International Journal for Housing Science and Its Applications

Publish September 9, 2025. Volume 47, Issue 1 Pages 721-735

https://doi.org/10.70517/ijhsa47163

Research on Defect Detection Algorithm for Subway Bogies Based on Deep Convolutional Neural Networks

Baohua Jing[1,*] and Yufeng Jiang[2]

[1] School of Mechanical Engineering and Transportation, Changzhou Vocational Institute of Industry Technology, Changzhou, Jiangsu, 213164, China

[2] Crrc Changzhou Tech-mark Industrial Co., Ltd., Changzhou, Jiangsu, 213125, China

Corresponding authors: (e-mail: jbhjava@163.com).

Abstract In recent years, China's high-speed train sets have developed rapidly, with numerous trains operating across the nation's "Eight Vertical and Eight Horizontal" high-speed rail network, posing significant challenges for train operation, maintenance, and inspection. This paper establishes a defect detection plan based on subway bogie inspection standards, conducting inspection and analysis on components such as the axle box front cover and anti-roll torsion bar. Following the general bogie defect detection process, the inspection plan designed by the operation depot is implemented. A defect detection model is constructed using a convolutional neural network (CNN) algorithm. Through a three-stage detection process, defect detection in the bogie region is accomplished. Performance metrics are used to quantify the model's performance on the test dataset. On the test set v2, the predicted values for the three metrics—MAE, MAPE, and RMSE at a 5-step length—are 1.1125, 3.0421, and 1.9866, respectively, outperforming other models. Simultaneously, the model maintains tracking sensitivity above 90% during train emergency braking scenarios, demonstrating its high prediction accuracy.

Index Terms Convolutional Neural Network, Defect Object Detection, Tracking Sensitivity, Bogie Defect Detection

I. Introduction

With the rapid development of the urban rail transit industry, subways have become an indispensable mode of transportation for commuters [1], [2]. As the running gear of subway vehicles, the operational status of bogies directly impacts train safety [3]. A bogie primarily consists of the following components: the drive unit (comprising the motor and gearbox), the traction device (comprising the traction pin and center pin assembly), the braking system (comprising the brake pipe and brake caliper), the suspension system (comprising the air spring and primary spring assembly), the welded frame, and the wheelset axle box assembly [4]-[7].

As critical components, bogies are subject to stringent quality standards. During prolonged subway operations, environmental factors inevitably impact bogies, making timely condition assessment and maintenance essential for train safety [8]-[10]. Current subway bogie maintenance follows a preventive-focused approach, emphasizing preservation and implementing a phased, full-lifecycle maintenance model [11]-[12]. Maintenance is conducted in parallel based on both operational mileage and duration, typically categorized into: train inspections, biweekly inspections, quarterly inspections, bogie scheduled repairs, bogie frame repairs, and bogie overhauls. Operating entities establish distinct maintenance cycles according to varying operational environments and requirements [13]-[16]. However, traditional manual inspections suffer from low efficiency and high costs, issues effectively addressed by advancements in artificial intelligence [17], [18]. Deep convolutional neural networks (DCNNs) represent a widely adopted deep learning algorithm within AI. Capable of processing complex image data, they find extensive application across various fields for defect detection, including industrial product inspection, mining equipment diagnostics, and building surface flaw identification [19]-[22]. Similarly, DCNNs prove suitable for subway bogie defect detection, where they classify anomalies by learning characteristic features from training datasets [23]-[25]. First, deep convolutional layers extract features from input images using different convolution kernels, enhancing detection efficiency and accuracy [26], [27]. Subsequently, pooling layers reduce computational load by dimensionality reduction on convolutional layer outputs [28]. Finally, fully connected layers classify output vectors to detect defects in subway bogies [29], [30].

Reference [31] investigates strength detection methods for subway vehicle bogies. Focusing on a new bogie structure, simulation experiments validate that this strength detection approach outperforms traditional frame strength detection methods. Reference [32] emphasizes that subway bogies are critical components ensuring stable vehicle operation. Based on a specific subway bogie model, it analyzes strength reliability during design and manufacturing processes to enhance the fatigue

reliability of bogie frames. Reference [33] proposes a dual-light image-based temperature detection method for bogies. This approach utilizes infrared thermal and visible light images for non-contact detection with high precision, demonstrating practical application value through testing. Reference [34] focuses on the motor hanger weld zones of bogie bodies, conducting research from practical maintenance and inspection perspectives. By detecting surface cracks and internal defects in bogies, it demonstrates that welding defects shorten bogie service life, providing insights for optimizing maintenance strategies. Reference [35] analyzes fundamental metro components and their inspection principles, proposing a comprehensive metro inspection system architecture and development plan. Reference [36] introduces a low-cost system for measuring bogie vibration and enabling rapid data acquisition. Experimental results indicate that the precise timing for maintenance execution depends on the distance recorded per bogie, though mechanical durability varies.

Reference [37] designed a fault detection algorithm combining Faster R-CNN and a single-class convolutional neural network for axle box cover mounting bolts in subways. Experimental validation demonstrated the algorithm's excellent speed and accuracy. Reference [38] constructed a geometric fault diagnosis model for subway tracks based on convolutional neural network models, analyzed the impact of structural hyperparameters on model performance, and revealed the model's effectiveness through case studies. Reference [39] reviews the application of convolutional neural networks (CNNs) in the railway maintenance industry. Based on this literature review, it outlines the latest techniques applying CNNs to railway track maintenance, summarizes related tasks and challenges, and describes diverse CNN applications within this scope, including rail surface defect identification and track component detection. Reference [40] proposes a novel crowd density estimation method based on deep convolutional neural networks, demonstrating its robust performance in practical applications. The aforementioned studies underscore the critical importance of bogie inspection for subway operational safety. They investigate aspects such as bogie strength assessment, temperature monitoring, and surface defect detection, while exploring the application of convolutional neural networks in bogie inspection.

This paper identifies key inspection locations on subway bogies. Following on-site project investigations, five specific inspection points were designed: axle box front cover, anti-roll torsion bar, brake caliper, and others. The general workflow for subway bogie defect detection was systematized, with inspection result reports automatically generated by the system. A deep convolutional neural network was introduced, utilizing fully connected layers for image dimensionality reduction and classification. Object detection based on R-CNN is achieved through three sequential stages: candidate region selection, CNN feature extraction, and classification with bounding box regression. Building upon this foundation, the spatial pyramid pooling layer, Fast R-CNN network, and Faster R-CNN network are proposed to optimize the original R-CNN model, thereby enhancing the localization accuracy of object detection. By collecting data related to subway train bogies and setting up simulation experiments, the effectiveness of the proposed algorithm is demonstrated by comparing it with five classical models in terms of prediction accuracy and sensitivity. Subsequently, the accuracy of the algorithm is validated through real-vehicle application cases.

II. Overall Design Plan for Defect Detection of Subway Bogies

II. A. Overall Framework of the Intelligent Inspection System for Subway Units

The bogie involves numerous inspection items distributed across various inspection locations, with each location requiring distinct checks. Analysis must be conducted based on inspection standards to design an inspection plan.

II. A. 1) Inspection Items and Requirements

Following an on-site inspection of the project location, an analysis was conducted on the inspection locations and content. Key challenges identified include: the large dimensions of the bogies, the extensive scope and numerous points requiring inspection, limited access to certain shooting positions, variations in material, color, and shape of some components, and the inclusion of measurement, identification, and visual inspection tasks among the inspection items.

(1) Axle Box Front Cover

This inspection involves analyzing the axle box front cover on the bogie. Requirements include:

- Correct orientation of anti-loosening steel wire ties with over 7 turns after securing
- Clear anti-loosening markings on fastening screws without misalignment
- Bolts fully tightened to 8.8 grade specifications
- Consistent batch numbers
- Rubber cover installed flush without lifting The rubber cover surface must be free of damage and wear, with the rubber chain securely fixed. The axle end position must be correct with no missing parts, and the wire bracket must be installed with the correct model.

