Received 6 September 2025
Revised 30 December 2025
Accepted 27 January 2026

Track intrusion detection using point cloud filtering and voxel feature learning

Peng Lin
CRRC Qingdao Sifang Rolling Stock Research Institute Co Ltd, Qingdao, China

Kunlin Cai
School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China, and

Xiukun Wei
School of Traffic and transportation, Frontiers Science Center for Smart High-Speed Railway System and State Key Laboratory of Advanced Rail Autonomous Operation, Beijing Jiaotong University, Beijing, China

Abstract

Purpose – The purpose of this paper is to design a fully supervised point cloud voxel network method for foreign object intrusion detection, which integrates CSF filtering and can effectively address the issue where radar data interference from railway track reflections affects the detection of small objects.

Design/methodology/approach – A supervised learning-based method for foreign object intrusion detection is proposed, which integrates a voxel-based point cloud network with the Cloth Simulation Filtering (CSF) algorithm.

Findings – The CSF algorithm is applied to filter out ground points from the railway scene, effectively mitigating the interference caused by strong radar reflections from the track surface, which often hinder the detection of small objects. Considering railway-specific scene characteristics, an improved Voxel R-CNN is introduced: the method first performs voxelization and voxel-wise feature encoding, then uses a three-dimensional (3D) convolutional backbone for multiscale feature fusion. The resulting features are compressed via Bird's Eye View projection, and a voxel query mechanism is used to generate 3D bounding boxes for regions of interest.

Research limitations/implications – Despite the promising performance of the proposed method, several limitations should be acknowledged. First, the detection accuracy of small objects (e.g. stones) remains susceptible to point cloud sparsity and LiDAR resolution, especially under long-range sensing conditions. Second, the current model is trained and validated exclusively on data collected from a controlled railway test environment, and its generalization capability to diverse track structures, sensor configurations and environmental conditions requires further verification. Notably, this study does not include a direct experimental comparison with video-based detection methods under adverse weather conditions. Although prior studies have demonstrated the superior robustness of LiDAR sensing in low-illumination and visually challenging scenarios, a rigorous cross-modality comparison necessitates synchronized multisensor data collected under identical weather conditions – an issue that will be addressed in future work. In future work, the authors plan to incorporate multisensor fusion and domain adaptation strategies to enhance robustness

Funding: This work was supported by the National Key Research and Development Program of China (Grant No. 2022YFB4301204) entitled *Research on High-Speed Train Safety Assurance and Maintenance Technologies Based on Active Safety.* The authors gratefully acknowledge the financial support provided by the above funding program, which has made this research possible.

Smart and Resilient Transportation
Vol. 8 No. 2, 2026
pp. 148-167
Emerald Publishing Limited
e-ISSN: 2632-0495
p-ISSN: 2632-0487
DOI 10.1108/SRT-09-2025-0024

across diverse railway scenarios. In addition, lightweight model optimization will be explored to facilitate real-time deployment on embedded railway monitoring platforms.

Originality/value – Experimental results show the method achieves average detection accuracies of 80% for pedestrians and stones, and 40% for boxes, demonstrating its effectiveness for on-site deployment.

Keywords Track intrusion detection, Light detection and ranging, Supervised learning, 3D object detection, Deep learning

Paper type Research paper

1. Introduction

With the rapid development of China's high-speed rail network, its efficiency and convenience have enabled an increasing number of people to enjoy the pleasure of travel, while also placing higher demands on its safety. During operation, high-speed trains have occasionally experienced safety incidents caused by foreign objects entering the tracks, leading to derailments. During high-speed train operations, due to high speeds and limited driver visibility, relying solely on the driver's visual observation to detect foreign objects and manually apply brakes constitutes a passive safety defense mechanism – when the driver identifies an obstacle, the train often collides with the foreign object at high speed due to insufficient braking distance. Therefore, establishing an active safety protection mechanism based on beyond-line-of-sight technology is of great significance for ensuring the stable and safe operation of high-speed trains.

Computer vision can be broadly divided into two stages in terms of processing radar data, with 2017 serving as a key dividing point between the two stages. Before 2017, traditional neural networks were unable to process three-dimensional (3D) data, so there were few direct applications of neural networks to the processing of radar points cloud. Some researchers projected 3D point clouds and then processed them using two-dimensional (2D) convolutions. Li *et al.* (2016) proposed the VeloFCN algorithm, which converts point clouds into forward-view 2D feature maps (histograms), analogous to the depth maps in RGBD (RGB color and depth map) data, while designing a special bounding box encoding method to enable 2D convolutional neural networks to predict complete 3D bounding boxes. Yang *et al.* (2019) proposed PIXOR, which projects point clouds along the *z*-axis into a two-dimensional (2D) bird's-eye view (BEV) and constructs occupancy features and intensity features as input to a fully convolutional structure as the backbone network. In addition, a multitask detection head without candidate regions is used to directly perform pixel-wise predictions in the BEV. Furthermore, regardless of the projection angle, compressing 3D data into 2D data reduces one-dimensional information, which is clearly detrimental to feature extraction by the network.

