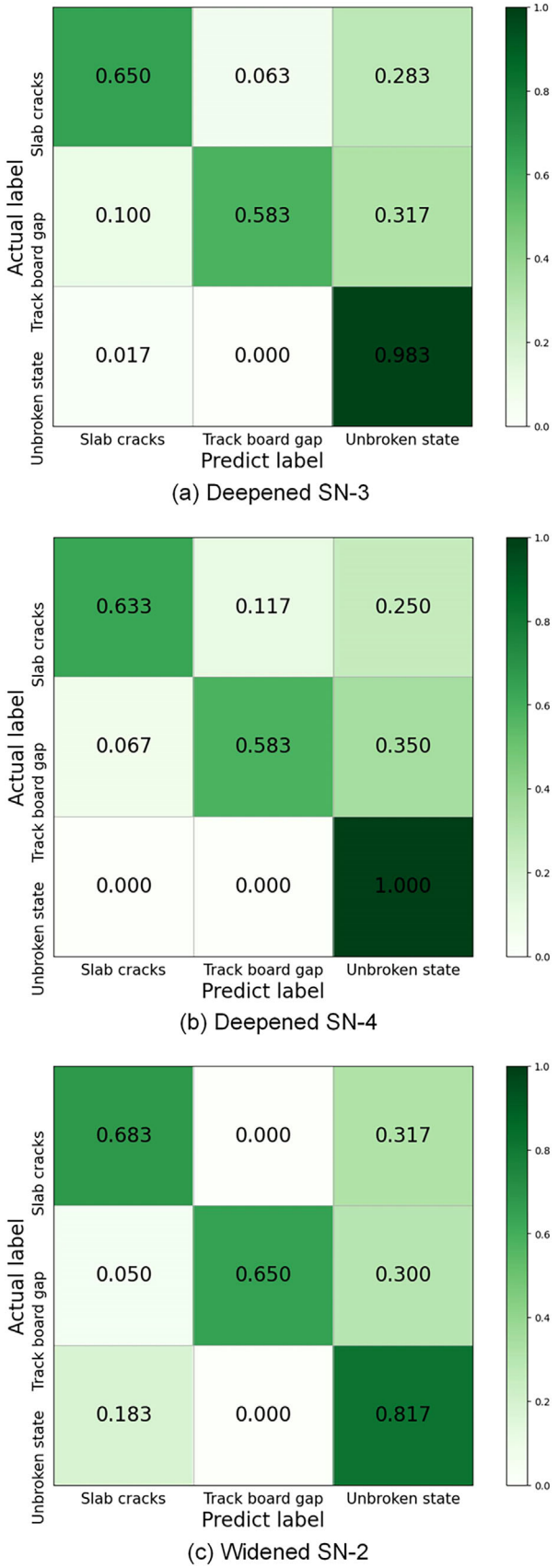

FIGURE 16 Confusion matrices of the top-3 improved SNs with the highest classification accuracy.

TABLE 4 The parameter quantity and testing time for each SN.

Test	Model	Parameter quantity	Testing time
1	Original SN	642,416,420	2 min 41 s
2	Lightweight SN-1	64,389,030	2 min 12 s
3	Lightweight SN-2	160,792,484	3 min 27 s
4	Deepened SN-1	160,825,508	4 min 14 s
5	Deepened SN-2	161,235,236	5 min 51 s
6	Deepened SN-3	2,835,236	4 min 18 s
7	Deepened SN-4	3,277,988	4 min 25 s
8	Deepened SN-5	642,426,520	2 min 41 s
9	Deepened SN-6	642,720,100	6 min 1 s
10	Widened SN-1	803,182,532	3 min 8 s
11	Widened SN-2	161,011,780	3 min 56 s

accuracy for all three categories and was more suitable for distinguishing between track board gaps and slab cracks. It should be noted that as this is very preliminary research focusing on using limited dataset for distress detection in HSR infrastructures, there exist limitations of this method, for example, not very significant improvement in the accuracy as we care more about the reduction of computation time. Meanwhile, the elapsed time may not represent a fair metric to compare or account for the computational effort. These limitations will be considered and addressed in future research.

