

Combined CNN and RNN Neural Networks for GPR Detection of Railway Subgrade Diseases

Liu, Huan; Wang, Shilei; Jing, Guoqing; Yu, Ziyue; Yang, Jin; Zhang, Yong; Guo, Yunlong

DOI

[10.3390/s23125383](https://doi.org/10.3390/s23125383)

Publication date

2023

Document Version

Final published version

Published in

Sensors

Citation (APA)

Liu, H., Wang, S., Jing, G., Yu, Z., Yang, J., Zhang, Y., & Guo, Y. (2023). Combined CNN and RNN Neural Networks for GPR Detection of Railway Subgrade Diseases. *Sensors*, 23(12), Article 5383. <https://doi.org/10.3390/s23125383>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Article

Combined CNN and RNN Neural Networks for GPR Detection of Railway Subgrade Diseases

Huan Liu ^{1,2}, Shilei Wang ³, Guoqing Jing ⁴ , Ziyu Yu ⁵, Jin Yang ¹, Yong Zhang ^{2,*} and Yunlong Guo ^{6,*} 

- ¹ School of Geophysics and Information Technology, China University of Geosciences, Beijing 100083, China; 3010180021@cugb.edu.cn (H.L.)
- ² Railway Engineering Research Institute, China Academy of Railway Sciences Co., Ltd., Beijing 100081, China
- ³ Infrastructure Inspection Research Institute, China Academy of Railway Sciences Co., Ltd., Beijing 100081, China
- ⁴ School of Civil Engineering, Beijing Jiaotong University, Beijing 100044, China
- ⁵ Institute of Geophysics, China Earthquake Administration, Beijing 100081, China
- ⁶ Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2628 CN Delft, The Netherlands
- * Correspondence: kldr_ql@126.com (Y.Z.); yunlong.guo@tudelft.nl (Y.G.)

Abstract: Vehicle-mounted ground-penetrating radar (GPR) has been used to non-destructively inspect and evaluate railway subgrade conditions. However, existing GPR data processing and interpretation methods mostly rely on time-consuming manual interpretation, and limited studies have applied machine learning methods. GPR data are complex, high-dimensional, and redundant, in particular with non-negligible noises, for which traditional machine learning methods are not effective when applied to GPR data processing and interpretation. To solve this problem, deep learning is more suitable to process large amounts of training data, as well as to perform better data interpretation. In this study, we proposed a novel deep learning method to process GPR data, the CRNN network, which combines convolutional neural networks (CNN) and recurrent neural networks (RNN). The CNN processes raw GPR waveform data from signal channels, and the RNN processes features from multiple channels. The results show that the CRNN network achieves a higher precision at 83.4%, with a recall of 77.3%. Compared to the traditional machine learning method, the CRNN is 5.2 times faster and has a smaller size of 2.6 MB (traditional machine learning method: 104.0 MB). Our research output has demonstrated that the developed deep learning method improves the efficiency and accuracy of railway subgrade condition evaluation.

Keywords: ground-penetrating radar; GPR; CNN; RNN; subgrade anomalies



Citation: Liu, H.; Wang, S.; Jing, G.; Yu, Z.; Yang, J.; Zhang, Y.; Guo, Y. Combined CNN and RNN Neural Networks for GPR Detection of Railway Subgrade Diseases. *Sensors* **2023**, *23*, 5383. <https://doi.org/10.3390/s23125383>

Academic Editor: Bill Moran

Received: 8 March 2023

Revised: 23 May 2023

Accepted: 5 June 2023

Published: 6 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Railway subgrade is a critical component of the railway system, essentially providing stable support for tracking (ballasted track, slab track), thus ensuring safe train operation [1]. However, with increased intensive high speed train loading, axle load, and traffic volume, the subgrade has undergone unacceptable rapid degradation, leading to frequent maintenance activities [2]. Subgrade defects (also as diseases and anomalies) are hazardous because they lead to rapid track degradation (such as track geometry irregularities, ballast differential settlement, sleeper damages, etc.), affecting ride comfort and safe train operations. In some situations, subgrade issues may even pose a risk to the train derailment [3].

Ground-penetrating radar (GPR) is a non-destructive technique that has been extensively used in various geophysical applications, including the inspection of transportation infrastructure, the evaluation of buried utilities, and the study of archaeology [4–9]. GPR can provide comprehensive information on the condition of the subgrade without excavation, making it a valuable tool for infrastructure managers to monitor and maintain the safety of train operations.

GPR was first applied in railway infrastructure diagnostic studies in the 1990s, using ground-coupled antennas, and, later, high-frequency air-coupled horn antennas were used with the advantage of non-contact testing at faster data acquisition speeds [10,11]. Until now, the hardware of GPR applied for railway infrastructure is mature. However, the software for processing and interpreting railway infrastructure GPR data still need improvements. One main reason is that the size of GPR data is growing exponentially, as the inspection speed, data acquisition rate, and railway line mileage have all dramatically increased. In particular, the process of applying GPR to the railway subgrade is still at the early stage of data processing. Further developments on the automated GPR data process and interpretation must be developed for rapid railway subgrade defect inspection (defect identification), finally achieving the goal of real-time railway infrastructure health monitoring.

The current GPR-based subgrade defect identification methods rely on manually designed features; afterwards, machine learning methods were used for classification. Examples are given as follows.

- In [12], the amplitude spectral characteristics at frequency inflection points were used to classify three types of railway ballast with a support vector machine (SVM) approach, achieving a classification accuracy of 99.5%.
- In [13], two-dimensional signal features were extracted, including energy and variance, as well as histogram statistical features such as mean, standard variance, smoothness, third-order moments, consistency, and entropy for radar images and structure image samples of typical subgrade defects. They then constructed a classification recognition model based on SVMs, achieving a recognition accuracy of over 85%.
- In [14], Hou divided radargrams of railway subgrade defects into blocks, extracted their demixing points, energy, and variance, and established optimal sparse radar features based on the L1 minimum norm method. Fuzzy C-means (FCM) and generalized regression neural network (GRNN) algorithms were used to identify railway subgrade defects, and the results showed that the classification accuracies of the sinkhole, mud-caking, and settlement were 100%, 100%, and 59.1%, respectively.

Deep learning is an important improvement in machine learning for GPR data. With the rapid development of computer technology, deep learning algorithms, which rely on powerful computing power, have been vigorously developed for NDT infrastructure health monitoring [15–18]. CNN and RNN have the ability to learn the data structure information, and the dependencies contained between data elements have obvious superiority in target recognition and classification [19,20].

- Besaw et al. [21] used CNN to classify potentially hazardous explosives below ground from GPR data, eliminating the feature selection step and demonstrating that high-precision subsurface anomaly identification can be achieved by combining neural network and ground-penetrating radar techniques.
- In [22], a unique Cascade R-CNN target detection framework was developed by employing 1030 annotated overturning GPR images to identify mud-pumping defects, achieving an average accuracy of 43.7%.
- Similarly, in [23], an LS-YOLOv3 network structure model was developed using 403 images of subgrade defects, including 261 GPR images of mud pumping and 279 GPR images of subsidence, achieving real-time fault detection with an average accuracy of 82.67%.
- Kang et al. [6] used several B-scan maps and horizontal slice maps to form a grid image as a dataset for the GPR image feature recognition of urban cavities and pipelines using a pre-trained AlexNet model for migration learning.

CNNs have a powerful feature extraction capability, but it is not applicable for time-series GPR data. Because of this, CNNs cannot learn time-series correlation to extract the dependencies embedded in the data, while RNN networks can compensate for this shortcoming [24].