In 2017, Charles *et al.* (2017) pioneered the PointNet model, which successfully addressed the issues of input disorder, 3D rotation invariance and local information correlation in point clouds. Subsequently, to address the issues of missing local feature information and nonuniformity in point clouds due to distance, the PointNet++ network model was proposed. In PointNet++, Qi *et al.* (2017) introduced density-adaptive layers, namely, Multiscale Grouping (MSG) and Multiresolution Grouping (MRG). MSG and MRG can adaptively capture the local structure and different granularity information of point clouds. The emergence of the PointNet series of networks has provided a new approach to directly processing point cloud data. Finally, a series of variants based on the PointNet series of networks emerged, such as Point R-CNN and 3DSSD. Similarly to PointNet, these models directly extract features from the raw point cloud through set abstraction layers (SA) and use

feature propagation layers (FP) for upsampling to propagate features (Shi *et al.*, 2019; Yang *et al.*, 2020).

In 2018, another approach to point cloud processing emerged: voxels. Similarly to how 2D images are composed of small square pixels, a 3D space is composed of small cubic voxels. Zhou and Tuzel (2018) proposed VoxelNet, an end-to-end deep learning network for 3D object detection. This network first explicitly voxelizes the 3D point cloud space and uses a Voxel Feature Encoding layer (VFE) for feature extraction, resulting in a sparse 4D tensor. Subsequently, VoxelNet uses 3D convolutions to construct convolutional layers, expanding the receptive field and incorporating more descriptive information. Finally, it proposes a 3D RPN layer modeled after the 2D RPN layer. To accelerate the computation speed of sparse tensors, the introduction of Second (Sparsely Embedded Convolutional Detection) accelerates the network's computational results and convergence speed (Yan *et al.*, 2018). Subsequently, researchers began to recognize that setting the voxelization granularity too large results in significant information loss, while setting it too small leads to a geometric increase in computation time. PointPillar extends the z-axis of the voxel grid infinitely, voxelizing the point cloud into pillars, thereby transforming the 3D space into a pseudo-image feature format. It then processes this using a 2D CNN network and uses an anchor-free detection network to obtain 3D bounding boxes (Lang *et al.*, 2019).

Railway intrusion detection and risk assessment have been extensively investigated in recent studies. For example, quantitative risk assessment methods based on text mining and fuzzy rule-based Bow-Tie models have been proposed to analyze intrusion risk levels. In addition, several works integrate YOLO-based visual detection with risk assessment mechanisms to provide early warning for railway obstacle intrusion. These approaches mainly focus on risk modeling and decision-making, while the present work emphasizes LiDAR-based point cloud detection for robust perception in railway environments.

It should be noted that the aforementioned research on point cloud detection mainly focuses on the development of general 3D object detection technologies. In the specific scenario of high-speed railways, however, railway track intrusion detection and risk assessment have become key research directions in recent years, with substantial achievements accumulated. For example, Li *et al.* proposed a quantitative risk assessment method based on text mining and fuzzy rule–based Bow-Tie models to analyze the risk levels of track intrusions (Huang *et al.*, 2022); in addition, Zhang *et al.* integrated YOLO-based visual detection technology with risk assessment mechanisms to construct an early warning system for railway obstacle intrusion (Zhang *et al.*, 2024). These research methods mainly focus on the levels of risk modeling and decision-making, while this study focuses on LiDAR-based point cloud detection technology, aiming to achieve robust perception in railway scenarios and make up for the deficiencies of existing research in the accurate perception of intrusion targets.

Visual deep learning has achieved notable success in the field of track foreign object intrusion detection, encompassing diverse detection technologies ranging from 2D images to 3D point clouds and multisensor feature fusion. However, current research has largely failed to adequately explore the complexity of data set construction and its impact on model generalization performance. Given this, this paper focuses on a supervised learning-based method for foreign object intrusion detection, aiming to simplify the data set construction process while addressing the model limitations in real-time performance, detection accuracy and adaptability to complex environments, thus achieving significant improvements. The primary contribution of this work lies in the engineering adaptation of voxel-based 3D detection frameworks to railway track intrusion scenarios, rather than proposing a fundamentally new detection architecture.

2. Improved voxel R-CNN

In recent years, with the development of 3D sparse convolution, the voxel method has become as fast as directly processing 3D points in the development of 3D object detection algorithms. Furthermore, in the detection of small objects, the voxel method has certain advantages over the farthest point sampling due to its uniform sampling. In 2021, Deng Jiajun *et al*. proposed the Voxel R-CNN 3D object detection method based on voxels, which demonstrated efficient, accurate and concise performance in 3D object detection tasks. This paper uses the Voxel R-CNN-based detection method for 3D object detection on point clouds of foreign object intrusion in railway scenarios and improves the Voxel R-CNN model to better suit railway foreign object intrusion scenarios. The following sections will provide a detailed explanation of the Voxel R-CNN method.

Ground points are removed using the Cloth Simulation Filtering (CSF) algorithm. In our implementation, the cloth resolution is set to 0.1 m to control the spatial granularity of the simulated cloth, while the rigidity parameter is set to 3 to balance cloth stiffness and adaptability to terrain variations. The simulation is performed for a maximum of 500 iterations with a time step of 0.65 to ensure stable convergence. Furthermore, a distance threshold of 0.03 m is used to classify ground and nonground points based on their vertical separation from the fitted cloth surface. The overall architecture of the model is shown in Figure 1.