4.3 | Analysis of lightweight degree

This study evaluated two aspects of lightweight models: parameter quantity and test time. Parameter quantity reflects the difficulty of model training and storage volume. Models with fewer parameters are normally easier to train and occupy less storage. Test time represents the time it takes to identify each HSR defect image, and the shorter the testing time, the faster the repair work and maintenance by railway engineers. Therefore, lightweight models with fewer parameters and shorter testing times are considered to be optimal. Table 4 shows the parameter quantity and testing time for each SN, where test time refers to the total time for testing all the data in the test set. For each SN trained three times separately, its parameter number was fixed, and its testing time was expressed as the average of the three tests.

The experiment results of the degree of lightweight models showed that the proposed methods effectively reduced the parameter quantity and test time for HSR defect classification using few-shot learning networks. The lightweight SN-1 and lightweight SN-2 effectively reduced the parameter quantity by decreasing the output nodes of the FC layer and the input nodes of it, respectively. The improved

TABLE 5 Comprehensive comparative of the improved SNs.

Model	Accuracy (%)	Increase (%)	Parameter quantity ($\times 10^6$)	Decrease (%)	Testing time (s)	Decrease (%)
Original SN	64.4	/	642.4	/	161	/
Lightweight SN-1	66.7	2.3	64.4	90.0	132	18.0
Lightweight SN-2	68.3	3.9	160.8	75.0	207	−28.6
Deepened SN-1	70.0	5.6	160.8	75.0	254	−57.8
Deepened SN-2	68.9	4.5	161.2	74.9	351	−118.0
Deepened SN-3	73.9	9.5	2.8	99.6	258	−60.2
Deepened SN-4	73.9	9.5	3.3	99.5	265	−64.6
Deepened SN-5	63.4	−1.0	642.4	0	161	0
Deepened SN-6	69.5	5.1	642.7	−0.05	361	−124.2
Widened SN-1	68.3	3.9	803.2	−25.0	188	−16.8
Widened SN-2	71.7	7.3	161.0	74.9	236	−46.6

SN with the least parameter quantity was the deepened SN-3, with approximately 2.8 million parameters. It incorporated three 3×3 maximum pooling layers with stride 2×2, which reduced the size of feature maps and significantly reduced the input of FC layers. The deepened SN-4 also used three 3×3 maximum pooling layers with stride 2×2, resulting in a small number of parameters, but more than the deepened SN-3 due to the additional convolution layers. The improved SN with the most parameters was the widened SN-1, which showed the negative effect of widening the inception module. However, adding a 1×1 convolution layer while widening the inception module can solve this problem as seen in the comparisons between Test-11 and Test-10 and in comparison with Test-1.

For the evaluation of reducing test time, the lightweight SN-1 achieved positive results as it saved computation resources in the FC layer. However, the test time for the lightweight SN-2 increased due to the computational cost of the 1×1 convolution layer exceeding the savings from the FC layer. The deepened and widened SN had higher testing times than the original SN, as expected, since more layers inevitably brought more computational costs. However, note that the test time of deepened SN-5 was equal to that of the original SN. This was because it only added an FC layer with very few input and output nodes, resulting in a minimal increase in computational cost. It should be noted that the SN with the lowest parameter quantity may not necessarily have the shortest test time, as most of the parameters were concentrated in the FC layers while the convolution layers carried out most of the computations.

4.4 | Comprehensive comparative analysis

Table 5 provides a comprehensive analysis of different SNs, including accuracy, parameter quantity, testing time, and changes based on the original SNs. Note that the increase in accuracy referred to the absolute increase in the numerical value, while the decreases in parameter quantity and test time were relative.