- McLaughlin et al. [19] proposed an RNN in order to train feature extraction networks for human re-identification, combining CNN and RNNs structures. For a video sequence consisting of a full body image of a human, each image is passed through the CNN to generate a vector, which is a vector representation of the activation mapping of the CNN output layer. The vector is then passed to the recurrent layer as an input, where it is projected into a low-dimensional feature space and combined with information from previous moments.
- In order to improve the recognition accuracy, Xu et al. [25] proposed the convolutional gated recurrent neural network (CGRNN). The model first uses CNN as a feature extractor, and then the extracted robust features are fed into a bidirectional gated recurrent unit (BGRU). Since the GRU can only utilize historical information, the model uses the BGRU to learn long-term audio patterns in order to utilize future information.

The combination of CNN and RNN utilizes richer information in extracting target features, which is conducive to a better description of the target, thus improving the target detection and recognition performance. However, the combination of CNN and RNN has not been studied much in ground-penetrating radar data, especially for railway subgrade. So, it is very promising, beneficial, and particularly novel to combine CNN and RNN to process the GPR data of railway subgrade for defect recognition.

Based on the above-mentioned, this paper focuses on deep neural network (DNN) detection methods for subgrade defects using raw GPR data. We introduce a novel DNN model that employs a multi-layered one-dimensional CNN to automatically learn feature functions from signal channel waveforms, resulting in a model with fewer parameters and a faster run time compared to previous DNN models that use two-dimensional CNNs. Additionally, we incorporate a multi-layer RNN to process multiple channel features from the CNN, making the model more consistent with B-scan data. We first present the design of our CNN-RNN (CRNN) model and then demonstrate its performance using manually labeled B-scan data. Our results show that the CRNN achieves comparable accuracy to Faster RCNN at a higher speed and with a smaller model size.

2. Methodology

2.1. GPR Principles

The principles of GPR are based on the propagation and reflection of high-frequency electromagnetic waves, typically in the range of 10–3000 MHz, through the subsurface. When a GPR antenna transmits a pulse of electromagnetic wave energy into the ground, some of the wave energy is reflected on the surface by subsurface interfaces, such as changes in soil or rock types, voids, or buried objects. The reflected wave energy is then detected by the same antenna or a different one, and the signals are recorded and processed to create an image of the subsurface [26].

Specifically, the waveform of GPR signals varies depending on the dielectric properties and geometry of the subsurface medium. By analyzing properties such as two-way travel time and amplitude changes in the received signal, geophysicists can deduce the depth, shape, and other characteristics of the subsurface target body, as shown in Figure 1.

The amount of electromagnetic wave energy reflected by the internal structural layer of the railway substructure (ballast layer, sub-ballast layer, and subgrade) depends on the differences in dielectric constants between the media of each layer [27,28]. As the GPR antenna moves along the track, it records the motion properties of the reflected signal, including dual-range travel duration, amplitude, and phase [29].

The GPR system works by synchronously transmitting and receiving electromagnetic waves. As the antenna moves along the track, it creates a series of scan lines (A-Scans). The A-Scans from each measurement point are gathered together based on the acquisition interval, resulting in a GPR image of the railway substructure, especially the subgrade. This image visually represents changes in the physical properties of the railway structural layer interface and the subsurface medium, which can be used to identify and extract information necessary to understand the possible defects in the railway substructure [30].

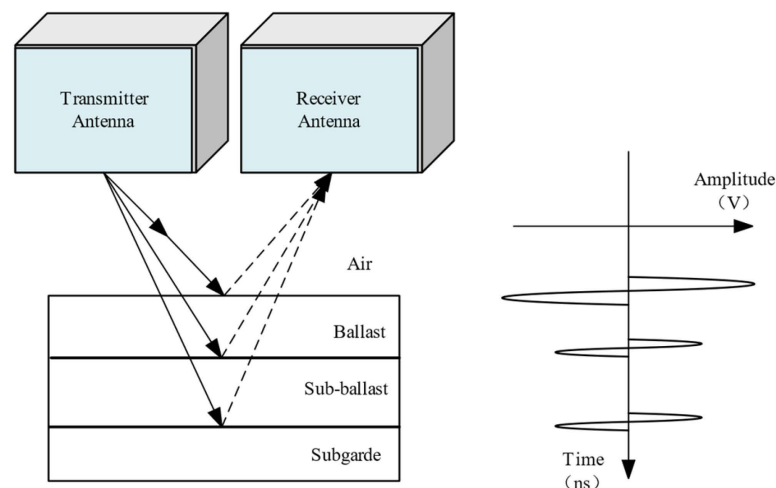


Figure 1. Schematic diagram of GPR detection for railway subgrade.

2.2. GPR Data Acquisition and Processing

The RIS GPR, developed by the Italian company Ingegneria dei Sistemi (IDS), is utilized to acquire GPR data, as demonstrated in Figure 2. The radar comprises a three-channel antenna group (tri-band antenna system), central control system, host radar, signal display instrument, range finder, and transmission cable. The antenna group is installed under the inspection train (developed by the China Academy of Railway Science), 30 cm above the ballast surface. The acquisition time window is set to 60 ns, with 512 sampling points and a trace interval of 11.25 cm.



Figure 2. The subgrade status inspection vehicle with a three-channel GPR antenna group.

Upon system startup, the Doppler rangefinder commands the radar system to transmit pulse signals at predetermined intervals. The collector then records the radar wave signals reflected from the structural layer of the roadbed. The received signal is displayed on the screen in real time, revealing the location of each railway substructure layer.

In GPR measurements, a wideband signal is often used to capture numerous interfering signals simultaneously, resulting in more effective reflected wave characteristics. However, to extract subgrade defect information using AI models, it is essential to eliminate these interfering signals to enhance the data's signal-to-noise ratio (SNR) [31]. Interference signals can come from three aspects [32]: First, the interference signal from the device system itself includes antennas, cables, and connectors inside. This interference is inevitable. The second is an interference from other signal sources, such as radio and television transmissions, communication signals, etc. The last one is from ground or underground/subterranean interference. This type of interference is caused by transmitting

and receiving antennas that directly couple waves, or form ground-reflected waves. Due to the inhomogeneous underground medium and the occurrence of strong scattering, bypass waves are superimposed on the formation of interference waves.

Therefore, it is very necessary to pre-process the GPR data. In this study, the data were pre-processed, using the GR processing software developed by the China University of Mining and Technology (Beijing), in six steps:

- **Sampling.** Before filtering the original data, discrete sampling is required, and the sampling process needs to satisfy the sampling theorem; otherwise, the data spectrum will be mixed and thus generate false frequencies. The sampling theorem in the frequency domain equation is $\omega_s = 2\omega_N \geq 2\omega_{max}$, where ω_s is the sampling frequency, ω_N is Nyquist frequency, and ω_{max} is the highest frequency of the signal [33].
- **Zero-line correction.** The zero-line setting is mainly carried out by cutting the air layer to a fixed threshold value, which is set at the most stable point on the electromagnetic wave trajectory A-scan. Depending on the type of antenna and the center frequency, setting the appropriate threshold position along the A-scan reflects the accuracy of the results, and this threshold can be summarized as (1) the initial arrival of the wave; (2) the position of the first trough; (3) the position of the first wave crest location of the zero amplitude value between the first wave crest and the trough; (4) the location of half the amplitude value between the first wave crest and the trough; (5) the position of the first wave crest [34].
- **Gain setting.** Using automatic gain control (AGC). When the signal is strong, its gain automatically decreases and when the signal is weak its gain automatically increases. It can ensure the uniformity of strong and weak signals and facilitate the tracking of effective waves [35].
- **FIR bandpass filtering.** The bandpass filter works by cutting off the fringe bands from the spectrum of GPR data. The filter consists of two filters, a high-pass filter and a low-pass filter, which modify the ground-penetrating radar signal by removing the low- and high-frequency components of the spectrum. As a rule of thumb, a bandwidth of 1.5 times the survey center frequency can be used initially [32,35].
- **Running average filters.** The background noise of the GPR signal is high, and the autoregressive sliding average spectral estimation (ARMA) in modern spectral estimation is used to analyze and identify the non-smooth signal of the effective reflected signal, which can extract the signal features at a low signal-to-noise ratio with a high accuracy of spectral estimation [36].