2.1 Voxelization

For the given point cloud $\{P_n = (x_n, y_n, z_n, r_n)\}_{n=1}^{N}$, where N is the total number of points in the point cloud input. Generally, the number of points obtained from each frame of point cloud data ranges from 30,000 to 100,000. Due to limitations in GPU computational power, the actual number of point cloud points used in the model is significantly fewer than the total number of points in each frame of the original point cloud. Among them, (x_n, y_n, z_n) represents the 3D coordinates of the point cloud, while r_n denotes the reflectance intensity of the point cloud. Let us assume that the required point cloud scene has boundaries of $[X_{max}, X_{min}]$ along the x-axis, $[Y_{max}, Y_{min}]$ along the y-axis, and $[Z_{max}, Z_{min}]$ along the z-axis.

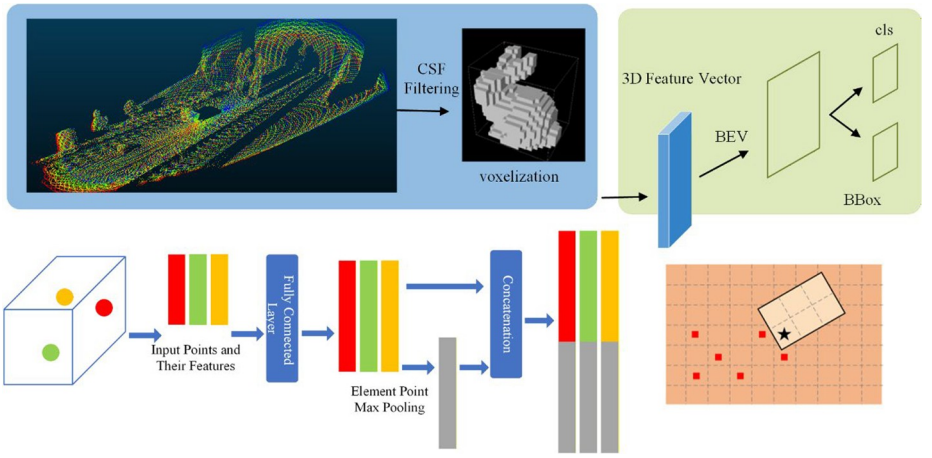


Figure 1. Algorithm flow chart
Source(s): Authors' own work

Each voxel corresponds to a 3D volume of (V_x, V_y, V_z) along the *x*-, *y*- and *z*-axes, respectively. The voxel coordinates (i_n, j_n, k_n) corresponding to point p_n are:

$$\begin{cases} i_n = \dfrac{x_i - X_{min}}{V_x} \\ j_n = \dfrac{y_i - Y_{min}}{V_y} \\ k_n = \dfrac{z_i - Z_{min}}{V_z} \end{cases} \tag{1}$$

For any voxel, its input features $F = (f_x, f_y, f_z, f_r)$ are obtained by the Voxel Feature Encoder (VFE) from the points within the voxel, as expressed by the following formula:

$$F_{local} = MLP(p_1, p_2 \ldots, p_n) \tag{2}$$

$$F_{global} = Maxpool(F_{local}) \tag{3}$$

$$F = \text{concat}(F_{global}, F_{global}) \tag{4}$$

Assume that there are *n* points within each voxel, which are then fed into a Multilayer Perceptron (MLP) for dimensionality reduction, as shown in equation (2). The resulting local features are then concatenated with the global features obtained via max pooling, thereby representing the features of the voxel.

2.2 Backbone network

After obtaining the features of the voxels, the Voxel Backbone Network can be used to further extract features. This backbone network consists of 3D sparse convolution layers and 3D submanifold convolution layers, which work together to achieve deep feature extraction and output 3D voxel features. The 3D backbone network consists of an input layer, an output layer and four sets of convolution modules. Each convolution module includes one 3D sparse convolution layer with downsampling functionality to reduce data size, as well as several 3D submanifold convolution layers that preserve the input resolution. The overall architecture of the model is shown in Figure 2.

2.3 Two-dimensional feature extraction

In point cloud processing scenarios, 3D feature maps inherently contain height information, yet direct processing of these maps is computationally expensive and suffers from high sparsity. To address this issue, an overhead view projection approach is adopted to compress the height information onto a 2D plane. This strategy effectively reduces computational burden while preserving critical spatial information, enabling a more intuitive representation of the relative positions and distances between objects and mitigating the impact of occlusions. Specifically, the 3D voxel features from the final layer of the voxel backbone network are compressed along the *z*-axis to generate an overhead view feature map, following the transformation: $F_{3D} \in \mathbb{R}^{C \times X \times Y \times Z} \rightarrow F_{2D} \in \mathbb{R}^{(C \times Z) \times X \times Y}$.