Based on the comprehensive comparison, it can be concluded that the deepened SN-3 and the deepened SN-4 achieved the highest classification accuracy for HSR defect recognition, both reaching 73.9%. Furthermore, the deepened SN-3 also had the least number of parameters, reducing it by 99.6%, compared to the original SN. The SN with the shortest test time was the lightweight SN-1, which only took 132 s to complete all tests, equivalent to 2.2 s per image. It should be noted that the computation is lightweight structure-based, indicating that civil engineers can conveniently perform HSR defect recognition without relying on large indoor computation equipment. Additionally, if engineers have requirements for both accuracy and training time, the widened SN-2 is an optimal model.

5 | CONCLUSION

This paper proposed a method for intelligent recognition of HSR infrastructure defects based on lightweight SN. This method can classify three categories of images: slab cracks, track board gaps, and unbroken state, based on a very small dataset. Overall, 10 improved SNs were proposed to increase accuracy, reduce parameter quantity, and shorten test time. Results showed that three improved SNs had optimal performances:

1. If the priority is focused on HSR defect classification accuracy or using a network with the least parameter quantity, the deepen-designed SN-3 is the optimal model with an accuracy of 73.9% and only 2.8 million parameters.

2. If the priority is focused on the computation speed of the HSR defect recognition model, the lightweight-designed SN-1 is the optimal model, which can complete recognition in 2.2 s with an accuracy of 66.7%.
3. In the priorities are focused on both accuracy and computation speed of the model, the widen-designed SN-2 is the optimal model, with faster speed than the deepened SN-3 and higher accuracy than the lightweight SN-1, although its 161 million parameters occupy more storage space.

Overall, the optimum model can fast and satisfactorily recognize the defects in HSR with a very limited data size of 10 samples for each training category. The methods proposed in this paper provide an effective tool for rail engineers engaged in HSR operation and maintenance work, enabling HSR defect recognition to be done automatically, conveniently, and quickly. The model based on few-shot learning also provides a new idea for the development of the railway inspection field.

It should be noted that this study also has some shortcomings. First, as it is a preliminary study, there were only three categories of HSR service conditions studied. Second, the dataset used was limited. Third, as an innovative preliminary study, the accuracy achieved in this paper is not particularly high.

Future research will take these issues into consideration and investigate newer supervised machine learning/classification algorithms, such as neural dynamic classification algorithm (Rafiei & Adeli, 2017b), dynamic ensemble learning algorithm (Alam et al., 2020), finite element machine for fast learning (Pereira et al., 2020), and self-supervised learning (Rafiei et al., 2023), in order to continue to explore and advance the field of automated defect detection in HSR infrastructures.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China No. 52178405, Project of Science and Technology Research and Development Program of China State Railway Group Co., Ltd. No. SY2022T002; Open Fund of National Key Laboratory of High-speed Railway Track Technology No. 2021YJ053.

REFERENCES

Adeli, H. (2001). Neural networks in civil engineering: 1989–2000. *Computer-Aided Civil and Infrastructure Engineering*, *16*(2), 126–142.

Adeli, H., & Kamal, O. (1989). Parallel structural analysis using threads. *Computer-Aided Civil and Infrastructure Engineering*, *4*(2), 133–147.

Adeli, H., & Park, S. (1995). Counterpropagation neural networks in structural engineering. *Journal of Structural Engineering*, *121*(8), 1205–1212.

Ahmadkhanlou, F., & Adeli, H. (2005). Optimum cost design of reinforced concrete slabs using neural dynamics model. *Engineering Applications of Artificial Intelligence*, *18*(01), 65–72.

Ai, C., Qiu, S., Xu, G., Zhang, A., & Wang, K. C. P. (2018). A nonballasted rail track slab crack identification method using a level-set-based active contour model. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 571–584.

Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing with Applications*, *32*(10), 8675–8690.

Aldwaik, M., & Adeli, H. (2016). Cost optimization of reinforced concrete flat slabs of arbitrary configuration in irregular highrise building structures. *Structural and Multidisciplinary Optimization*, *54*(01), 151–164.