After pre-processing, the inevitable and random interference signals in the GPR image are suppressed, increasing the SNR. The variations of dual trip time (or depth) and mileage are reflected in the GPR image features. These image features are the reflected wave amplitude, waveform, phase, and spectrum of the inter-layer interface of the subgrade, as well as the subgrade defect. A collection of typical GPR images for various defect types is shown in Table 1.

Table 1. Different types of railway subgrade defects.

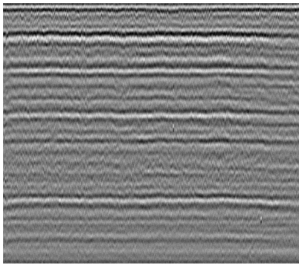
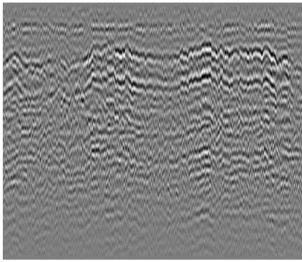
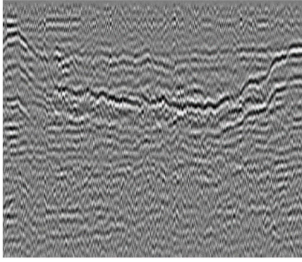
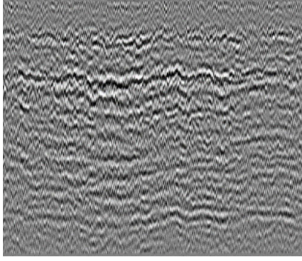
Railway Subgrade Defect Types	Typical Defects Image	Image Features
Normal subgrade without defect		The lamellar structure is obvious. The in-phase axis is straight and continuous. The reflected energy is uniform.

Table 1. Cont.

Railway Subgrade Defect Types	Typical Defects Image	Image Features
Mud pumping		The wave group is a disorganized, discontinuous, low-frequency strong reflection shape that resembles a mountain tip or straw hat.
Settlement		The reflection of the in-phase axis of the settlement radar image is significantly bent, with depth-downward offset.
Water anomaly		The signal is attenuated. The reflected energy of the top surface is strong. Multiple waves exit.

2.3. Training and Testing Dataset

Our approach is designed to process original pre-processed GPR data. The original signal files record a series of binary data called A-scan. We used MATLAB to convert the processed data into decimal data based on the arrangement of GPR raw data, which was then saved in MAT format. The MAT format dataset is a two-dimensional matrix, where n is the number of channels (or the number of A-scan) and m is the perpendicular length of waveform data. The echo data F can be represented by an $m \times n$ matrix as follows:

$$\mathbf{F} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

where x_{ij} represents the response recorded at the j th channel and the i th sampling point.

We selected 3000 channels of GPR pre-processed data and inverted them into an image, as shown in Figure 3 (left figure). The number of subgrade defects among these 3000 GPR data was counted, as shown in the right graph in Figure 3.

Additionally, to test the efficiency of the CRNN network model, we compared it with object detection network models such as Faster R-CNN and Yolov3 models. The two-dimensional matrix dataset was converted into GPR images using an IDS signal treatment software named SRS DPA basic 02.02.004, as shown in Figure 4.

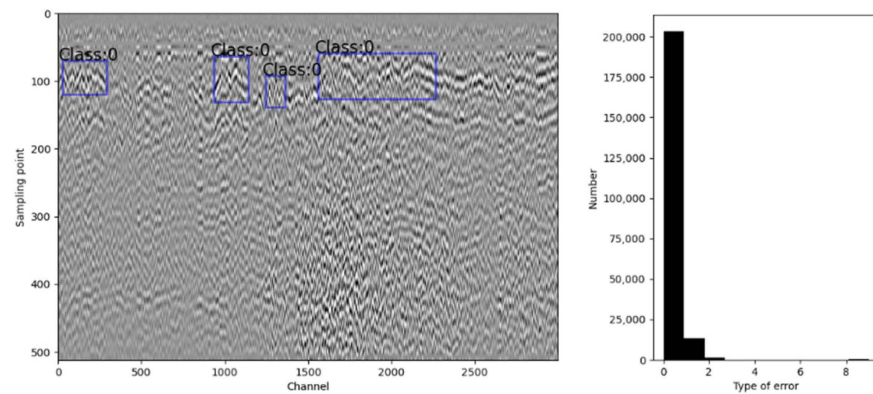


Figure 3. Original GPR data and its statics.

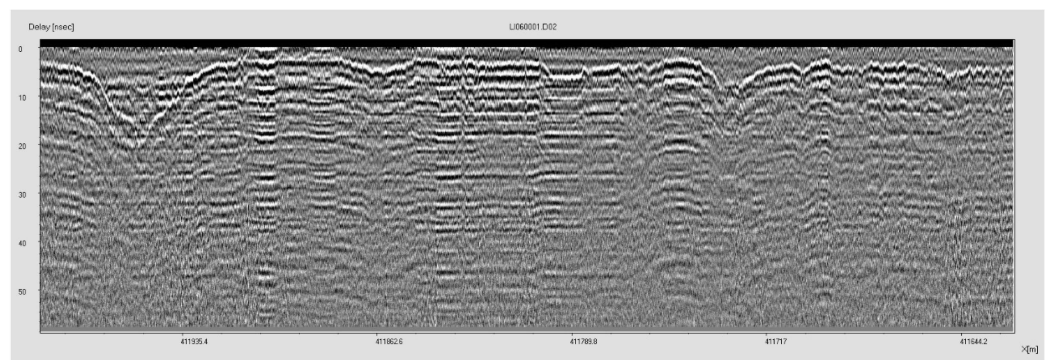


Figure 4. GPR image data exported by SPS DPA software.

The object detection formula is $x^{\text{img}} \in \mathbb{R}^{H \times W \times 3}$. H is the height of images for which the sampling point N exists in GPR data. W indicates the width of an image by the number of channels of GPR data. To compare CRNN and object detection methods, the inputs of an image are x^{img} and x^{GPR} .

$$x_{i,j}^{\text{GPR}} = \frac{1}{3} \sum_k x_{i,j,k}^{\text{img}}$$

We note that x^{GPR} is different to x^{orig} . x^{GPR} is the grayscale map of an image x^{img} , which includes a boundary output by SRS DPA software. The bit-rates of x^{GPR} are 8 and 24 of x^{orig} .

2.4. CRNN Network Structure

The CRNN is a fusion network that combines CNN and RNN to process different GPR features. The network architecture is shown in Figure 5. CNNs were first proposed by Le Cun et al. as a highly nonlinear mapping method in supervised learning mode for establishing the connection between target samples and inputs [37]. RNNs are neural networks with recurrent connections that can model sequence data for sequence recognition and prediction [38]. They can also store information using recurrent iterative functions, capture contextual information well, and achieve transient dependency learning. The CRNN network takes advantage of both CNN and RNN to process GPR data, making it an effective tool for detecting defects in the subgrade.

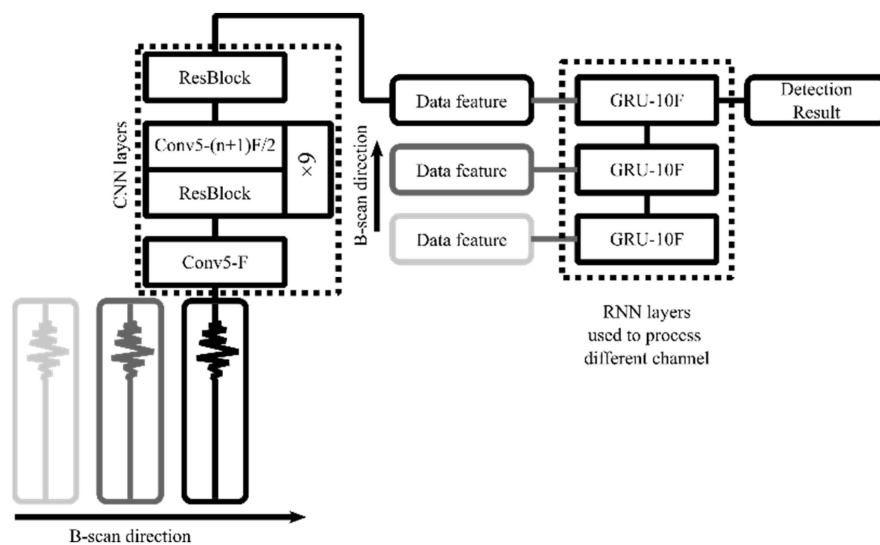


Figure 5. Network structure diagram of CRNN.