(2) Anti-Roll Torsion Bar

Inspection requirements: Anti-loosening wire must be intact, bound in the correct direction, and have more than 7 turns after binding. Anti-loosening markings on fastening screws must be clear and not misaligned, with bolts fully tightened. Spring washers must be installed between the support bracket and fastening bolts. The washer opening must be less than half its

thickness. The torsion arm shim, disc spring assembly (with convex surface facing the hex bolt head), stop block, and bolts must be installed in the correct sequence. One end of each support bracket is triangular, and the other end is circular. Fastening bolts must be grade 8.8 with matching batch numbers. The horizontal difference between the upper surfaces of the two-end torsion arm process pins is ≤35mm.

(3) Brake Caliper

Inspection requirements: Anti-loosening markings on fastening screws are clear and not misaligned. Anti-loosening wire is bundled in the correct direction with over 7 turns after bundling. Bolts are fully tightened, and fastening bolts are grade 8.8. Batch numbers must match. In the released state, the clearance between the brake hose and caliper must not be less than 15mm. The clearance between two adjacent brake hoses must not be less than 15mm. The release cable spiral hose must show no cracks. Two black cable ties must be secured to the brake hose. The hose fitting on the caliper must face downward at an angle of 0 to 45 degrees relative to the horizontal.

(4) Component History Information

Identify component serial numbers or scan QR codes to retrieve component details. Store and remotely share this information to create history templates as required. Identify expired rubber components, lubricants, or other parts.

(5) Cables and Cable Ties

Inspection requirements: Cables must be in good condition with no tangling, burning, damage, or breakage. Clearance between sensor cables and frame edges must be 3–10 mm. Spacing between moving parts and wiring must exceed 40 mm. Cable ties must be correctly positioned, complete, and free from breakage.

II. A. 2) Inspection Equipment Layout

Due to their large size and weight, it is impractical to manually push bogies into and out of the inspection area at the start and end of inspections. Therefore, an automatic assist device has been designed to replace manual labor in moving the bogies. At the start of inspection, it automatically pushes the bogies into the inspection area, and upon completion, it pulls them out.

The bogie self-propulsion device comprises a large slide rail, large slide drive motor, small slide drive electric cylinder, swing arm support cylinder, bogie towing wheels, towing wheel telescopic cylinder, and bogie towing swing arm. The specific operating principle is as follows: The large slide is driven by a servo motor, reducer, gear, and rack mechanism. An additional telescopic shaft is incorporated into the drive structure to ensure the bogie can be transported along the entire length of the large slide guide rails. When the large slide table moves to the front end of the guide rail, the small slide table simultaneously moves to the rear end driven by the electric cylinder. The swing arm cylinder of the towing mechanism retracts, lowering the swing arm to a horizontal position. At this point, the bogie wheel telescopic cylinder retracts, pulling the bogie wheels inward to avoid the track support components.

II. B. Overall Operational Process for Bogie Defect Detection

After completing the mechanical structure design of the bogie and the visual inspection system design, the following outlines the general inspection process for the bogie. The operational flow of the inspection system is shown in Figure 1, with specific steps listed below:

Step1: Manually push the bogie into the inspection area. Position sensors emit signals until the bogie is stopped by the first-level stop barrier. Manually trigger the button to start the equipment.

Step2: The inspection equipment performs a self-check of all systems. If abnormalities are detected, manual troubleshooting is conducted. If no issues are found, the first-stage stop cylinder releases, and the inspection system proceeds to the next phase.

Step3: The bogie's automatic assist mechanism smoothly advances it into the inspection zone. Contact sensors along the track are sequentially triggered by the wheelsets until the second-stage stop is activated, causing the assist mechanism to immediately halt.

Step4: The automatic bogie assist device retracts to its origin. Three robots, coordinated with overhead or ground rails, perform bogie inspections. They verify correct installation of all components, check for loose fastening bolts, verify bolt grade and batch numbers, confirm washer installation, inspect for misaligned anti-loosening marks, and verify integrity and correct orientation of anti-loosening wire ties. They also measure bogie component dimensions.

Step5: During the collection of 2D image data and 3D point cloud data, computer vision processes and synchronously analyzes the acquired data. It analyzes and statistically evaluates inspection and measurement results while recording all process data of the equipment operation.

Step6: After the robot completes inspection of most bogie components in the designated area, the secondary stop cylinder releases. The transfer cart of the assist mechanism transports the bogie to the completed inspection zone, then pushes it to the designated area. The robot and track spindle return to their original positions.

Step7: The system aggregates the 2D and 3D inspection results of the bogie, compiles comprehensive statistics on the finished bogie's inspection outcomes, and details the theoretical specifications and actual conditions for each inspection item. It automatically generates an inspection results report.

Step8: The bogie's automatic assist device retracts to its original position, awaiting the next bogie to be pushed in.

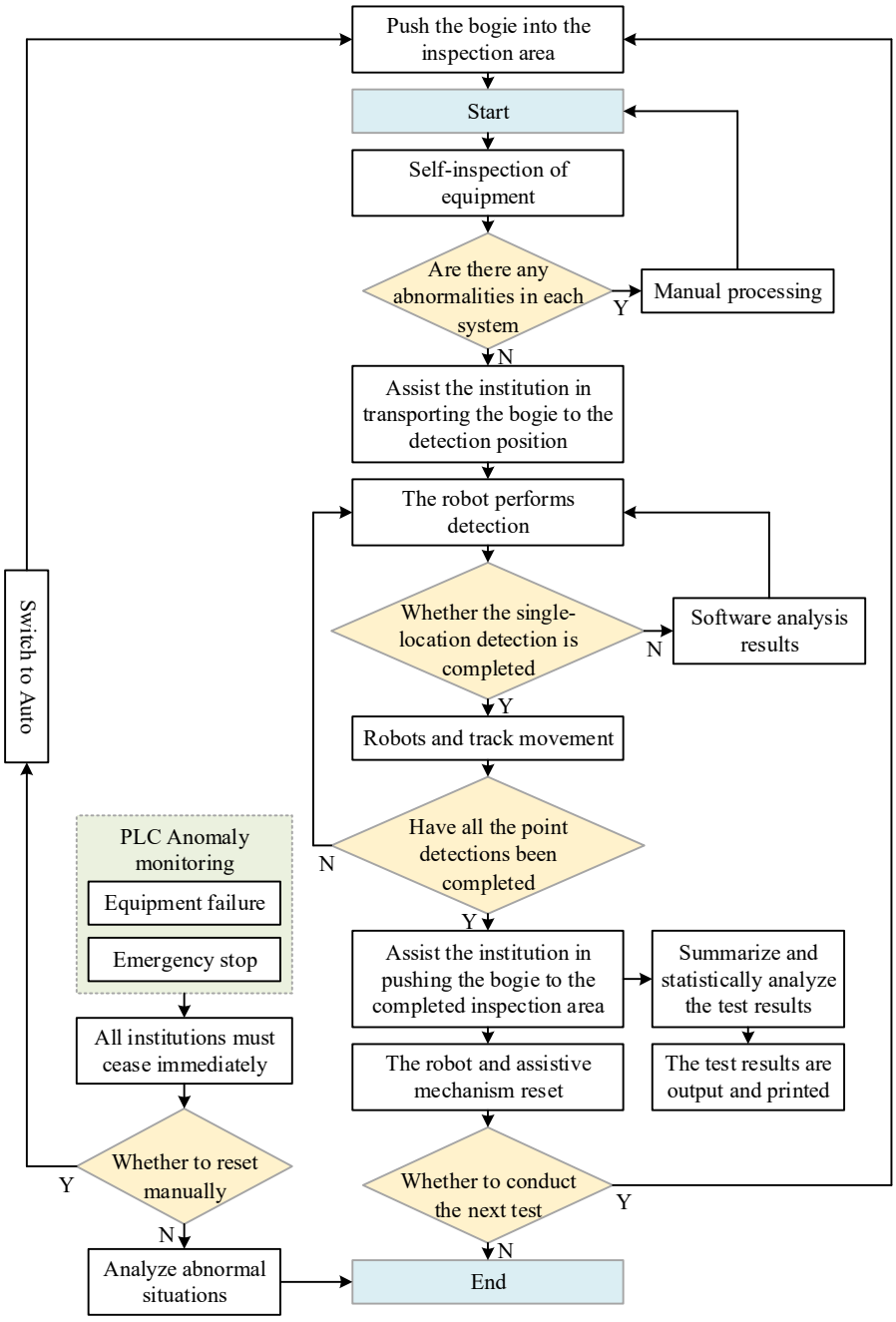

Figure 1: Test system operation process

III. Object Detection Based on Deep Convolutional Neural Networks

III. A. Convolutional Neural Networks

III. A. 1) Fully Connected Networks

For neural networks, neurons interact through connections between them, and the manner of these connections determines the network's complexity. Figure 2 depicts a fully connected neural network with three hidden layers. This network consists of three components: the input layer, hidden layers, and output layer. Data enters through the input layer, undergoes processing in the hidden layers, and finally exits via the output layer. As shown, in a fully connected network, every neuron in one layer is connected to every neuron in the next layer. This results in a highly complex network structure.