Bang, S., Hong, Y., & Kim, H. (2022). Proactive proximity monitoring with instance segmentation and unmanned aerial vehicle-acquired video-frame prediction. *Computer-Aided Civil and Infrastructure Engineering*, *36*(6), 800–816.

Bertinetto, L., Henriques, J. F., Valmadre, J., Torr, P., & Vedaldi, A. (2016). Learning feed-forward one-shot learners. *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*, Barcelona, Spain (pp. 523–531).

Bromley, J., Guyon, I., Lecun, Y., Sckinger, E., & Shah, R. (1993). Signature verification using a Siamese time delay neural network. *Advances in Neural Information Processing Systems 6 (7th NIPS Conference)*, Denver, CO.

Cao, Z., Qin, Y., Xie, Z., Liu, Q., Zhang, E., Wu, Z., & Yu, Z. (2022). An effective railway intrusion detection method using dynamic intrusion region and lightweight neural network. *Measurement*, *191*, 110564.

Chaurasia, A., & Culurciello, E. (2017). LinkNet: Exploiting encoder representations for efficient semantic segmentation. *IEEE Visual Communications and Image Processing*, St. Petersburg, FL (pp. 1–4).

Chen, F. C., & Jahanshahi, R. (2018). NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion. *IEEE Transactions on Industrial Electronics*, *65*(99), 4392–4400.

Chen, J., Liu, Z., Wang, H., Nunez, A., & Han, Z. (2018). Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network. *IEEE Transactions on Instrumentation and Measurement*, *67*(2), 257–269.

Chen, N., Xu, Z., Liu, Z., Chen, Y., Miao, Y., Li, Q., Hou, Y., & Wang, L. (2022). Data augmentation and intelligent recognition in pavement texture using a deep learning. *IEEE Transactions on Intelligent Transportation Systems*, *23*(12), 25427–25436.

Chopra, S., Hadsell, R., & Lecun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA (pp. 539–546). https://doi.org/10.1109/CVPR.2005.202

Choudhary, T., Mishra, V., Goswami, A., & Sarangapani, J. (2020). A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, *53*(3), 5113–5155.

Chun, P. J., Shimamoto, Y., Okubo, K., Miwa, C., & Ohga, M. (2017). Deep learning and random forest based crack detection from

an image of concrete surface. *Journal of Japan Society of Civil Engineers Ser F3*, *73*(2), 297–307.

Chun, P. J., Ujike, I. M., Mishima, K., Kusumoto, M., & Okazaki, S. (2020). Random forest-based evaluation technique for internal damage in reinforced concrete featuring multiple nondestructive testing results. *Construction and Building Materials*, *253*, 105310.

Chun, P. J., & Hayashi, S. (2021). Development of a concrete floating and delamination detection system using infrared thermography. *IEEE/ASME Transactions on Mechatronics*, *26*(6), 2835–2844.

Chun, P. J., Yamane, T., & Maemura, Y. (2021). A deep learning-based image captioning method to automatically generate comprehensive explanations of bridge damage. *Computer-Aided Civil and Infrastructure Engineering*, *37*(11), 1387–1401.

Chun, P. J., Suzuki, M., & Kato, Y. (2023). Iterative application of generative adversarial networks for improved buried pipe detection from images obtained by ground-penetrating radar. Computer-Aided Civil and Infrastructure Engineering, 1–19. https://doi.org/10.1111/mice.13070

Clark, M., Gordon, M., & Forde, M. (2003). Issues over high-speed non-invasive monitoring of railway trackbed. *NDT & E International*, *37*(2), 131–139.

Deng, L., Li, G., Han, S., Shi, L., & Xie, Y. (2020). Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, *108*(4), 485–532.

Fan, D. P., Zhou, T., Ji, G. P., Zhou, Y., & Shao, L. (2020). Inf-Net: Automatic COVID-19 lung infection segmentation from CT images. *IEEE Transactions on Medical Imaging*, *39*(8), 2626–2637.