The CRNN network employs one-dimensional convolution to process the original waveform directly. To enhance the convergence speed of the network, we incorporate a batch normalization layer into the convolutional network. The calculation of a single-layer network can be expressed as the following formula:

$$\begin{aligned}
 x^{k+1} &= x^k * w^k + b^k \\
 \hat{x}^{k+1} &= \gamma \frac{x^{k+1} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta^k \\
 y^{k+1} &= \text{RELU}(\hat{x}^k)
 \end{aligned}$$

where $*$ represents the convolution operation, and w^k, b^k are the trainable parameters in the convolution layer. μ and σ represent the mean and standard deviation of the data, while γ and β are the trainable parameters in the normalization layer. The addition of normalization in the convolution layer can effectively accelerate the model’s convergence. Additionally, the trainable parameters added to the normalization layer can be used to calculate the data’s magnitude and mean value.

For downsampling, we add a stride of 2 to the convolutional layer instead of using a pooling layer in the model. The original single-channel waveform contains 512 sampling points. After 9 convolutional layers with a stride of 2, we end up with a feature vector of length $9F$, where F is the number of filters in the convolutional layers. This feature vector is extracted from the waveform and is used for further processing.

Waveform features are typically manually created using techniques such as wavelet transform and frequency domain transform, while CNN features are automatically extracted from training data [20]. This suggests that it is possible to obtain characteristics automatically, without physical effort.

RNNs can be viewed as a collection of interconnected networks, where the output of one network is fed into the next in a chained design [39]. As a result, both the input and output of the network are determined before the network has an impact on the output of the subsequent network. In contrast, the input and output of CNNs are independent of each other. RNNs have a “memory” that retains all the computed information. In our model, RNN layers are used to process features from a single waveform, where the feature from the k th station is $h_k \in \mathbb{R}^{10F}$. There are N channels in total, and so the input to the RNN is $h \in \mathbb{R}^{N \times 10F}$. Our model’s RNN is a unidirectional RNN, meaning it can process both the current and prior channel features and produce accurate detection results, represented as o_k . Unidirectional RNN predicts the outcome as

$$p(o_k | x_k, x_{k-1}, \dots, x_1)$$

This implies that the detection result depends on the previous waveform. Processing the B-Scan in time is enabled by the unidirectional RNN structure, which is simple, easy to train, and achieves high accuracy.

The final step in our CRNN model involves adding two fully connected output layers on top of the RNN output. The first output layer generates the classification result $c_k \in \mathbb{R}^2$, which determines the type of the k th waveform. We have two types of waveforms: normal and anomalous. The softmax function is applied to constrain the output c_k :

$$c_k^i = \frac{\exp(y_{k,i})}{\sum_m \exp(y_{k,m})}$$

Here, $y_{k,i}$ is the original output of the classification layer, which is derived from the k th waveform of the i th type.

The second output layer generates the regression result $r_k \in \mathbb{R}^2$, which indicates the upper and lower positions of an anomalous waveform. The sigmoid function [40] is applied to constrain the output r_k :

$$r_k^i = 512 \cdot \text{sigmoid}(y_{k,i})$$

We use the sigmoid function to limit the boundaries of the regression output so that the upper and lower positions do not exceed the length of the waveforms. This enhances the robustness of the regression output [41].

2.5. Anomaly Detection

The outputs of CRNN are regression and classification. To obtain a box position of anomaly, some post-processing is required. The two types of output are shown in Figure 6. The vertical red line represents the left border are represented by b_{x1} and b_{x2} , respectively, while b_{y1} and b_{y2} indicate the upper and lower boundaries. The predictions of the CRNN, represented by r_k and c_k , are used to determine the upper and lower boundaries and the type of anomaly in the k th channel. The start and end of an anomaly type in c_k can be used to constrain the left and right boundaries of the anomaly, i.e., b_{x1} is the start of an anomaly type and b_{x2} is the end of it. The upper and lower boundaries are determined by the mean of the anomaly type, $x_i = \mathbb{E}(r)$, $i = 1, 2$.

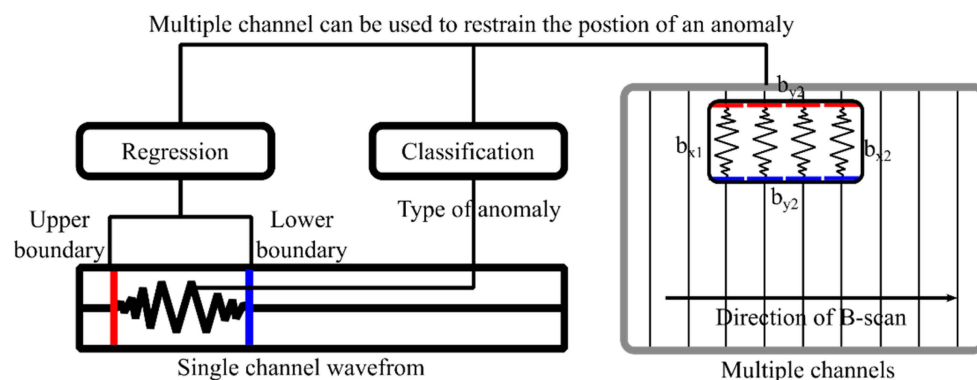


Figure 6. Illustration of two outputs of CRNN. The vertical red line represents the left boundary. The vertical blue line represents the right boundary. The horizontal red line represents the upper boundary. The horizontal blue line represents the lower boundary.

2.6. Training of CRNN

The CRNN loss function includes both classification and regression errors, and it is expressed by the following formula:

$$L = -\frac{1}{N} \sum_k \log c_{k,d} + \frac{1}{N} \sum_k w_k \sum_i (r_{k,i} + \hat{f}_{k,i})^2$$

where d represents the type of the current channel, and $\hat{r}_{k,i}$ is the manually labeled boundary. The weight of the regression loss is represented by w_k , which is set to 0.0 when there is no anomaly, and it is set to 1.0 when the k th channel is anomalous.

To optimize the CRNN structure, the Adam algorithm [42] was used instead of stochastic gradient descent. The initial learning rate was set to 1×10^{-4} , and after 1000 iterations, we reduced it to 1×10^{-5} to ensure a more stable convergence. Figure 7 shows the visualization of the CRNN loss function.