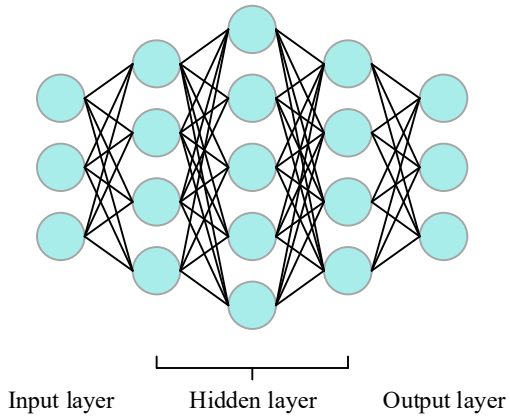

Figure 2: Structure Diagram of the fully connected neural network

III. A. 2) Convolutional Neural Network Algorithm

Figure 3 illustrates the convolutional neural network (CNN) algorithm. A CNN typically consists of multiple convolutional layers, multiple pooling layers, and multiple or a single fully connected layer [41]. The workflow of a convolutional neural network is as follows: First, the image is scanned using convolutional kernels within the convolutional layer to extract the spatial information of image features. Then, the image undergoes pooling operations through the pooling layer, reducing the size of the convolved image via downsampling to obtain abstract features. The image is subsequently subjected to dimensionality reduction through the fully connected layer, enabling final image classification.

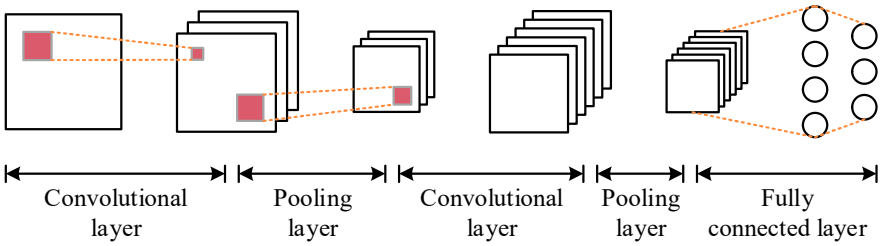

Figure 3: Basic Structure Diagram of Convolutional Neural Network

(1) Convolution Layer:

The convolution layer is a crucial structure in convolutional neural networks, enabling the abstraction of image features. By performing local perception on various features within an image, it reduces the receptive field of image features. The convolution kernel is the core component of the convolution layer, functioning as a filter. Before automatically extracting image features, the convolutional layer must first configure the kernel's parameters, including kernel size, number of kernels, and stride. As illustrated, the configured kernel is mapped onto the image and slid according to the specified stride. At each position, the kernel's weights are multiplied by the corresponding image data points, and the results are summed. This process can be expressed as:

$$\mathrm{con}\, v_{x,y} = \sum_{i}^{p*q} w_i v_i \tag{1}$$

Here, (x, y) represents the coordinates of the input image, $p \times q$ denotes the size of the convolution kernel, w_i is the weight value within the i th convolution kernel, and v_i is the pixel value of the image corresponding to the i th convolution kernel.

The formula for calculating the size of the convolved image is:

$$W = \frac{W - F + 2P}{S} + 1, H = \frac{H - F + 2P}{S} + 1 \tag{2}$$

(1) Convolution Layer

Here, W, H denote the image width and height, $F * F$ represents the convolution kernel size, S is the stride, and P indicates the padding method.

(2) Pooling Layer

Pooling performs downsampling on the image to reduce the resolution of the feature map, enabling the output feature map to capture more abstract characteristics. Pooling layers typically follow convolutional layers. After applying convolution to an image, feature maps are obtained, but the image parameters remain relatively numerous. Using these directly for training can easily lead to overfitting. Therefore, pooling operations are essential to compress the image while preserving quality, thereby reducing redundant detail parameters. The pooled image reduces its scale while preserving key information, significantly enhancing the network's generalization ability during training and improving model test accuracy.

The formula for calculating the size of the pooled image is:

$$W = \frac{W - F}{S} + 1, H = \frac{H - F}{S} + 1 \tag{3}$$

Analysis shows that after applying pooling operations to defect images, the image resolution decreases and defect features become more abstract. Therefore, for larger convolutional neural networks, the number of pooling layers should be limited.

(3) Fully Connected Layer

The fully connected layer is typically positioned as the final layer in a convolutional neural network [42]. It compresses images processed through multiple convolutions and pooling operations into one-dimensional information. The image data is highly abstracted into features with greater information content, which are then fed into the fully connected layer to perform image classification.

III. B. Object Detection Algorithms Based on Regional Proposals

III. B. 1) R-CNN

The R-CNN network performs object detection primarily through three stages: candidate region selection, CNN feature extraction, and classification with bounding box regression.

(1) Candidate Region Selection Stage

The goal of candidate region selection is to extract candidate regions containing potential objects from the input image. Traditional exhaustive methods for screening candidate boxes require evaluating every possible location, resulting in massive computational overhead and generating numerous redundant candidates. The selective search algorithm enhances computational efficiency by reducing redundant candidate regions. The R-CNN network model employs selective search for candidate region extraction. This algorithm primarily utilizes image segmentation and multi-scale processing to segment the input image, generating candidate regions that are subsequently normalized.

(2) CNN Feature Extraction Stage

Following candidate region selection, normalized candidate boxes undergo CNN feature extraction. Normalization is essential because CNN's fully connected layers require fixed-size input images. R-CNN employs the simplest normalization method: cropping and warping.

(3) Classification and Bounding Box Regression Stage

Features extracted via CNN in the previous stage are fed as feature vectors from the final fully connected layer into the classifier and regressor for classification and bounding box prediction. A pre-trained linear SVM classifier handles feature classification for R-CNN, with a trained SVM model dedicated to each category.

To enhance candidate box localization accuracy, the R-CNN network employs a candidate bounding box regression strategy. The primary operation involves applying parameter transformations to candidate boxes obtained during fine-tuning training.

Candidate bounding box regression takes as input N training pairs $\{(P^i, G^i)\}_{1,\ldots,N}$, where $P^i = (P^i_x, P^i_y, P^i_w, P^i_h)$ denotes the i th candidate box, and $G^i = (G^i_x, G^i_y, G^i_w, G^i_h)$ represents the position of the i th ground truth object. P^i_x, P^i_y denote the x and y coordinates of the i th candidate box, while P^i_w, P^i_h denote its width and height. Similarly, $G^i_x, G^i_y, G^i_w, G^i_h$ denote the x-coordinate, y-coordinate, width, and height of the i th ground truth bounding box, respectively. A transformation method is required to bring the input candidate boxes closer to the ground truth boxes. However, if the overlap between a candidate box and a ground truth box is minimal or non-existent, transforming the candidate box parameters becomes meaningless. Therefore, an overlap ratio, known as the Intersection over Union (IoU), is defined.