Feng, M., Zheng, J., Ren, J., Hussain, A., & Liu, Q. (2019). Big data analytics and mining for effective visualization and trends forecasting of crime data. *IEEE Access*, *7*(99), 106111–106123.

Gao, L. (2015). *Track engineering*. China Railway Publishing.

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 748–768.

Gibert, X., Patel, V. M., & Chellappa, R. (2015). Robust fastener detection for autonomous visual railway track inspection. *IEEE Winter Conference on Applications of Computer Vision*, Waikoloa, HI (pp. 694–701).

Gibert, X., Patel, V. M., & Chellappa, R. (2017). Deep multitask learning for railway track inspection. *IEEE Transactions on Intelligent Transportation Systems*, *18*(1), 153–164.

Grabczewski, K., & Jankowski, N. (2007). Versatile and efficient meta-learning architecture: Knowledge representation and management in computational intelligence. *IEEE Symposium on Computational Intelligence & Data Mining*, Honolulu, HI (pp. 51–58).

Guo, B., Geng, G., Zhu, L., Shi, H., & Yu, Z. (2019). High-speed railway intruding object image generating with generative adversarial networks. *Sensors*, *19*(14), 3075.

Guo, F., Qian, Y., Wu, Y., Leng, Z., & Yu, H. (2021). Automatic railroad track components inspection using real-time instance segmentation. *Computer-Aided Civil and Infrastructure Engineering*, *36*(3), 362–377.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV (pp. 770–778).

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Hu, W., Wang, W., Ai, C., Wang, J., Wang, W., Meng, X., Liu, J., Tao, H., & Qiu, S. (2021). Machine vision-based surface crack analysis for transportation infrastructure. *Automation in Construction*, *132*, 103973.

Huang, J., Yang, X., Zhou, F., Li, X., Zhou, B., Lu, S., Ivashov, S., Giannakis, I., Kong, F., & Slob, E. (2023). A deep learning framework based on improved self-supervised learning for ground-penetrating radar tunnel lining inspection. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.13042

Jelodar, H., Wang, Y., Orji, R., & Huang, H. (2020). Deep sentiment classification and topic discovery on novel coronavirus or COVID-19 online discussions: NLP using LSTM recurrent neural network approach. *IEEE Journal of Biomedical and Health Informatics*, *24*(10), 2733–2742.

Kang, D., & Cha, Y.-J. (2018). Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging, *Computer-Aided Civil and Infrastructure Engineering*, *33*(10), 885–902.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2013). One-shot learning by inverting a compositional causal process. In C. J. C. Burges, L. Bottou, & M. Welling (Eds.), *NIPS'13: Proceedings of the 26th international conference on neural information processing systems–Vol. 2*, (pp. 2526–2534). Curran Associates Inc.

Li, J., Zhou, F., & Ye, T. (2018). Real-world railway traffic detection based on faster better network. *IEEE Access*, *6*, 68730–68739.

Li, X., Huang, Y., Zhou, Y., Wang, X., Zhao, D., & Chen, C. (2023). Predicting the life cycle durability and ecological performance of cementitious coatings with a fuzzy-stochastics–based approach. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.13048

Li, Y., Bao, T., Huang, X., Chen, H., Xu, B., Shu, X., Zhou, Y., Cao, Q., Tu, J., Wang, R., & Zhang, K. (2022). Underwater crack pixel-wise identification and quantification for dams via lightweight semantic segmentation and transfer learning. *Automation in Construction*, *144*, 104600.

Li, Y., Dong, H., Li, H., Zhang, X., Zhang, B., & Xiao, Z. (2020). Multi-block SSD based on small object detection for UAV railway scene surveillance. *Chinese Journal of Aeronautics*, *33*(6), 1747–1755.

Liu, Z., Pan, S., Gao, Z., Chen, N., Li, F., Wang, L., & Hou, Y. (2023). Automatic intelligent recognition of pavement distresses with limited dataset using generative adversarial networks. *Automation in Construction*, *146*, 104674.