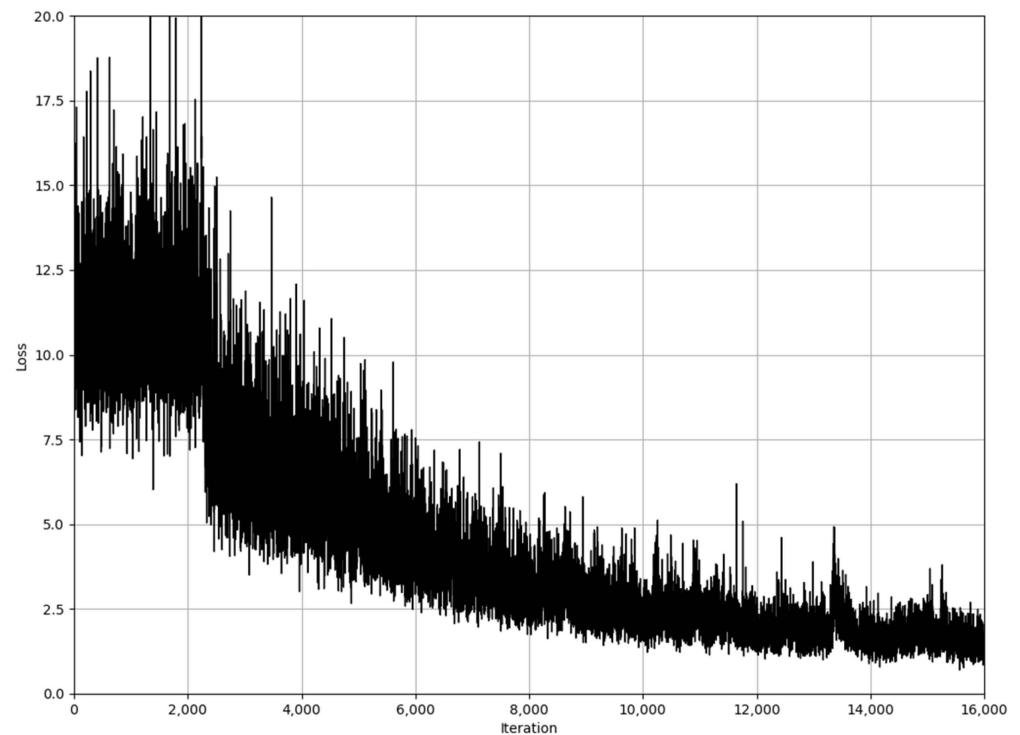


Figure 7. Loss function of CRNN structure.

Our network structure here is training on Titan RTX with a 24G memory and a Threadripper 3975WX CPU. It needs 12 h to train our model. The deep learning framework is PyTorch 1.9 and CUDA 11.3. The loss function ranges from 12.5 to about 1.5 after 16,000 iterations.

For the initial 2500 iterations of the iterative process, we used a 1×10^{-4} learning rate to warm up the network, which can prevent the instability problem during the iterative process, and the learning rate in the subsequent iterations was chosen as 1×10^{-3} , and the convergence speed became faster. Since the data within each batch are randomly selected, there are oscillations in the iterations, but the overall trend remains constant after 14,000 iterations, with a small increase in the loss function around 13,500 iterations, due to re-training after program termination.

3. Case Study

3.1. Detection Result on Real Railway Subgrade GPR Data

Our methods can process the original GPR waveform. We selected one typical Chinese existing railway line as the test dataset, which consists of eight data packets. Every GPR data packet includes 266,666 channels. We selected 1000 channels of the fifth data packet to test the detection result, as shown in Figure 8.

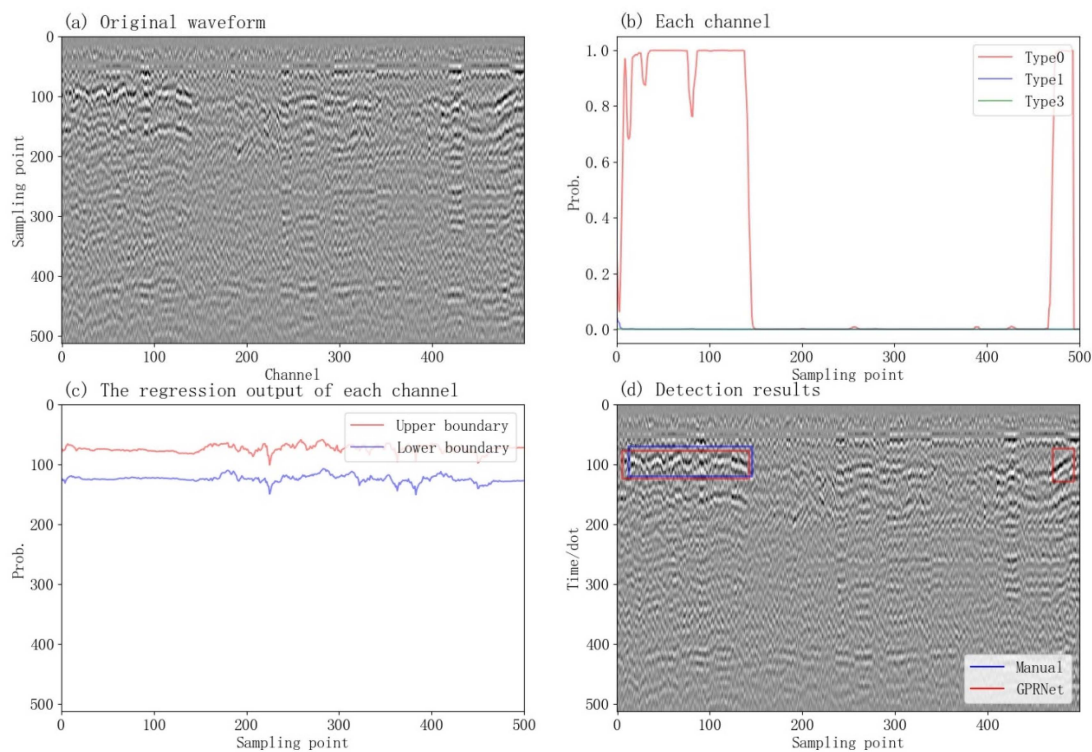


Figure 8. The detection result of 1000 channels from the fifth data packet. (a) Original waveform data. (b) Each channel. (c) The regression output of each channel. (d) Detection results.

The results from Figure 8 demonstrate that the left and right boundaries of an anomaly are constrained by the probability of the classification output. Anomalies start with a probability greater than 0.5 and end with a probability smaller than 0.5 (as shown in Figure 8b). The upper and lower boundaries are determined by the regression output, and the mean of the regression is used to establish the boundary (as shown in Figure 8c). The anomaly box can be determined by combining the regression and classification results (as shown in Figure 8d). Overall, the detection results are in close agreement with the manually labeled data.

During the detection process, each channel classifies anomalies, and the lateral accuracy depends on the channel spacing. In Figure 8b, we observe that the probability of anomalies is not constant, and some anomalies have a lower confidence level due to being offset in the homophase axis in the GPR. However, CRNN can synthesize data from other channels, which enables it to identify the data belonging to an anomaly by combining information from other channels. This is one of the advantages of RNN modeling compared with single-channel data analysis.

Figure 8c shows that even the normal tract has predicted upper and lower boundaries due to the lack of constraints on the training data. Therefore, some randomness may occur in the prediction of the normal tract. However, since the regression output does not constrain the horizontal position of the abnormality, it needs to be combined with the classification output in Figure 8b to determine the lateral boundary. The combination of these two approaches results in the accurate detection of the abnormal position, as shown in Figure 8d.

Our detection model is unidirectional and can be applied to GPR data of any length in the B-scan direction. To test the accuracy of our model, we utilized the full GPR section, which consists of 26,666 channels. We evaluated our model using four indexes:

1. Precision (P). This is the rate of successfully detected channels. It is calculated as
$$P = \frac{TP}{TP+FP}.$$

2. Recall (R). This is the rate of the detected anomalous channels out of all the anomalous channels in the dataset. It is calculated as $R = \frac{TP}{TP+FN}$.
3. Mean error of the four boundaries. This indicates the bias of our model.
4. The standard deviation of the four boundaries. This measures the variability in our model's performance.
5. The detection model is a single-way model.

TP samples are channels that are both manually labeled and detected by our model. FP samples are detected by our model but not manually labeled, while FN samples are manually labeled but not detected by our model. The results of the four evaluation indicators are presented in Table 2.

Table 2. Statistical results with different strides.