The IoU formula is defined as:

$$IoU = \frac{A \cap B}{A \cup B} \tag{4}$$

It can be seen that set C represents the intersection of sets A and B, so $A \cup B$ can be transformed into $A + B - C$. At this point, loU is expressed as:

$$IoU = \frac{A \cap B}{A + B - C} \tag{5}$$

When set to 0, it indicates that the two regions do not overlap. When set to 1, it indicates that the two regions completely overlap. Values between 0 and 1 represent the degree of overlap between the two regions.

In R-CNN, setting this value to 0.6 and then performing parameter transformations on the candidate boxes yields a predicted value $\hat{G}^i$ closer to the true bounding box G^i. The transformation process is as follows:

$$\hat{G}_x^i = P_w^i d_x(P^i) + P_x^i \tag{6}$$

$$\hat{G}_y^i = P_h^i d_y(P^i) + P_y^i \tag{7}$$

$$\hat{G}_w^i = P_w^i \exp(d_w(P^i)) \tag{8}$$

$$\hat{G}_h^i = P_h^i \exp(d_h(P^i)) \tag{9}$$

where $d_w(P^i)$ and $d_x(P^i)$ transform the width and height, respectively, and $d_x(P^i)$ and $d_y(P^i)$ transform the center of the candidate box. Denoting this collectively as $d_*(P^i)$, it essentially represents a linear function of the convolutional feature map after CNN feature extraction. Assuming the candidate box undergoes pooling through the final layer of the CNN, we have:

$$d_*(P^i) = W_*^T \varphi(P^i) \tag{10}$$

W_*^T is the weight matrix, which is adjusted via the BP algorithm. The loss function is defined as:

$$L_{W_*} = \sum_{i=1}^{N} (t_*^i - W_*^T \varphi(P^i))^2 + \lambda \| W_* \|^2 \tag{11}$$

In the equation, t_*^i denotes the parameter transformation between P^i and G^i, specifically represented as follows:

$$t_x^i = \frac{(G_x^i - P_x^i)}{P_w^i} \tag{12}$$

$$t_y^i = \frac{(G_y^i - P_y^i)}{P_h^i} \tag{13}$$

$$t_w^i = \log(\frac{G_w^i}{P_w^i}) \tag{14}$$

$$t_h^i = \log(\frac{G_h^i}{P_h^i}) \tag{15}$$

Candidate box regression can effectively improve the localization accuracy of candidate boxes and enhance the accuracy of object detection through the aforementioned parameter transformation methods.

III. B. 2) Spatial Pyramid Pooling Layer

In R-CNN network models, due to the limitations of CNN's fully connected layers, input images must be uniformly sized. Consequently, all candidate region images require normalization through cropping and warping before being fed into the CNN for feature extraction. This process inevitably leads to incomplete preservation of image information, resulting in reduced object detection accuracy. In 2015, researchers introduced Spatial Pyramid Pooling (SPP) to address both the size constraint and storage issues, enabling multi-scale data input [43]. Since the convolutional layers in CNNs employ sliding window operations on input images, they impose no size restrictions. However, fully connected layers do impose constraints. Therefore, SPP is positioned between the convolutional and fully connected layers. The convolutional spatial pyramid pooling layer converts CNN outputs into fixed-size feature vectors, enabling CNNs to process images of arbitrary input dimensions. The spatial pyramid pooling layer structure is illustrated in Figure 4.

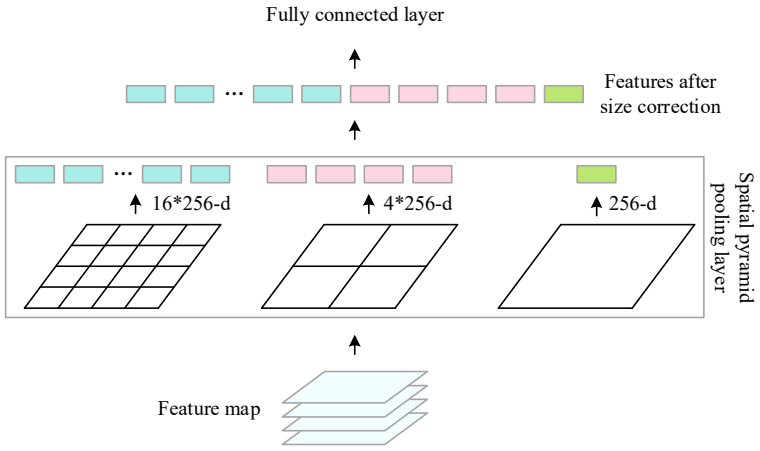

Figure 4: Spatial pyramid pooling structure

III. B. 3) Fast R-CNN Network

(1) Region of Interest Pooling Layer

Spatial pyramid pooling employs multiple pooling layers of varying scales to process input feature maps, whereas ROI Pooling uses only a single-scale pooling layer for this task. Consequently, it represents a special case within the spatial pyramid pooling methodology. ROI Pooling differs from Spatial Pyramid Pooling in that it uses a single-scale pooling layer to downsample the feature map matrix. It divides the convolutional features of each region of interest into 4×4 sub-blocks, performing max pooling on each sub-block to yield a 16-dimensional feature vector. Additionally, the ROI Pooling layer incorporates a bounding box mapping function, enabling backpropagation through the network model and thus resolving the training challenges associated with SPP layers. ROI Pooling resolves the inability of spatial pyramid pooling to update weights while retaining the advantage of spatial pyramid pooling layers—no restrictions on input image dimensions. Within the Fast R-CNN architecture, the ROI Pooling layer serves two purposes: mapping regions of interest (ROIs) from the input image to corresponding positions in the convolutional features, and feeding these convolutional feature regions into the pooling layer to produce fixed-size features. These fixed features are ultimately sent to the fully connected layer.

(2) Multi-Task Training

The R-CNN network model employs a multi-stage pipeline for parameter training, as detailed in the R-CNN network model section. This involves fine-tuning CNN parameters first, followed by training a linear SVM classifier, and finally applying the trained weight connection matrix during bounding box regression. This approach is cumbersome, consumes substantial storage resources, and hinders model transferability. Therefore, Ross Girshick adopted a multi-task simultaneous training approach using a multi-task loss function. Specifically, within the CNN, a classification output layer replaces the linear SVM classifier. This layer employs a Softmax classifier to output N+1 categories (where N represents object categories and 1 denotes background). The introduction of inter-class competition through Softmax yields superior classification performance compared to linear SVM. Additionally, Smooth Loss replaces bounding box regression. The multi-task loss function in FastR-CNN is defined as follows:

$$L(p,u,t^u,v) = L_{cls}(p,u) + \lambda[u \geq 1]L_{loc}(t^u,v) \tag{16}$$

In the above equation, u represents the true class label, t^u is the correction vector, p denotes the probability of the output class, and v is the parameter vector for candidate box transformation.

$L_{cls}(p,u)$ is the classification loss function, defined as follows:

$$L_{cls}(p,u) = -\log p_u \tag{17}$$

$L_{loc}(t^u,v)$ denotes the candidate box transformation loss function, defined as follows:

$$L_{loc}(t^u,v) = \sum_{i=(x,y,w,h)} smoothL1(t_i^u - v_i) \tag{18}$$

The smoothL1 function is defined as:

$$smoothL1(x) = \begin{cases} 0.5x^2, |x| < 1 \\ |x| - 0.5, \text{other} \end{cases} \tag{19}$$

III. B. 4) Faster R-CNN Network

PN networks are a type of network architecture based on fully convolutional networks (FCNs). The emergence of RPNs has truly integrated the entire object detection process into a single deep learning network structure. The combination of RPNs and Fast R-CNN forms the entirely new Faster R-CNN network architecture.

After adopting RPN, Faster R-CNN unified candidate box extraction, transformation, and classification. The schematic diagram of the Faster R-CNN network architecture is shown in Figure 5 [44], [45].