The F_{2D} feature map is then fed into a 2D feature extraction backbone network, which comprises two core components: a set of encoder modules and a multiscale feature fusion

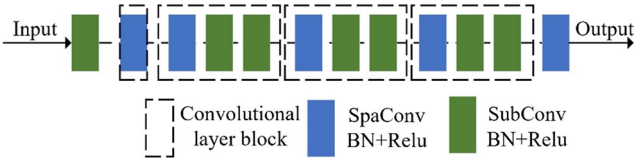


Figure 2. Schematic diagram of the backbone network
Source(s): Authors' own work

network. The encoder modules are primarily responsible for upsampling features to capture high-dimensional semantic information, while the multiscale feature fusion network concatenates 2D feature maps at different scales to generate comprehensive multiscale feature information. As illustrated in Figure 3, each encoder module consists of five convolutional layers, all using 3×3 convolution kernels. Encoder Module 1 is configured with 64 convolution kernels per layer, whereas Encoder Module 2 uses 128 convolution kernels per layer. In addition, the first layer of Encoder Module 2 uses a stride-2 downsampling operation to reduce the size of the feature map. The decoding section of the network is composed of transposed convolutional layers. Both decoding modules are equipped with 128 transposed convolution kernels but adopt distinct processing strategies: Decoding Module 1 retains the original feature map size without upsampling, while Decoding Module 2 performs double upsampling to expand the feature map dimensions.

2.4 Candidate region generation

After obtaining the 2D feature extraction, the RPN network predicts the candidate regions (Region of Interest [RoI]). The RPN network consists of three layers of parallel convolutions of 1×1, which predict the category probability of each anchor position in the candidate region, the coordinates of the anchor position bounding box and the direction of the target. The RPN network adopts an Anchor mechanism similar to that in the Reference Shi *et al.* (2020). The specific information of the Anchor is shown in Table 1.

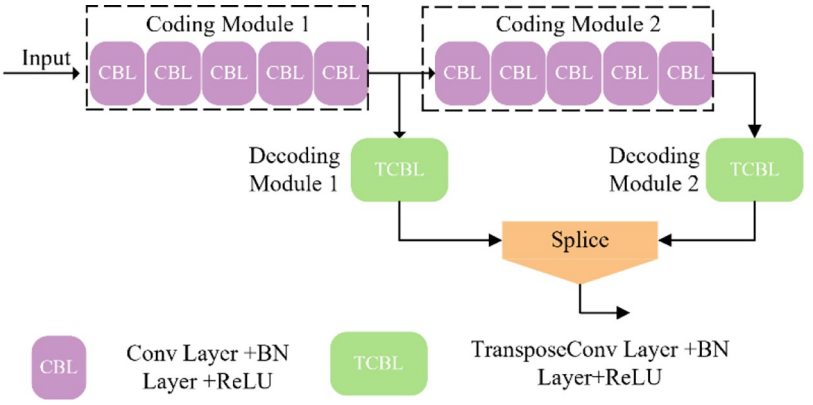


Figure 3. Schematic diagram of 2D feature extraction encoding and decoding
Source(s): Authors' own work

During the training phase, to improve detection efficiency and accuracy, up to 9,000 detection boxes are retained before applying the Non-maximum Suppression (NMS) algorithm, and 512 detection boxes are retained after NMS. The Intersection over Union (IoU) threshold for candidate regions is set to 0.8, meaning that when the IoU is greater than the threshold, the candidate region is classified as a positive sample, and when it is less than the threshold, it is classified as a negative sample. During the testing phase, the IoU threshold for NMS is set to 0.7, and up to 2,048 detection boxes are retained before NMS filtering. For the boxes detected after filtering, the top 100 candidate regions are selected based on confidence as the final results.

2.5 Voxel region of interest pooling

The RPN network predicts many candidate regions, and its core task is to aggregate the 3D features corresponding to these candidate regions. The main steps include segmentation, querying and grouping, feature aggregation and feature splicing. Detailed steps are shown in Figure 4.

2.5.1 Segmentation. The purpose of segmentation is mainly to divide the 3D candidate box into $G \times G \times G$ grids of the same size. Each candidate region (Region Proposal) is a standard rectangular prism. In this paper, it is divided into a $6 \times 6 \times 6$ space with equal intervals, and the center point coordinates of each subvoxel are taken as the center point of that subvoxel (hereinafter referred to as the subvoxel center), as shown in Figure 5.

2.5.2 Voxel query and grouping. Suppose that the feature map output by the 3D feature backbone network is F_3D. By aggregating the feature vectors within its adjacent neighborhood, we integrate local information for each subvoxel in the candidate region to obtain the feature vector of that subvoxel. In the PointNet network, a ball query method is used, while in this section, we use a voxel query method based on Manhattan distance. Voxel query involves finding up to K neighboring voxels for any subvoxel center point located in F_{3D}. The Manhattan distance between any two voxel center points can be represented as $v_1(x_1, y_1, z_1)$ and $v_2(x_2, y_2, z_2)$:

$$D(\nu_1, \nu_2)|x_1 - x_2||y_1 - y_2||z_1 - z_2|_m \tag{5}$$

Analysis reveals that for any given center point of a subvoxel, whose coordinates are represented as (i, j and k), the coordinates of all nonempty voxels in its vicinity can be expressed as a triplet of offsets, denoted as $(\Delta i, \Delta j, \Delta k)$. Thus, for any given center point coordinate, the corresponding voxel coordinates (i, j, k) are first calculated using equation (1). Then, by translating (i, j, k) within a certain range, the nonempty voxels within the neighborhood are identified. That is:

$$|\Delta i| + |\Delta j| + |\Delta k| \leq d \tag{6}$$

where d is the distance threshold and $(\Delta i, \Delta j, \Delta k) \in \{0, 1, -1\}$.