Model	P	R				
S = 1	0.79	0.74	−14.73	−40.47	30.25	48.59
S = 2	0.79	0.7	−17.67	−44.56	31.05	50.98
S = 3	0.79	0.71	−18.13	−42.86	30.4	49.75
S = 4	0.81	0.67	−20.74	−47.71	32.06	52.6
S = 5	0.76	0.73	−16.38	−39.54	28.87	48.58
S = 6	0.76	0.65	−22.41	−50.83	31.3	52.39

Table 2 presents the results of our evaluation using different down-sampling rates ranging from one to six. To downsample, we took one channel out of each S channel. The x-direction and y-direction are in different units: the unit of the x-direction is the number of channels, while the unit of the y-direction is the number of sampling points. The error of the x-direction (left and right boundary) is approximately 20 channels, and the error of the y-direction (upper and lower boundary) is approximately 18 sampling points.

We also calculated the P–R curve for different strides and plotted it in Figure 9. The results show that our CRNN model achieves high accuracy across all strides.

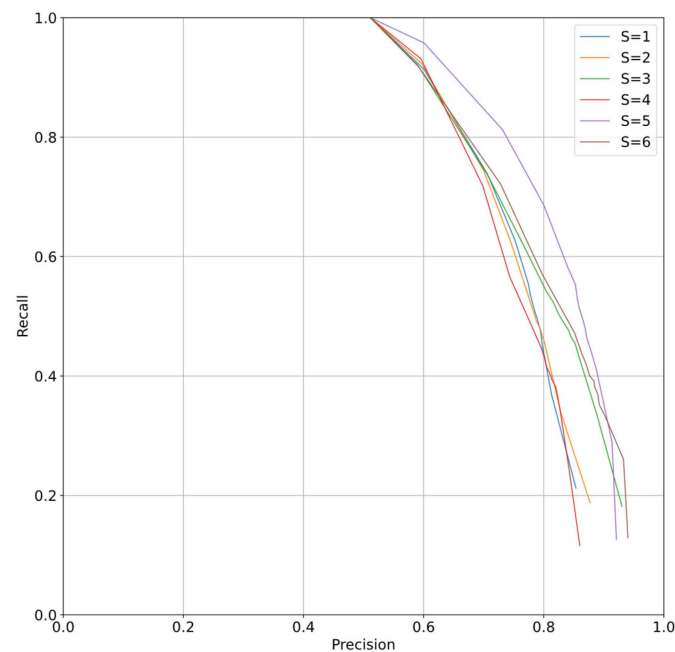


Figure 9. P-R curve of different strides of CRNN.

By using stride, we downsampled the GPR data in the scanning direction while maintaining consistent accuracy. This allowed our model to handle GPR data of different scales and channel spacing while ensuring accuracy.

3.2. The Inferring Speed in GPR Data

Inferring speed is a critical consideration when deploying a deep neural network. To ensure our CRNN model can be run on a wide range of devices, we tested it on both CPU and GPU devices, as shown in Table 3. We also transformed the CRNN model to an ONNX format, which is a lightweight DNN framework that can run faster than other models.

Table 3. Inferring speed (test on 266,666 channels).

Running Device	Inferring Time (ms)
CPU	1325
GPU	2139

Since the RNN model is not easily parallelizable, its speed is similar to CPU or GPU. The inference speed of our model is approximately 2.139 s. Being a single-way model, our model can analyze GPR data instantly. The unidirectional model ensures that the current result depends only on the information from the previous channel, making it computationally more efficient and suitable for real-time processing.

As the RNN model is time-dependent, its hypothetical effect is not apparent on the GPU, but it is computationally faster on the CPU.

3.3. Comparison with Object Detection Models

We compared our methods with Faster R-CNN, which takes an image as the input. To enable comparison, we transformed the color image, where the height and width of the image were transformed into the GPR data format. First, we transformed the color image into gray by taking the mean of its channels:

$$x_{i,j}^{\text{gray}} = \frac{1}{3} \sum_{k=1}^3 x_{k,i,j}$$

The resulting gray image had the format $x^{\text{gray}} \in \mathbb{R}^{(H \times W)}$. However, GPR data have a different unit, where W represents the number of channels and H indicates the number of sampling points in one channel. Since CRNN requires $H = 512$, we resized the image x^{gray} to have $W' = W \cdot 512 / H$ channels, resulting in GPR-like data $x^{\text{gpr}} \in \mathbb{R}^{(512 \times W')}$. The accuracy was tested on the zoomed data x^{gpr} . Since the image has padding, we retrained our model to fit the image data rather than the GPR data.

We manually labeled 2800 images, of which 2400 were used to train the models and 400 were used to test the accuracy. Both CRNN and Faster R-CNN used the same training and test datasets. The hyperparameters of CRNN were the same as for real GPR data. Faster R-CNN is an open-source project that can be downloaded from GitHub.

3.3.1. Detection Result on Image Data

The detection results from CRNN and Faster R-CNN are shown in Figure 10.

Figure 10 shows that the object detection and CRNN models produce different results. The CRNN model tends to detect small anomalies due to its reliance on the class predictions of the tracks. Therefore, if there is noise in consecutive anomalous channels that are detected as normal, the detection process may split a large anomaly into multiple small ones. To address this issue, we removed the anomalies containing less than X number of channels in the post-processing stage. In contrast, the Faster R-CNN model produces more complete results since it is based on individual anomalous image pixels rather than individual channels. The longitudinal resolution of the object detection model is arbitrary, while our model needs to be combined with RNN for processing. Hence, the number of longitudinal sampling points is limited to 512, and if it is insufficient or exceeded, manual cropping is required.

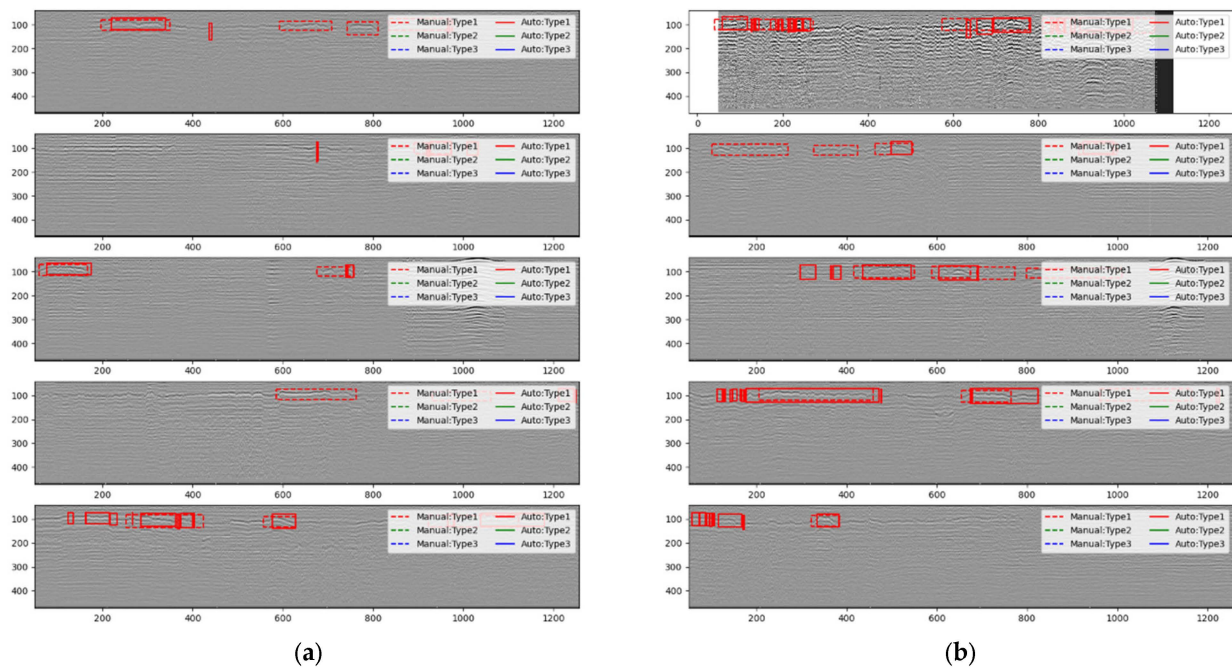


Figure 10. Detection results of the two models. (a) CRNN detection results, (b) Faster R-CNN detection results.