The RPN network loss function is defined as:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{box}}\sum_i p_i^* L_{box}(t_i, t_i^*) \tag{20}$$

In the formula, i is the index of the anchor box within each mini-batch, p_i corresponds to the predicted probability for each anchor point, and p_i^* represents the ground truth (GT) label (0 or 1) from the training set. t_i denotes the vector of predicted bounding box coordinates, while t_i' is the corresponding vector for the ground truth label. $L_{cls}(p_i, p_i')$ is the loss function for both target and non-target classes, defined as:

$$L_{cls}(p_i, p_i^*) = -\log[p_i p_i^* + (1-p_i^*)(1-p_i)] \tag{21}$$

L_{box} is the regression loss, defined as:

$$L_{box}(t_i, t_i^*) = R(t_i - t_i^*) \tag{22}$$

Here, R denotes the smooth L1 function, and $p_i^* L_{box}$ indicates that regression loss exists only when the foreground is present (=1), and otherwise no regression loss exists (=0).

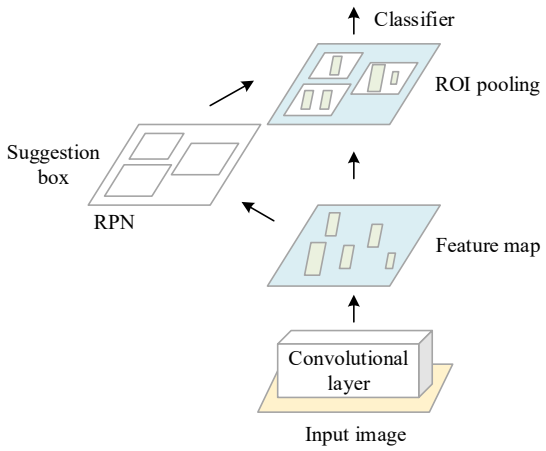

Figure 5: Schematic Diagram of the Faster R-CNN Network Structure

III. C. Experimental Setup and Results Analysis

III. C. 1) Experimental Setup

The experimental design comprises three components: dataset processing, experimental environment specifications, and experimental evaluation metric descriptions.

(1) Dataset Processing

The data used in this paper was collected from the WTDS dataset v of a CRH5 EMU train over 10 days from May 10 to May 19, 2022. The data was sampled at 1-minute intervals. Since only temperature status data was considered, the first bogie of car 02 was selected as the research subject, with a total of 18 axle temperature measurement points, i.e., a=18.

(2) Evaluation Metrics

To quantitatively assess the performance of different models on test datasets v2 and v3, this chapter employs three commonly used prediction evaluation metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). These are specifically defined by Equations (23) to (25):

$$MAE = \frac{1}{n \times a}\sum_{j=1}^{n}\sum_{i=1}^{a}\left|T_{i,j}^* - T_{i,j}\right| \tag{23}$$

$$MAPE = \frac{1}{n \times a} \sum_{j=1}^{n} \sum_{i=1}^{a} \left| \frac{T_{i,j}^{*} - T_{i,j}}{T_{i,j}} \right| \tag{24}$$

$$RMSE = \sqrt{\frac{1}{n \times a} \sum_{j=1}^{n} \sum_{i=1}^{a} (T_{i,j}^{*} - T_{i,j})^{2}} \tag{25}$$

(3) Experimental Environment

Experiments were conducted using the Faster R-CNN model and its comparative methods for short-term prediction of subway train bogie status. Both the Faster R-CNN model and its comparative methods were implemented in Python, with all deep learning architectures utilizing the PyTorch library. Training of the Faster R-CNN model was primarily performed on GPUs.

III. C. 2) Analysis of Experimental Results

This section compares the Faster R-CNN model with five classical models to demonstrate its effectiveness, primarily evaluating prediction accuracy and sensitivity. Table 1 compares the prediction accuracy of the six models at different strides. The Faster R-CNN model achieved the best MAE, MAPE, and RMSE metrics on both Test Set v2 and Test Set v3. For instance, on Test Set v2, the predicted values for the three metrics at step lengths of 5, 10, 15, and 20 were (1.1125, 3.0421, 1.9866), (3.7451, 9.9345, 6.4285), (5.4896, 15.5985, 9.7852), and (6.4685, 19.1958, 11.5463), respectively, outperforming other models.

Table 1: The analysis of the prediction accuracy of the six models

Model	Index	Test set v2			
		5th	10th	15th	20th
Faster R-CNN	MAE	1.1125	3.7451	5.4896	6.4685
	MAPE (%)	3.0421	9.9345	15.5985	19.1958
	RMSE	1.9866	6.4285	9.7852	11.5463
GRU	MAE	1.3048	4.2169	6.0496	7.0496
	MAPE (%)	3.3485	11.0485	17.0259	21.1648
	RMSE	2.6485	7.5968	10.8236	12.3496
LSTM	MAE	1.2485	4.3485	6.4985	7.3498
	MAPE (%)	3.1936	12.1654	17.9463	22.2468
	RMSE	2.5139	8.4698	11.5169	12.6954
kNN	MAE	3.5326	5.2425	6.8463	7.7836
	MAPE (%)	9.1248	15.4985	19.6116	22.8463
	RMSE	5.6985	9.4325	11.6485	13.2648
SVR	MAE	4.2648	6.1348	8.1385	10.0485
	MAPE (%)	12.2698	19.5094	21.1648	28.7985
	RMSE	6.3486	11.7498	14.0698	17.4985
ARIMA	MAE	4.7486	7.6426	11.4963	13.5985
	MAPE (%)	14.0698	23.9419	35.5498	46.0648
	RMSE	7.6248	14.0485	20.4686	24.2498
Model	Index	Test set v3			
		5th	10th	15th	20th
Faster R-CNN	MAE	2.4985	4.1636	5.5498	6.2485
	MAPE (%)	5.4996	10.5345	13.3498	17.9496
	RMSE	4.3115	7.7966	9.4858	10.7485
GRU	MAE	6.7463	9.1866	10.1926	15.4963
	MAPE (%)	10.5436	17.0048	26.1478	29.4185
	RMSE	7.4985	10.1648	15.6425	18.8563
LSTM	MAE	7.9763	13.7485	11.4936	19.1698
	MAPE (%)	9.4985	19.8463	22.7435	27.0189
	RMSE	9.3452	15.2498	13.7452	20.9468
kNN	MAE	4.9342	14.6368	18.7496	26.5365
	MAPE (%)	23.4685	21.3648	33.0974	35.0425
	RMSE	7.8462	16.4985	19.7465	30.0348
SVR	MAE	10.5749	13.1048	17.5361	22.7498
	MAPE (%)	25.8466	28.9482	38.3495	40.4348
	RMSE	12.9455	15.8399	22.4985	26.8168
ARIMA	MAE	11.4698	17.1648	18.6485	25.4185
	MAPE (%)	31.1636	33.6436	44.5498	42.0985
	RMSE	18.1648	21.6488	31.3489	35.1348

(1) Accuracy Comparison Across Different Methods

Due to space constraints, this paper presents prediction error metrics only for steps 5, 10, 15, and 20 on Test Set v2 and Test Set v3. Comprehensive analysis reveals that prediction errors across different models gradually accumulate as the number of prediction steps increases. For models like ARIMA and SVR, this error accumulation ultimately leads to significant deviations in final predictions. Compared to other models, Faster R-CNN exhibits the slowest and smallest rate of error accumulation. The MAE, MAPE, and RMSE values for all six models at steps 5, 10, 15, and 20 are plotted, with results for Test Set v2 shown in Figure 6 and Test Set v3 in Figure 7.

Comparing Figures 6 and 7 side-by-side reveals that under identical stride conditions, the Faster R-CNN model outperforms the other five models in prediction accuracy, exhibiting the lowest MAE, RMSE, and MAPE metrics. Compared to the GRU model, Faster R-CNN achieved improvements of 0.19–0.58 in MAE, 0.31–1.97 in MAPE, and 0.69–1.2 in RMSE.

Specifically, the Faster R-CNN model extracts spatial features of subway train bogies, enabling a more accurate description of bogie state trends and thereby effectively improving prediction accuracy. Among these, neural network models (GRU, LSTM) outperform traditional machine learning models (kNN, SVR) and time series models (ARIMA).