Table 1. Anchor box information table

Parameters	Box	Stone	Person
Anchor frame dimensions	[0.4, 0.4, 0.4]	[0.2, 0.2, 0.2]	[0.8, 0.6, 2]
Anchor frame rotation angle	[0, 1.57]	[0, 1.57]	[0, 1.57]
Anchor frame height	−1.78	−1.78	[−0.6]
Feature map stride	8	8	8
Align object center	False	False	False
Mismatch threshold	0.35	0.35	0.35
Matching threshold	0.5	0.5	0.5

Source(s): Authors' own work

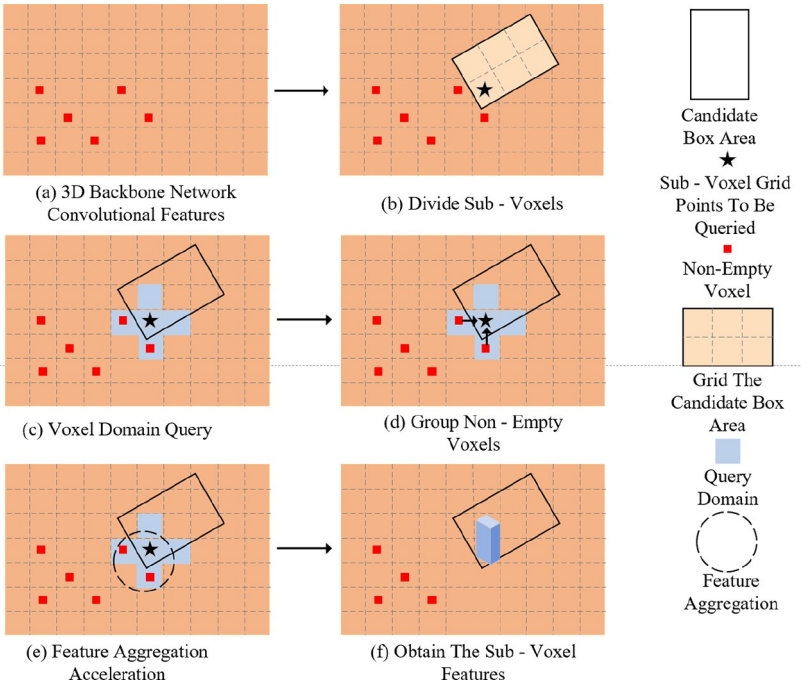


Figure 4. Process diagram of voxel region of interest (RoI) pooling
Source(s): Authors' own work

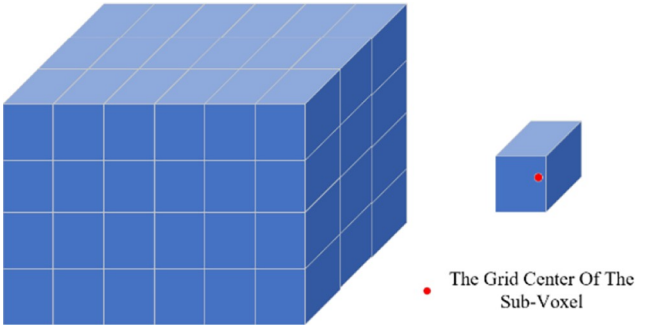


Figure 5. Schematic diagram of voxel segmentation
Source(s): Authors' own work

2.5.3 Feature aggregation. When aggregating features of voxels within the neighborhood of the subvoxel center point, Voxe1 R-CNN designs a method that is faster than the PointNet method to improve aggregation efficiency. The specific aggregation method is as follows. The main difference is that, instead of performing dimension-raising

operations on all feature information and coordinate information as in the original method, which results in a large amount of redundant computation, this section performs dimension-raising operations only on the required features, thereby avoiding redundant computation of voxel features. First, for all nonempty voxels, an MLP with shared weights is used to extract spatial features. Then, the voxel query method based on Manhattan distance mentioned in the previous section is used to divide the neighboring voxels around each subvoxel center point. If the total number of voxels in F_{3D} is N, the formula for a subvoxel center point is as follows:

$$\eta_1 = MLP\left(Maxpool\left(S_f + MPL\left(S_p^{'}\right)\right)\right) \tag{7}$$

Assuming that $S_v = \{v_1, v_2 \cdots, v_K\}$ denote the K nonempty voxels within its neighborhood, the feature vectors corresponding to the nonempty voxels are denoted as $S_f = \{f_1, f_2 \cdots, f_K\}$, and the relative coordinates with respect to the voxel center are denoted as $S_p = \{p_1, p_2 \cdots, p_K\}$.