3.3.2. Statics on Image Data

As shown in Table 4, our CRNN model achieves higher accuracy than Faster R-CNN when IoU is greater than 0.25 or 0.5, indicating that our model is effective in detecting railway subgrade anomalies. However, when the IoU is greater than 0.75, the accuracy of Faster R-CNN is higher than our model. This may be due to the fact that Faster R-CNN is based on individual anomalous image pixels rather than individual channels, which allow it to detect more detailed information in the image. Nonetheless, our CRNN model still performs well in detecting subgrade anomalies, and can provide valuable information for railway maintenance and safety.

Table 4. Precision and recall comparison with different methods.

Model Name	Faster RCNN			CRNN (Ours)		
	Precision	Recall	F1Score	Precision	Recall	F1Score
IoU > 0.25	0.777	0.936	0.849	0.795	0.941	0.862
IoU > 0.50	0.752	0.671	0.709	0.834	0.773	0.803
IoU > 0.75	0.527	0.260	0.348	0.422	0.158	0.230

Table 5 demonstrates that the CRNN model has a much smaller size compared to Faster R-CNN, making it more memory efficient for inference. Additionally, our CRNN model exhibits faster inference speeds on both CPU and GPU, allowing for deployment on a wider range of devices while maintaining satisfactory performance in railway subgrade anomaly detection.

Table 5. The size and training time of different models.

Model Name	Faster RCNN	CRNN (Ours)
Detected 238 images on CPU (ms)	18,000	2600
Detected 238 images on GPU (ms)	2000	500
Model Size (MB)	104	2.6

Based on the experimental results and literature survey, this paper analyzes the advantages and disadvantages of four network models, respectively, CNN, RNN, CRNN, and Faster RCNN, as shown in Table 6.

Table 6. The tabulation of the advantages and disadvantages of different deep models.

Deep Model	Advantages	Disadvantages
CNN	One-dimensional CNN can extract features in-depth, i.e., longitudinally, in the time dimension, generating a vector for each GPR waveform which can reduce the storage requirements [19].	Since CNN has no memory function for time series, it cannot take the input data sequence into account like RNN [24].
RNN	RNN can be regarded as a series of interconnected networks that have a chained architecture, where the output of the next network depends on both its input and the output of its predecessor network [43].	Because of the chain derivative rule and the existence of a nonlinear activation function, RNN is prone to the problem of gradient vanishing and explosion [44].
CRNN	Lightweight, the model size is only 2.6 MB. Faster inference speeds on both CPU and GPU are easy to deploy on hardware devices. No complex data processing steps are required (e.g., filtering and information compression), taking into account the physical mechanisms of GPR data and their temporal dynamic behavior.	Due to the insufficient sample number and sample imbalance, the CRNN model only determines the existence of anomalous regions, and the identified anomalies still need to be interpreted manually. When the IoU is greater than 0.75, the accuracy of the CRNN model is low, and more neurons may need to be added to improve accuracy.
Faster R-CNN	Faster RCNN, as a two-stage network, contains RPN and RCNN, and has higher recognition accuracy compared to the one-stage network. Many deep learning frameworks have a practical [45] Faster RCNN source code that is easy to use.	Although the detection accuracy of Faster RCNN is improved, the training speed of its network model is slow [46].

4. Conclusions and Perspective

We developed a novel GPR anomaly detection model called the CRNN network, which is based on a hybrid CNN and RNN architecture. While most existing GPR detection models rely on object detection models, which are more widely used and easier to train, these image-based models do not consider the unique physical characteristics of GPR data, such as the different physical meanings of the sampling points in the B-Scan direction of the channels and waveforms [34]. This can lead to accuracy degradation since GPR data need to be converted to image data during processing.

In contrast, our CRNN model is specifically designed to account for both longitudinal and transverse physical quantities, making it more targeted than object detection models built directly using CNNs. Our one-dimensional CNN processes waveform sampling point data, while the RNN processes transverse channel data. This allows our model to directly process GPR data without the need for image conversion, resulting in a smaller and more computationally efficient model.

Our CRNN network has a size of only 2.6 MB, compared to 104 MB for the Faster R-CNN model, making it more computationally efficient. Additionally, our model is four times faster than object detection models, as it only deals with single-channel waveforms and does not need to process inter-channel data. Overall, our CRNN network provides a more accurate and efficient approach to GPR anomaly detection.

Our CRNN network demonstrates comparable accuracy to Faster R-CNN, with some differences in performance at different IoU thresholds. Specifically, our model performs better at $\text{IoU} \leq 50\%$ due to its ability to detect anomalies in channels during the detection process. However, it may detect some large anomalies as multiple small anomalies due to noise, resulting in more small anomalies at the anomaly boundary that require filtering based on expert experience. Additionally, the starting position of anomalies may not always be clear, leading to alternating normal and anomaly readings at the boundary.

During deployment in a real production environment, our lightweight model can save computational resources as it does not require converting GPR data into images, or post-processing for anomaly localization. With sufficient training data, the CNN model can even process GPR data directly without filtering, further shortening the processing time. Additionally, the use of a one-way RNN model allows for real-time data processing without interception.

Our detection model is limited to determining the presence of anomalies and does not classify specific types of anomalies. More data should be added in subsequent work to improve classification accuracy. Additionally, the testing accuracy of our model may be slightly lower than other articles due to the complexity of our actual working condition data, highlighting the need for further data accumulation to improve accuracy.

Author Contributions: Conceptualization, H.L.; methodology, H.L. and Z.Y.; validation, S.W., G.J., J.Y. and Y.Z.; formal analysis, Y.G.; writing—original draft preparation, H.L. and Z.Y.; writing—review and editing, S.W., G.J. and Y.G.; funding acquisition, S.W. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The study was funded by the Science and Technology Research and Development Program of China State Railway Group Co., Ltd. [Grant No. K2022G015] and the Fund Project of China Academy of Railway Sciences Corporation Limited [Grant No. 2022YJ305]. The article processing charge (APC) was funded by the Delft University of Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, S.; Liu, G.; Jing, G.; Feng, Q.; Liu, H.; Guo, Y. State-of-the-Art Review of Ground Penetrating Radar (GPR) Applications for Railway Ballast Inspection. *Sensors* **2022**, *22*, 2450. [[CrossRef](#)] [[PubMed](#)]
2. Fontul, S.; Paixão, A.; Solla, M.; Pajewski, L. Railway Track Condition Assessment at Network Level by Frequency Domain Analysis of GPR Data. *Remote Sens.* **2018**, *10*, 559. [[CrossRef](#)]
3. Xiao, J.; Liu, L. Permafrost Subgrade Condition Assessment Using Extrapolation by Deterministic Deconvolution on Multifrequency GPR Data Acquired Along the Qinghai-Tibet Railway. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 83–90. [[CrossRef](#)]
4. Bano, M.; Tsend-Ayush, N.; Schlupp, A.; Munkhuu, U. Ground-Penetrating Radar Imaging of Near-Surface Deformation along the Songino Active Fault in the Vicinity of Ulaanbaatar, Mongolia. *Appl. Sci.* **2021**, *11*, 8242. [[CrossRef](#)]
5. Solla, M.; Pérez-Gracia, V.; Fontul, S. A Review of GPR Application on Transport Infrastructures: Troubleshooting and Best Practices. *Remote Sens.* **2021**, *13*, 672. [[CrossRef](#)]
6. Kang, M.-S.; Kim, N.; Lee, J.J.; An, Y.-K. Deep learning-based automated underground cavity detection using three-dimensional ground penetrating radar. *Struct. Health Monit.* **2020**, *19*, 173–185. [[CrossRef](#)]
7. Manataki, M.; Vafidis, A.; Sarris, A. GPR Data Interpretation Approaches in Archaeological Prospection. *Appl. Sci.* **2021**, *11*, 7531. [[CrossRef](#)]
8. Francke, J. Applications of GPR in mineral resource evaluations. In Proceedings of the XIII International Conference on Ground Penetrating Radar, Lecce, Italy, 21–25 June 2010; pp. 1–5.
9. Valles, J.; Chapa, T.; Matesanz, J.; González, M.A.M. Combined application of Multi-Channel 3D GPR and Photogrammetry from UAVs for the study of Archaeological sites. In Proceedings of the 3th Technoheritage 2017 International Congress, Cádiz, Spain, 20–23 May 2017; pp. 20–24.
10. Lai, W.W.-L.; Dérobert, X.; Annan, P. A review of Ground Penetrating Radar application in civil engineering: A 30-year journey from Locating and Testing to Imaging and Diagnosis. *NDT E Int.* **2018**, *96*, 58–78.
11. Liu, S.; Lu, Q.; Li, H.; Wang, Y. Estimation of Moisture Content in Railway Subgrade by Ground Penetrating Radar. *Remote Sens.* **2020**, *12*, 2912. [[CrossRef](#)]
12. Shao, W.; Bouzerdoum, A.; Phung, S.L.; Su, L.; Indraratna, B.; Rujikiatkamjorn, C. Automatic classification of GPR signals. In Proceedings of the XIII International Conference on Ground Penetrating Radar, Lecce, Italy, 21–25 June 2010; pp. 1–6.
13. Du, C.; Zhang, Q.; Liu, J. Intelligent identification of railway roadbed defects by vector machines. In Proceedings of the 2017 Conference of China Civil Engineering Society, Guangzhou, China, 10–12 November 2017; pp. 355–365. (In Chinese).