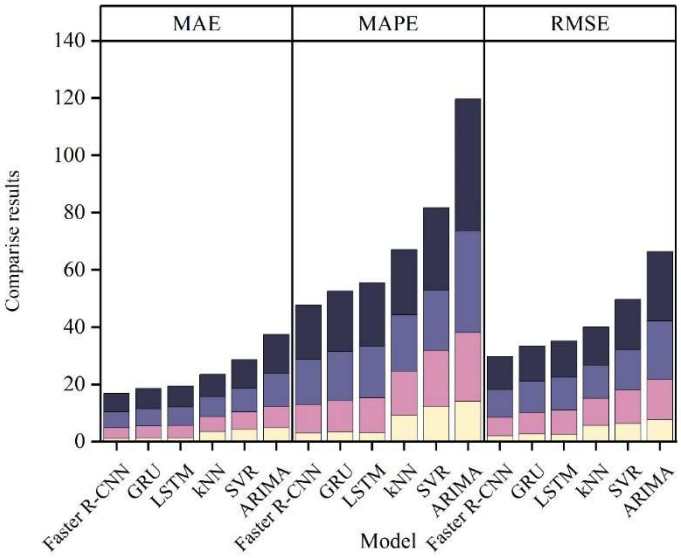

Figure 6: The test set v_2 results

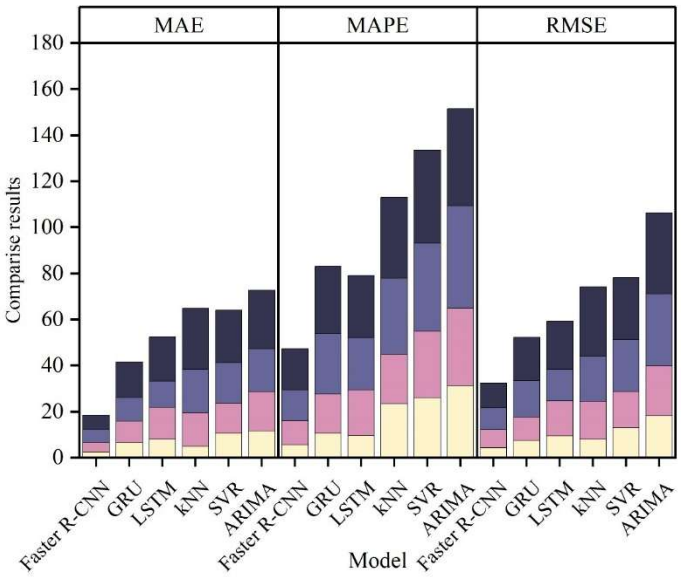

Figure 7: The test set v_3 results

(2) Tracking Sensitivity

Since the primary status data for subway train bogies consists of temperature readings, and temperature data is inherently

prone to abrupt changes as a streaming data type, any malfunction in the bogie will cause abnormal fluctuations in temperature readings. This demonstrates that changes in axle temperature can be used to infer potential bogie failures.

To evaluate the tracking sensitivity of the Faster R-CNN model, this chapter selected continuous data segments representing abnormal events (e.g., emergency braking) from dataset v to form dataset vr for experimentation. Predictions were made using the Faster R-CNN model, GRU model, and LSTM model on dataset vr, with the prediction accuracy of each model statistically analyzed as shown in Figure 8.

At t=700 in Figure 8, the train initiates emergency braking, causing a sudden temperature change. During this emergency braking event, the Faster R-CNN model demonstrated the fastest tracking speed. Throughout the duration of the sudden change from t=700 to t=720, the Faster R-CNN model maintained tracking prediction accuracy above 90%. This indicates that Faster R-CNN can rapidly react to sudden train events, learn new spatio-temporal features, and deliver high-accuracy predictions within a short timeframe. In contrast, GRU and LSTM models exhibit weaker initial responsiveness after emergency braking but can regain their previous prediction accuracy after a learning period. The accuracy achieved by the GRU model after new learning even surpassed that of the GCG model during certain periods. However, since the GRU model takes longer than the Faster R-CNN model to recover its accuracy after a drop following sudden changes, the Faster R-CNN model better meets the practical demands of high precision and rapid response in real-world applications.

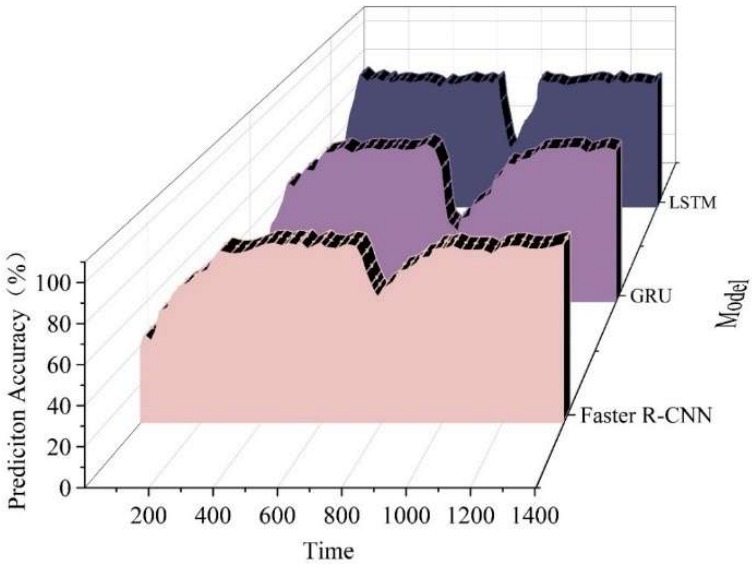

Figure 8: Sensitivity comparison of step 10th

IV. Application Analysis of Defect Detection in Subway Bogies

IV. A. In-Vehicle Application Validation

The bogie defect detection algorithm developed in this paper is deployed across 55 CR400BF platform and intelligent EMU trainsets. Assigned to multiple railway bureaus, these trains primarily operate on the Beijing-Shanghai, Beijing-Zhangjiakou, Beijing-Guangzhou, and Beijing-Harbin lines, achieving a maximum operational speed of 350 km/h and accumulating a total mileage of 1,243,640 kilometers.

The three-in-one safety monitoring system integrates vibration, instability, and smoothness monitoring functions. Components including axle boxes, anti-roll torsion bars, brake calipers, cables, and cable ties are connected to the host after integration with pre-processors and instability/smoothness sensors. The four-in-one safety monitoring system combines vibration, temperature, instability, and smoothness monitoring functions, eliminating the need for pre-processors.

To validate the accuracy of the bogie defect detection algorithm, this experiment carefully selected representative components as test samples. Four images capturing different spatial positions of the same component affected by defects were extracted, and each image underwent five detection runs. The actual defect distance values within each image were calculated and compared against true values to derive relative errors. By comparing the error rate between the average values and true values, the algorithm's accuracy and reliability were determined—lower error rates indicating greater precision and reliability. Detailed experimental results are presented in the table below. Table 2 shows the actual defect distances of the detected objects.

This algorithm demonstrated high precision in detecting defect locations. The error percentage for all detected samples was controlled within 4%, with a minimum error of 1.86% and a maximum error of 3.33%. This indicates the algorithm's excellent stability during repeated testing, effectively suppressing the interference of random errors on detection results. The high proximity between the average detected values and the actual values demonstrates that this algorithm possesses reliable

accuracy and practicality when handling defect target detection tasks.