In equation (7), Maxpool is the maximum pooling function for feature dimensions, and MLP denotes a multilayer perceptron with shared weights. The purpose is to perform element-wise addition between S_f and $S_p^{'}$ and then apply maximum pooling on the feature dimensions to obtain the feature of the subvoxel center point. $S_p^{'}$ is the position feature extracted from the vectors in S_p using MLP, as shown in Figure 6.

2.6 Head detection network

After completion of the pooling step for the voxel RoI, the feature vectors of the candidate regions are obtained and then fed into the head detection network. This head detection network consists of two parallel modules and a feature transformation module: one is the 3D Bounding Box regression module, and the other is the confidence prediction module. First, the feature transformation module converts the candidate box features into feature vectors,

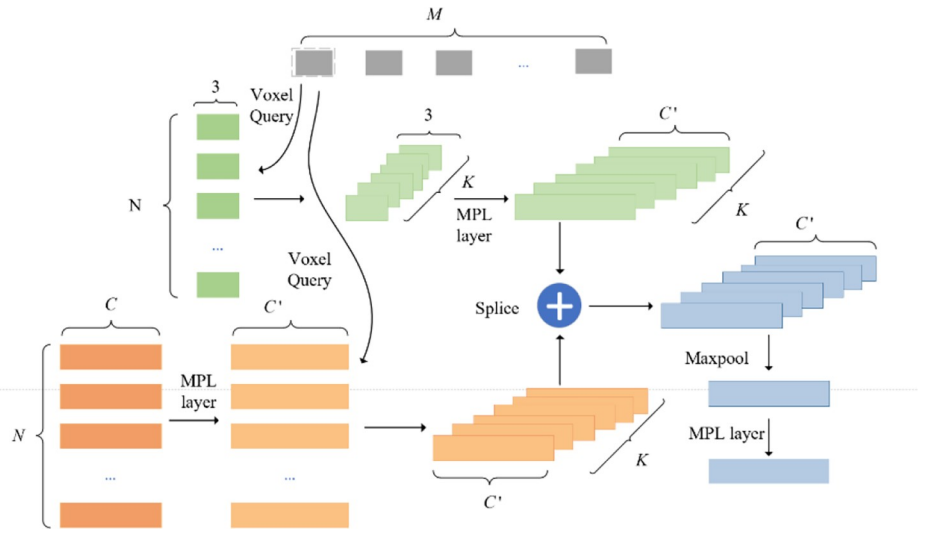


Figure 6. Schematic diagram of the acceleration module
Source(s): Authors' own work

which are then input into the 3D boundary Box regression module and the confidence prediction module. The feature transformation module consists of two Full Connect Layers, while the modules for predicting the 3D Bounding Box and confidence each consist of three fully connected layers. All modules include Batch Normalization (BN) and ReLU activation functions.

2.7 Loss function

Voxel R-CNN is a two-stage detection method. When calculating the loss function, it consists of two parts: the loss of the region proposal network ($\mathcal{L}_{RPN}$) and the loss of the head detection network ($\mathcal{L}_{Head}$). The loss function of the region proposal network includes the classification loss function and the regression loss function for anchor boxes; while the loss function of the head detection network is composed of the classification loss and the regression loss for candidate boxes.

The formula for the loss function of the region proposal network is as follows:

$$\mathcal{L}_{RPN} = \frac{1}{N_{fg}}\left[\sum_i \mathcal{L}_{cls}\left(p_i^a, c_i^*\right) + \prod\left(c_i^* \geq 1\right)\sum_i \mathcal{L}_r\left(\delta_i^a, t_i^*\right)\right] \tag{8}$$

where p_i^a and δ_i^a are the predicted class and the predicted localization transformation parameters of the ith anchor box to the corresponding ground truth object box, respectively; c_i^* is the true class of the ith anchor box; t_i^* is the parameter for transforming the ith anchor box into the corresponding true target box; N_{fg} is the total number of foreground anchor boxes; and Π is the indicator function, indicating that the regression loss calculation only covers positive sample anchor boxes. Among them, $\mathcal{L}_{cls}$ is the Focal Loss, as shown in equation (9), and $\mathcal{L}_r$ is the Smooth L1 Loss, as shown in equation (10):

$$\mathcal{L}_{Focal}(p, c) = \begin{cases} -\alpha(1-p)^{\gamma} log p, & c = 1 \\ -(1-\alpha)p^{\gamma} log(1-p), & c = 0 \end{cases} \tag{9}$$

where p is the predicted value, α is the sample weight coefficient and γ is the difficulty modulation coefficient, typically set to 2 (Lin *et al.*, 2017):

$$\mathcal{L}(x, y) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases} 0.5 \times (y_i - x_i)^2, & if|y_i - x_i| < 1 \\ |y_i - x_i| - 0.5, & otherwise \end{cases} \tag{10}$$

where x_i is the predicted value and y_i is the true value.

In the head detection network, the confidence prediction target value of the candidate box depends on the IoU between the candidate box and the true target box:

$$l_i^*(IoU_i) = \begin{cases} 0, & IoU_i < \theta_L \\ \dfrac{IoU_i - \theta_L}{\theta_H - \theta_L}, & 0 \leq IoU_i < \theta_H \\ 1, & IoU_i > \theta_H \end{cases} \tag{11}$$

where IoU_i is the intersection-over-union ratio between the ith predicted 3D bounding box and the corresponding true 3D bounding box, and θ_L and θ_H are the IoU thresholds for positive and negative sample candidate boxes, respectively.