Liu, Z., Wang, L., Yang, C., & Han, Z. (2018). A high-precision loose strands diagnosis approach for isoelectric line in high-speed railway. *IEEE Transactions on Industrial Informatics*, *14*(3), 1067–1077.

Luo, J., Si, W., & Deng, Z. (2022). New classes inference, few-shot learning and continual learning for radar signal recognition. *IET Radar, Sonar & Navigation*, *16*(10), 1641–1655.

Marino, F., Distante, A., Mazzeo, P. L., & Stella, E. (2007). A real-time visual inspection system for railway maintenance: Automatic hexagonal-headed bolts detection. *IEEE Transactions on Systems Man & Cybernetics Part C*, *37*(3), 418–428.

Marano, G. C., Rosso, M. M., Aloisio, A., & Cirrincione, G. (2023). Generative adversarial networks review in earthquake-related

engineering fields. *Bulletin of Earthquake Engineering*. Advance online publication. https://doi.org/10.1007/s10518-023-01645-7

Nurmikolu, A. (2012). Key aspects on the behaviour of the ballast and substructure of a modern railway track: Research-based practical observations in Finland. *Journal of Zhejiang University Science A (Applied Physics & Engineering)*, *13*(11), 825–835.

Ouyang, A., Vanessa, D. M., Martin, C., Roddy, C., John, A. O., & Li, Z. (2023). Automated pixel-level crack monitoring system for large-scale underground infrastructure—A case study at CERN. *Tunnelling and Underground Space Technology*, *140*, 105310.

Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). ENet: A deep neural network architecture for real-time semantic segmentation. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*. https://doi.org/10.48550/arXiv.1606.02147

Pan, S., Yan, H., Liu, Z., Chen, N., Miao, Y., & Hou, Y. (2023). Automatic pavement texture recognition using lightweight few-shot learning. *Philosophical Transactions of the Royal Society A*, *381*(2254), 20220166.

Park, J.-W., Lee, K.-C., Sim, S.-H., Jung, H.-J., & Spencer, B. F. Jr. (2016). Traffic safety evaluation for railway bridges using expanded multisensor data fusion. *Computer-Aided Civil and Infrastructure Engineering*, *31*, 749–760.

Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing and Applications*, *32*(10), 6393–6404.

Rafiei, M. H., & Adeli, H. (2016). A novel machine learning model for estimation of sale prices of real estate units. *Construction Engineering and Management*, *142*(2), 04015066.

Rafiei, M. H., & Adeli, H. (2017a). NEEWS: A novel earthquake early warning system using neural dynamic classification and neural dynamic optimization model. *Soil Dynamics and Earthquake Engineering*, *100*, 417–427.

Rafiei, M. H., & Adeli, H. (2017b). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Rafiei, M. H., & Adeli, H. (2018). A novel machine learning model for construction cost estimation taking into account economic variables and indices. *Journal of Construction Engineering and Management*, *144*(12), 04018106.

Rafiei, M. H., Gauthier, L., Adeli, H., & Takabi, D. (2023). Self-supervised learning for electroencephalography. *IEEE Transactions on Neural Networks and Learning Systems*. Advance online publication. https://doi.org/10.1109/TNNLS.2022.3190448

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete compressive strength. *ACI Materials Journal*, *114*(2), 237–244.

Rosso, M. M., Aloisio, A., Randazzo, V., Tanzi, L., Cirrincione, G., & Marano, G. (2023). Comparative deep learning studies for indirect tunnel monitoring with and without Fourier pre-processing. *Integrated Computer-Aided Engineering*. Advance online publication. https://doi.org/10.3233/ICA-230709

Rosso, M. M., Marasco, G., Aiello, S., Aloisio, A., Chiaia, B., & Marano, G. C. (2023). Convolutional networks and transformers for intelligent road tunnel investigations. *Computers & Structures*, *275*(15), 106918.