14. Hou, Z.; Zhao, W.; Yang, Y. Identification of railway subgrade defects based on ground penetrating radar. *Sci. Rep.* **2023**, *13*, 6030. [[CrossRef](#)]
15. Alyoubi, K.H.; Sharma, A. A Deep CRNN-Based Sentiment Analysis System with Hybrid BERT Embedding. *Int. J. Pattern Recognit. Artif. Intell.* **2023**, *37*, 2352006. [[CrossRef](#)]
16. Lee, K.; Lee, S.; Kim, H.Y. Deep Learning-Based Defect Detection Framework for Ultra High Resolution Images of Tunnels. *Sustainability* **2023**, *15*, 1292. [[CrossRef](#)]
17. Zhang, L.; Zhang, K.; Pan, H. SUNet++: A Deep Network with Channel Attention for Small-Scale Object Segmentation on 3D Medical Images. *Tsinghua Sci. Technol.* **2023**, *28*, 628–638. [[CrossRef](#)]
18. Gao, J.; Yuan, D.; Tong, Z.; Yang, J.; Yu, D. Autonomous pavement distress detection using ground penetrating radar and region-based deep learning. *Measurement* **2020**, *164*, 108077. [[CrossRef](#)]
19. McLaughlin, N.; Del Rincon, J.M.; Miller, P. Recurrent Convolutional Network for Video-Based Person Re-identification. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1325–1334.
20. Tong, Z.; Gao, J.; Sha, A.; Hu, L.; Li, S. Convolutional Neural Network for Asphalt Pavement Surface Texture Analysis: Convolutional neural network. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 1056–1072. [[CrossRef](#)]
21. Besaw, L.E.; Stimac, P.J. Deep convolutional neural networks for classifying GPR B-scans. In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XX*; Bishop, S.S., Isaacs, J.C., Eds.; SPIE Defense + Security: Baltimore, MD, USA, 2015; p. 945413.
22. Xu, X.; Jiang, B.; Huang, Q. Intelligent identification method for railway roadbed slurry and mud infestation based on Cas-cade R-CNN. *Railw. Eng.* **2019**, *29*, 99–104. (In Chinese)
23. Ma, Z.; Yang, F.; Qiao, X. Intelligent detection method for railway roadbed defects. *Comput. Eng. Appl.* **2021**, *57*, 272–278. (In Chinese)
24. Tong, Z.; Gao, J.; Yuan, D. Advances of deep learning applications in ground-penetrating radar: A survey. *Constr. Build. Mater.* **2020**, *258*, 120371. [[CrossRef](#)]
25. Xu, Y.; Kong, Q.; Huang, Q.; Wang, W.; Plumbley, M.D. Convolutional gated recurrent neural network incorporating spatial features for audio tagging. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 3461–3466.
26. Xie, F.; Lai, W.W.; Dérobert, X. Building simplified uncertainty models of object depth measurement by ground penetrating radar. *Tunn. Undergr. Space Technol.* **2022**, *123*, 104402. [[CrossRef](#)]
27. Read, D.; Meddah, A.; Li, D.; Mui, W. Volpe Center. In *Ground Penetrating Radar Technology Evaluation on the High Tonnage Loop: Phase 1: DOT/FRA/ORD-17/18*; Federal Railroad Administration: Washington, DC, USA, 2017.
28. Brown, M.; Li, D. *Ground Penetrating Radar Technology Evaluation and Implementation: Phase 2: DOT/FRA/ORD-17/19*; Federal Railroad Administration: Washington, DC, USA, 2017.
29. Basye, C.; Wilk, S.; Gao, Y. *Ground Penetrating Radar (GPR) Technology Evaluation and Implementation: DOT/FRA/ORD-20/18*; Federal Railroad Administration: Washington, DC, USA, 2020.
30. Shapovalov, V.; Vasilchenko, A.; Yavna, V.; Kochur, A. GPR method for continuous monitoring of compaction during the construction of railways subgrade. *J. Appl. Geophys.* **2022**, *199*, 104608. [[CrossRef](#)]
31. Li, F.; Yang, F.; Yan, R.; Qiao, X.; Xing, H.; Li, Y. Study on Significance Enhancement Algorithm of Abnormal Features of Urban Road Ground Penetrating Radar Images. *Remote Sens.* **2022**, *14*, 1546. [[CrossRef](#)]
32. Bianchini Ciampoli, L.; Tosti, F.; Economou, N.; Benedetto, F. Signal Processing of GPR Data for Road Surveys. *Geosciences* **2019**, *9*, 96. [[CrossRef](#)]
33. Lombardi, F.; Griffiths, H.; Lualdi, M. The Influence of Spatial Sampling in GPR Surveys for the Detection of Landmines and IEDs. In Proceedings of the 2016 European Radar Conference (EuRAD), London, UK, 5–7 October 2016.
34. Benedetto, A.; Tosti, F.; Ciampoli, L.B.; D'amico, F. An overview of ground-penetrating radar signal processing techniques for road inspections. *Signal Process.* **2017**, *132*, 201–209. [[CrossRef](#)]
35. Jol, H.M. *Ground Penetrating Radar: Theory and Application*; Elsevier: Amsterdam, The Netherlands, 2009.
36. Cui, F.; Wu, Z.-Y.; Wang, L.; Wu, Y.-B. Application of the Ground Penetrating Radar ARMA power spectrum estimation method to detect moisture content and compactness values in sandy loam. *J. Appl. Geophys.* **2015**, *120*, 26–35. [[CrossRef](#)]
37. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
38. Zhang, A.; Wang, K.C.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.Q.; Chen, C. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces with a Recurrent Neural Network: Automated pixel-level pavement crack detection on 3D asphalt surfaces using CrackNet-R. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 213–229. [[CrossRef](#)]
39. Mauch, L.; Yang, B. A new approach for supervised power disaggregation by using a deep recurrent LSTM network. In Proceedings of the 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Orlando, FL, USA, 14–16 December 2015; pp. 63–67.
40. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia, Italy, 13–15 May 2010.
41. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.

42. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
43. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 103–111.
44. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training Recurrent Neural Networks. *arXiv* **2013**, arXiv:1211.5063.
45. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
46. Hou, F.; Rui, X.; Fan, X.; Zhang, H. Review of GPR Activities in Civil Infrastructures: Data Analysis and Applications. *Remote Sens.* **2022**, *14*, 5972. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.