Table 2: Test the actual distance of the object defect

Detection location	True value	Detection value 1/cm	Detection value 2/cm	Detection value 3/cm	Detection value 4/cm	Detection value 5/cm	Average detection value/cm	Percentage of error
Shaft front cover	44.4988	42.8796	43.4969	44.3645	42.9425	43.4125	43.4192	2.4261%
Side roll	50.4969	49.8185	51.3985	52.4395	50.9342	52.5969	51.43752	1.8627%
Brake clamp	62.7485	58.4396	60.5185	63.4152	61.4955	59.4125	60.65626	3.3343%
Cable and band	67.4694	63.4184	66.4987	68.7495	62.4966	66.6485	65.56234	2.8266%

IV. B. Analysis of Bogie Defect Cases

IV. B. 1) Axle Box Front Cover

Spectral analysis of impact data from the 7-position and 8-position axle boxes during bearing early warning is shown in Figure 9. The dashed lines represent the first-order and second-order spectrum lines for outer ring faults, respectively. Based on bearing structure calculations, at a rotational speed of 2024 rpm, the characteristic frequencies for the first- and second-order outer ring faults are 246 Hz and 498 Hz, respectively. Spectral data indicates that the 7th position axle housing measurement point on shaft 4 exhibits bearing outer ring 1x and 2x impact characteristics, while the 8th position on shaft 4 shows no impact features at the same time. This confirms that the bearing alarm was caused by outer ring damage, providing critical information for precise and rapid maintenance during subsequent inspections.

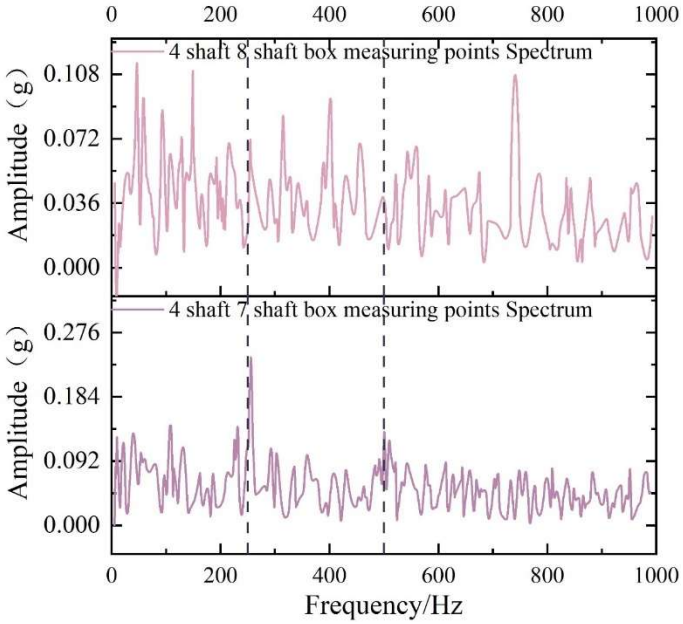

Figure 9: 4 axis 7 and 8 axle box shock data time frequency analysis

IV. B. 2) Anti-roll bar

Data samples from the anti-roll bar measurement points were extracted and analyzed, as shown in Figure 10. Analysis of the warning and pre-warning impact samples revealed: The anti-roll bar exhibits faint bearing outer ring impact characteristics, with distinct impact features at frequencies of 250Hz and 500Hz. The analysis results align with the warning status, indicating potential bearing abnormalities. Continuous monitoring of its condition changes is recommended, with priority inspection during subsequent maintenance procedures.

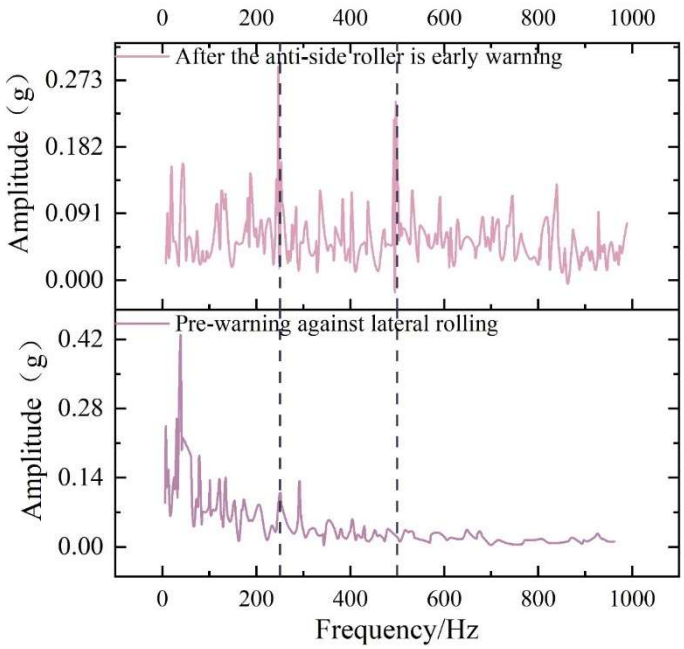

Figure 10: Analysis of frequency analysis of anti-lateral-roll torsion bar

V. Conclusion

This paper first designs an overall solution for metro bogie defect detection. Based on the established framework, it outlines the comprehensive operational workflow for metro bogie defect detection. An object detection algorithm is constructed using a convolutional neural network, with the Fast and Faster algorithms successively integrated to enhance detection performance.

To quantify the performance of different models on the test dataset, prediction evaluation metrics are employed. Experiments validate the detection accuracy and tracking sensitivity of the proposed algorithm. Compared to the GRU model, the proposed model achieves improvements of 0.19–0.58 in MAE, 0.31–1.97 in MAPE, and 0.69–1.2 in RMSE. The designed Faster R-CNN model exhibits the lowest cumulative error rate and severity.

Applying the model to practical subway bogie defect detection, this paper analyzes defect detection effectiveness using axle box front covers and anti-roll torsion bars as examples. The error percentage for all detection positions on subway bogies is controlled within 4%, with a minimum error of 1.86% and a maximum error of 3.33%. This demonstrates the algorithm's excellent stability and practicality in defect detection.

References

[1] Mao, P., Li, J., Xiong, L., Wang, R., Wang, X., Tan, Y., & Li, H. (2019). Characterization of urban subway microenvironment exposure—A case of Nanjing in China. International journal of environmental research and public health, 16(4), 625.

[2] Xue, X., Zhang, R., Zhang, X., Yang, R. J., & Li, H. (2015). Environmental and social challenges for urban subway construction: An empirical study in China. International Journal of Project Management, 33(3), 576-588.

[3] Sun, H., He, D., Zhong, J., Jin, Z., Wei, Z., Lao, Z., & Shan, S. (2023). Preventive maintenance optimization for key components of subway train bogie with consideration of failure risk. Engineering Failure Analysis, 154, 107634.

[4] Li, J., Wang, J., Li, X., Yang, J., & Wang, H. (2015). The experiment study for fatigue strength of bogie frame of Beijing subway vehicle under overload situation. The Open Mechanical Engineering Journal, 9(1), 260-265.

[5] Wang, B., Sun, S., Ma, S., & Wang, X. (2020). Fatigue damage and reliability assessment of subway train bogie frames under operating conditions. Advances in mechanical engineering, 12(1), 1687814020903590.

[6] Shen, L., & He, S. (2024). Modal analysis and frequency matching study of subway bogie frame under ambient excitation. Scientific Reports, 14(1), 21484.

[7] Fu, D., Wang, W., & Dong, L. (2015). Analysis on the fatigue cracks in the bogie frame. Engineering Failure Analysis, 58, 307-319.

[8] Yu, C., Chong, L., Zihua, L., Jie, S., & Xiebin, D. (2021, July). Research on Strength Test Technology and Fatigue Evaluation Method of Bogie Frame and Body Bolster. In Journal of Physics: Conference Series (Vol. 1972, No. 1, p. 012038). IOP Publishing.

[9] Liu, P., Yang, J., Liu, C., Wang, J., & Zhao, Y. (2025). SVMD-SqueezNet deep learning model for subway rail corrugation recognition. In Equipment Intelligent Operation and Maintenance (pp. 495-504). CRC Press.

[10] Wang, W., Liu, W., Lin, C., Li, M., Zheng, Y., & Liu, D. (2023). Fault detection system of subway sliding plug door based on adaptive EMD method. Measurement Science and Technology, 35(1), 015102.

[11] Karmiadji, D. W., Haryanto, B., Anwar, A., Prasetyo, B., Irawadi, Y., Farid, A. R., ... & Ahmad, H. (2022). Verification of urban light rail transit (LRT) bogie frame structure design lifetime under variable fatigue loads. Mechanical Engineering for Society and Industry, 2(1), 42-53.