Shajihan, S. A. V., Hoang, T. A., Mechitov, K., & Spencer, B. F. (2022). Wireless SmartVision system for synchronized displacement monitoring of railroad bridges. *Computer-Aided Civil and Infrastructure Engineering*, *37*, 1070–1088.

Sun, L., Yang, K., Hu, X., Hu, W., & Wang, K. (2020). Real-time fusion network for RGB-D semantic segmentation incorporating unexpected obstacle detection for road-driving images. *IEEE Robotics and Automation Letters*, *5*(4), 5558–5565.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA (pp. 1–9).

Tong, L., Jia, L., Geng, Y., Liu, K., Qin, Y., & Wang, Z. (2023). Anchor-adaptive railway track detection from unmanned aerial vehicle images. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.13004

Triantafillou, E., Zemel, R., & Urtasun, R. (2017). Few-shot learning through an information retrieval lens. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, Long Beach, CA (pp. 2252–2262).

Tsiflakos, K., & Owen, D. B. (1993). A decision support tool for the railway industry based on computer graphics and intuitive modelling techniques. *Computer-Aided Civil and Infrastructure Engineering*, *8*, 105–118.

Trosino, M., Cunningham, J. J., & Shaw, A. E., III. (2002). *Automated track inspection vehicle and method* (U.S. Patent No. US6064428-A). U.S. Patent and Trademark Office. https://patents.google.com/patent/US6064428A/en?oq=U.S.+Patent+No.+US6064428%E2%80%90A

Val, D. V., Stewart, M. G., & Melchers, R. E. (2000). Life-cycle performance of RC bridges: Probabilistic approach. *Computer-Aided Civil and Infrastructure Engineering*, *15*, 14–25.

Wang, J., Liu, X.-Z., & Ni, Y.-Q. (2018). A Bayesian probabilistic approach for acoustic emission-based rail condition assessment. *Computer-Aided Civil and Infrastructure Engineering*, *33*, 21–34.

Wang, K., Cao, W., Su, Z., Wang, P., Zhang, X., Chen, L., Guan, R. Q., & Lu, Y. (2020). Structural health monitoring of high-speed railway tracks using diffuse ultrasonic wave-based condition contrast: Theory and validation. *Smart Structures and Systems*, *26*(2), 227–239.

Wang, Q., Gao, T., He, Q., Liu, Y., Wu, J., & Wang, P. (2023). Severe rail wear detection with rail running band images. *Computer-Aided Civil and Infrastructure Engineering*, *38*, 1162–1180.

Wang, W., Jiang, Z., Liao, J., Ying, Z., & Zhai, Y. (2022). Explorations of contrastive learning in the field of small sample SAR ATR. *Procedia Computer Science*, *208*, 190–195.

Wilson, J. L., Wagaman, S. J., Veshosky, D. A., Shi, C. G., Adury, P., & Beidleman, C. R. (1997). Life-cycle engineering of bridges. *Computer-Aided Civil and Infrastructure Engineering*, *12*, 445–452.

Xu, T., Yang, P., Zhang, X., & Liu, C. (2018). LightweightNet: Toward fast and lightweight convolutional neural networks via architecture distillation. *Pattern Recognition*, *88*, 272–284.

Xue, G., Liu, S., Ren, L., & Gong, D. (2023). Adaptive cross-scenario few-shot learning framework for structural damage detection in civil infrastructure. *Journal of Construction Engineering and Management*, *149*(5), 04023020. https://doi.org/10.1061/JCEMD4.COENG-13196

Yamane, T., Chun, P. J., Dang, J., & Honda, R. (2023). Recording of bridge damage areas by 3D integration of multiple images and reduction of the variability in detected results. *Computer-Aided Civil and Infrastructure Engineering*. Advance online publication. https://doi.org/10.1111/mice.12971

Yang, A., Yang, B., Ji, Z., Pang, Y., & Shao, L. (2019). Lightweight group convolutional network for single image super-resolution. *Information Sciences*, *516*, 220–233.