The loss function of the head detection network consists of a confidence prediction branch and a bounding box regression prediction branch, with the following optimization objective:

$$\mathcal{L}_{head} = \frac{1}{N_s}\left[\begin{array}{c}\sum_i \mathcal{L}_{cls}\left(p_i, l_i^*(IoU_i)\right) + \\ \Pi(IoU_i \ldots \theta_{reg}) \sum_i \mathcal{L}_{reg}(\delta_i, t_i^*)\end{array}\right] \tag{12}$$

where $\mathcal{L}_{cls}$ represents confidence prediction, using the Binary Cross Entropy Loss function. $\mathcal{L}_{reg}$ represents bounding box regression prediction using the Smooth L1 Loss. N_s is the number of candidate boxes sampled during the training phase; $\Pi(IoU_i \ldots \theta_{reg})$ indicates that only candidate boxes with an IoU greater than the threshold θ_{reg} relative to the corresponding ground truth box are used for regression loss calculation. p_i and b_i are the category confidence prediction value and 3D bounding box prediction value for the ith target, respectively. $l_i^*(IoU_i)$ is the category confidence label calculated using IoU, whose formula is shown in equation (11), and t_i^* is the target label.

Regarding class imbalance among different object orientations, the orientation prediction is formulated as a classification problem by discretizing the continuous heading angle into multiple direction bins. A cross-entropy loss is used for direction classification, which naturally accounts for imbalanced category distributions by optimizing the predicted probability distribution over all direction classes. This formulation encourages the model to focus on discriminative learning among different orientation categories and improves overall robustness under uneven sample distributions.

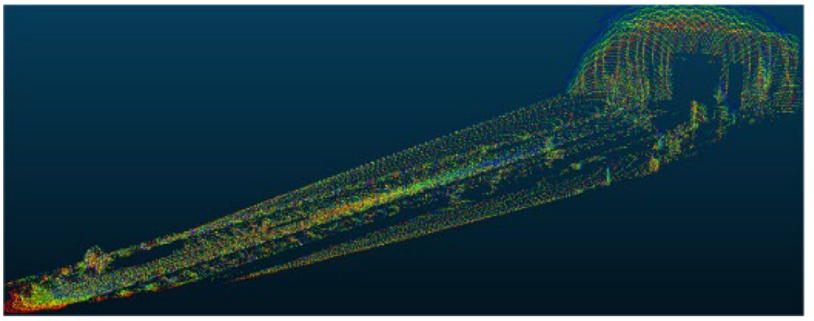
After direction category prediction, a Smooth L1 loss is further applied for residual orientation regression. The Smooth L1 loss is less sensitive to outliers caused by noise or abnormal predictions, ensuring stable gradient updates during training. When the prediction error is small, it behaves similarly to L2 loss, enabling fine-grained optimization near the optimum and improving the accuracy and stability of 3D orientation estimation.

3. Experiments and analysis of results

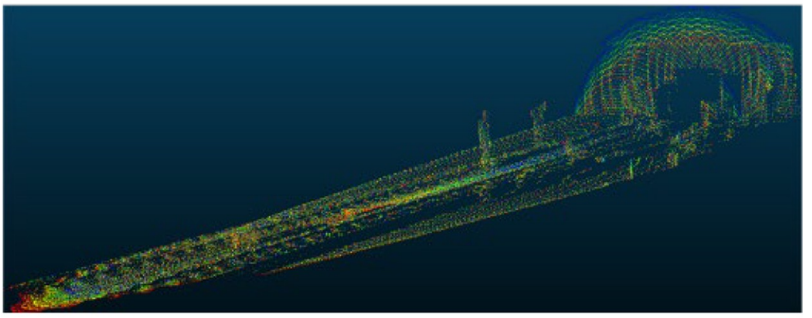
3.1 Track anomaly detection point cloud data set

A total of 3,951 data points were collected at the Huanghua Laboratory, the data format being .pcd. The point cloud data were acquired using a fixed LiDAR sensor deployed in a controlled railway test environment. Manual annotation was performed by labeling 3D bounding boxes for each intrusion object, and all samples were collected under normal operating conditions without severe weather interference. Because the current mainstream point cloud processing code frameworks use the .npy file format, the data must be converted to .npy format. Among these, 2,012 samples were used for training and testing, with an 8:2 ratio between the training set and the testing set. The sample labels are categorized into three types: stones, boxes and pedestrians. The primary distinction between stones and boxes lies in their size; stones are smaller, while boxes are larger. The number of box samples used for training is 5,097, the number of pedestrian samples is 990 and the number of stone samples is 7,350. Due to the presence of noise points, some samples with fewer points need to be filtered out. After filtering, the number of box samples is 5,066, the number of pedestrian