[12] He, D., Ma, R., Jin, Z., Ren, R., He, S., Xiang, Z., ... & Xiang, W. (2023). Welding quality detection of metro train body based on ABC mask R-CNN. Measurement, 216, 112969.

[13] Liu, R., Wang, F. T., Wang, Z. P., Wu, C. L., & He, H. D. (2022). Identification of subway track irregularities based on detection data of portable detector. Transportation research record, 2676(12), 703-713.
[14] Li, G., Li, Y., & Bi, H. (2024). A Fault Monitoring System for Mechanical and Electrical Equipment of Subway Vehicles Based on Big Data Algorithms. Scalable Computing: Practice and Experience, 25(5), 3376-3387.
[15] De Donato, L., Flammini, F., Marrone, S., Mazzariello, C., Nardone, R., Sansone, C., & Vittorini, V. (2022). A survey on audio-video based defect detection through deep learning in railway maintenance. IEEE Access, 10, 65376-65400.
[16] Hao, Y., Xie, K., & Xu, Z. (2021, November). Research progress of fault detection equipment and methods for track train's shape part. In Journal of Physics: Conference Series (Vol. 2113, No. 1, p. 012052). IOP Publishing.
[17] Ing, D. S., & Li, J. (2022). Railway defect detection method: a review. Construction, 2(2), 22-30.
[18] Ren, L. N., & Lu, N. (2016). Research on static loaded test bed for metro vehicle bogie. In Materials, Manufacturing Technology, Electronics and Information Science (MMTEI2015) Proceedings for the 2015 International Workshop on Materials, Manufacturing Technology, Electronics and Information Science (MMTEI2015) (pp. 169-175).
[19] Tulbure, A. A., Tulbure, A. A., & Dulf, E. H. (2022). A review on modern defect detection models using DCNNs–Deep convolutional neural networks. Journal of Advanced Research, 35, 33-48.
[20] Zhang, D., Hao, X., Liang, L., Liu, W., & Qin, C. (2022). A novel deep convolutional neural network algorithm for surface defect detection. Journal of Computational Design and Engineering, 9(5), 1616-1632.
[21] Chen, J., Liu, Z., Wang, H., Núñez, A., & Han, Z. (2017). 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.
[22] Wu, Z. Y., Kalfarisi, R., Kouyoumdjian, F., & Taelman, C. (2020). Applying deep convolutional neural network with 3D reality mesh model for water tank crack detection and evaluation. Urban Water Journal, 17(8), 682-695.
[23] Li, D., Xie, Q., Gong, X., Yu, Z., Xu, J., Sun, Y., & Wang, J. (2021). Automatic defect detection of metro tunnel surfaces using a vision-based inspection system. Advanced Engineering Informatics, 47, 101206.
[24] Wang, A., Togo, R., Ogawa, T., & Haseyama, M. (2022). Defect detection of subway tunnels using advanced U-Net network. Sensors, 22(6), 2330.
[25] Zhang, H., Song, Y., Chen, Y., Zhong, H., Liu, L., Wang, Y., ... & Wu, Q. J. (2021). MRSDI-CNN: Multi-model rail surface defect inspection system based on convolutional neural networks. IEEE Transactions on Intelligent Transportation Systems, 23(8), 11162-11177.
[26] Wang, Q., Chen, S., Zeng, J., Du, W., & Wei, L. (2024). A deep learning fault diagnosis method for metro on-board detection on rail corrugation. Engineering Failure Analysis, 164, 108662.
[27] Wei, X., Yang, Z., Liu, Y., Wei, D., Jia, L., & Li, Y. (2019). Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. Engineering Applications of Artificial Intelligence, 80, 66-81.
[28] Zhang, Y., Xie, X., Li, H., Zhou, B., Wang, Q., & Shahrour, I. (2022). Subway tunnel damage detection based on in-service train dynamic response, variational mode decomposition, convolutional neural networks and long short-term memory. Automation in Construction, 139, 104293.
[29] Liu, Q., Guo, Q., Wang, W., Zhang, Y., & Kang, Q. (2021). An automatic detection algorithm of metro passenger boarding and alighting based on deep learning and optical flow. IEEE Transactions on Instrumentation and Measurement, 70, 1-13.
[30] Xie, Q., Tao, G., He, B., & Wen, Z. (2022). Rail corrugation detection using one-dimensional convolution neural network and data-driven method. Measurement, 200, 111624.
[31] Shi, W., Hai-tao, H., Ye-ming, Z., Yu-guang, W., Wei, Z., & Feng, J. (2020, September). Strength Detection Method for Subway Vehicle Bogie Frame in Big Data Environment. In International Conference on Advanced Hybrid Information Processing (pp. 442-451). Cham: Springer International Publishing.
[32] Hu, D., Jina, X., Yangb, C., & Wangc, H. (2024, February). Strength Analysis of Subway Bogie Frame Based on Experiment. In Proc. of SPIE Vol (Vol. 13064, pp. 130641J-1).
[33] Du, K., Jiao, S., Li, Y., Liu, W., Lei, Z., & Xin, J. (2022, December). Bogie Temperature Detection Method for Subway Trains Based on Dual Optical Images. In China Intelligent Robotics Annual Conference (pp. 257-266). Singapore: Springer Nature Singapore.
[34] Li, X., Zhang, W., Wang, Z., Hua, S., & Luo, W. (2025). Research on multiaxial fatigue life of metro bogie frame with welding defects. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 09544097251360038.
[35] Xia, J., Liao, T., & Zhang, L. (2017). Comprehensive Detection Technology for Urban Subway. In MATEC Web of Conferences (Vol. 95, p. 11004). EDP Sciences.
[36] Barbosa, B., Serrador, A., & Casaleiro, J. (2023, May). Condition monitoring maintenance in train bogie:: A low-cost acceleration sensor proposal. In 2023 International Conference on Control, Automation and Diagnosis (ICCAD) (pp. 1-4). IEEE.
[37] Yang, Y., Hu, Y., Chen, L., Liu, X., Qin, N., & Liu, Z. (2020, July). Defect detection of axle box cover device fixing bolts in metro based on convolutional neural network. In 2020 39th Chinese Control Conference (CCC) (pp. 7504-7509). IEEE.
[38] Wang, Z., Liu, R., Wang, F., & Tang, Y. (2022). Development of metro track geometry fault diagnosis convolutional neural network model based on car-body vibration data. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 236(9), 1135-1144.
[39] Pappaterra, M. J., Pappaterra, M. L., & Flammini, F. (2024). A study on the application of convolutional neural networks for the maintenance of railway tracks. Discover Artificial Intelligence, 4(1), 30.
[40] Pu, S., Song, T., Zhang, Y., & Xie, D. (2017). Estimation of crowd density in surveillance scenes based on deep convolutional neural network. Procedia computer science, 111, 154-159.
[41] Kun Shen,Mengyao Wu & Rongbin Chen. (2025). Model Predictive Control Algorithm for Converter Based on a Convolutional Neural Network. Applied Sciences,15(15),8658-8658.
[42] Lulin Zhao & Linfei Yin. (2025). Fully-connected layers-embedded self-attention optimizer based on quantum-inspired and fuzzy logic for smart household energy management. Sustainable Computing: Informatics and Systems,47,101151-101151.
[43] Na Zhou,Yuan Yuan & Lei Chen. (2025). A two-stage knowledge graph completion based on LLMs' data augmentation and atrous spatial pyramid pooling. Applied Intelligence,55(7),663-663.
[44] Shivani Sood,Harjeet Singh,Surbhi Bhatia Khan & Ahlam Almusharraf. (2025). Leveraging the WFD2020 Dataset for Multi-Class Detection of Wheat Fungal Diseases with YOLOv8 and Faster R-CNN. Computers, Materials & Continua,84(2),2751-2787.
[45] Li Dongjie,Wang Mingrui,Zhang Yu & Zhai Changhe. (2023). Application of an improved VGG and RPN network in precision parts recognition. Journal of Intelligent & Fuzzy Systems,45(6),9403-9419.