Yao, H., Xu, Z., Hou, Y., Dong, Q., Pei, X., Ye, Z., Liu, P., Wang, L., & Wang, D. (2023). Advanced industrial informatics towards smart, safe and sustainable roads: A state of the art. *Journal of Traffic and Transportation Engineering (English Edition)*, *10*(2), 143–158.

Ye, T., Wang, B., Song, P., & Li, J. (2018). Automatic railway traffic object detection system using feature fusion refine neural network under shunting mode. *Sensors*, *18*(6), 1916.

Ye, T., Zhang, Z., Zhang, X., & Zhou, F. (2020). Autonomous railway traffic object detection using feature-enhanced single-shot detector. *IEEE Access*, *8*, 145182–145193.

Ye, W., Deng, S., Ren, J., Xu, X., Zhang, K., & Du, W. (2022). Deep learning-based fast detection of apparent concrete crack in slab tracks with dilated convolution. *Construction and Building Materials*, *329*, 127157.

Ye, W., Ren, J., Zhang, A. A., & Lu, C. (2023). Automatic pixel-level crack detection with multi-scale feature fusion for slab tracks. *Computer-Aided Civil and Infrastructure Engineering*, 1–18. https://doi.org/10.1111/mice.12984

Yoon, H., Ham, Y., Golparvar-Fard, M., & Spencer, B. F. Jr. (2016). Forward-backward approach for 3D event localization using commodity smartphones for ubiquitous context-aware applications in civil and infrastructure engineering. *Computer-Aided Civil and Infrastructure Engineering*, *31*, 245–260.

Yuan, J., Guo, H., Jin, Z., Jin, H., Zhang, X., & Luo, J. (2017). One-shot learning for fine-grained relation extraction via convolutional Siamese neural network. *IEEE International Conference on Big Data*, Boston, MA (pp. 2194–2199).

Zhai, W. (2020). *Vehicle-track coupled dynamics: Theory and application*. Springer.

Zhai, W., & Zhao, C. (2016). Advanced technology and challenge of modern rail transit engineering. *Journal of Southwest Jiaotong University*, *51*(2), 209–226.

Zhai, W., Zhao, C., Xia, H., Xie, Y., Li, G., Cai, C., Luo, Q., & Song, X. (2014). Basic scientific issues on dynamic performance evolution of the high-speed railway infrastructure and its service safety. *Scientia Sinica (Technologica)*, *44*(7), 645–660.

Zhang, A., Li, S., Cui, Y., Yang, W., & Hu, J. (2019). Limited data rolling bearing fault diagnosis with few-shot learning. *IEEE Access*, *7*, 110895–110904.

Zheng, Q., Wang, R., Tian, X., Yu, Z., Wang, H., Abdussalam, E., & Sergio, S. (2023). A real-time transformer discharge pattern recognition method based on CNN-LSTM driven by few-shot learning. *Electric Power Systems Research*, *219*, 109241.

Zhong, J., Liu, Z., Han, Z., Ye, H., & Zhang, W. (2019). A CNN-based defect inspection method for catenary split pins in high-speed railway. *IEEE Transactions on Instrumentation and Measurement*, *68*(8), 2849–2860.

Zhou, F., Jin, L., & Dong, J. (2017). Review of convolutional neural networks. *Chinese Journal of Computers*, *40*(6), 1229–1251.

Zhou, X., Liang, W., Shimizu, S., Ma, J., & Jin, Q. (2020). Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, *17*(8), 5790–5798.

Zhu, S., & Cai, C. (2012). Research on the damage characteristics of double-block ballastless track bed slab under temperature and vehicle dynamic loads. *Zhongguo Tiedao Kexue/China Railway Science*, *33*(1), 6–12.

How to cite this article: Cai, X., Tang, X., Pan, S., Wang, Y., Yan, H., Ren, Y., Chen, N., & Hou, Y. (2023). Intelligent recognition of defects in high-speed railway slab track with limited dataset. *Computer-Aided Civil and Infrastructure Engineering*, 1–18. https://doi.org/10.1111/mice.13109