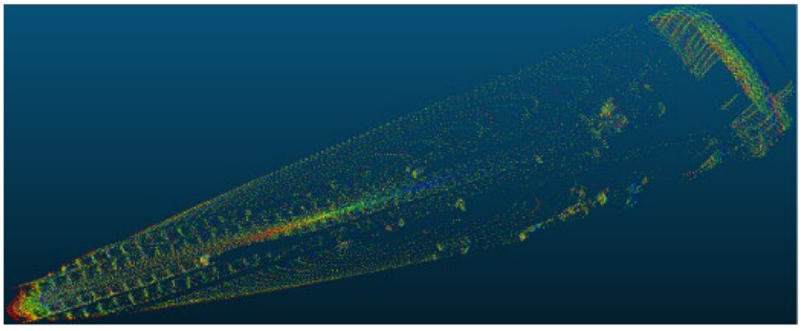
samples is 988 and the number of stone samples remains 7,350. Class imbalance among box, pedestrian and stone categories was addressed during training by category-aware sampling, ensuring that each minibatch contained a balanced proportion of samples from different classes. The display diagram of point cloud data is presented in Figure 7.



(a)



(b)



(c)

Figure 7. Display diagram of point cloud data
Note(s): (a) box; (b) Person; and (c) Stone (small goal)
Source(s): Authors' own work

3.2 Experimental setup

In this section, the range of the point cloud in the scene is adjusted. After adjustment, the range along the *x*-axis is $[0m, 40m]$, the range along the *y*-axis is $[-2.4m, 2.4m]$ and the range along the *z*-axis is $[-2m, 2m]$. Each voxel has a length of 0.05 m along the *x*-axis, 0.05 m along the *y*-axis and 0.01 m along the *z*-axis, with a maximum of five sampling points.

All models used in this paper are implemented using PyTorch and trained on a NVIDIA A6000 GPU (48G). The CPU is an Intel Xeon Silver 4314 @2.4 GHz. During training, this chapter uses the Adam optimizer with an exponential decay rate of 0.9 for the first momentum estimate and 0.999 for the second momentum estimate as the network optimization method. The initial learning rate for model training is set to 0.01. Throughout the learning process, the cosine annealing strategy is used to dynamically adjust the learning rate to optimize model training performance. In addition, the pretrained weight file from the KITTI data set is used as the initial weight file. The batch size is set to 4, and the entire training process consists of 80 epochs.

3.3 Experimental evaluation indicators

In this experiment, the following evaluation metrics were used:

IoU: Measures the consistency between predicted boxes and ground truth boxes by calculating the overlap between them. The value ranges from 0 to 1. In 3D object detection, this concept is extended to 3D IoU. Using 3D IoU, the accuracy of the detection results can be assessed more accurately, thereby optimizing the performance of the model, as shown in the following equation:

$$IoU(A, B) = \frac{A \bigcap B}{A \bigcup B} \tag{13}$$

3D Bounding Box Accuracy: Considering the presence of small target objects, only positive samples with an IoU value of 50% or higher are required. Thus, precision and recall are expressed as follows:

$$\text{precision} = \frac{TP}{TP + FP} \tag{14}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

In equations (14) and (15), precision denotes precision, recall denotes recall, TP denotes the number of true positive samples, FP denotes the number of false positive samples and FN denotes the number of false negative samples.

Average Orientation Similarity (AOS): To measure the prediction accuracy of this angle, KITTI (Geiger *et al.*, 2013) designed the Average Orientation Similarity metric, defined as follows:

$$AOS = \frac{1}{11} \sum_{r \in \{0, 0.1, \cdots, 1\}} \underset{\widetilde{r}:\widetilde{r} \ldots r}{max}\, s(\widetilde{r}) \tag{16}$$

$$s(r) = \frac{1}{|D(r)|} \sum_{i \in D(r)} 1 + \frac{cos\Delta_{\theta}^{(i)}}{2} \delta_i \tag{17}$$

where r is the recall rate.

The accuracy calculation method defined by the KITTI data set is to calculate the average accuracy at 40 evenly spaced recall points, as shown in equations (18):

$$AP = \frac{1}{|R_{40}|} \sum_{r \in R_{40}} f_p(r) \tag{18}$$

$$f_p(r) = max_{\{r':r' \geq r, r' \in R_{40}\}} \rho(r') \tag{19}$$

where AP (Average Precision) represents the average precision rate, and $\rho(r')$ denotes the precision rate when the recall rate is r'.

In KITTI, three metrics – AP_{BEV}, AP_{3D} and AOS – are used to evaluate detection performance. AP_{BEV} represents detection accuracy from a BEV perspective, AP_{3D} represents detection accuracy for true 3D bounding boxes and AOS measures the model's detection accuracy for target heading angles, with higher values indicating greater accuracy.

3.4 Experimental results

To comprehensively evaluate the multiclass detection capability of the proposed method, we report class-wise detection performance for pedestrians, stones and boxes in terms of BEV AP, 3D AP and AOS. Figure 8 shows the effect of the CSF filtering algorithm (Zhang *et al.*, 2016) on data processing from different perspectives. The blue dots represent the point cloud below the track surface, while the red dots represent the points that need to be retained. As can be seen from the intuitive display, the small foreign objects in the unfiltered CSF point cloud are very close to the track surface and difficult to identify. After filtering, the foreign objects above the track surface can be identified more clearly.

To provide a comprehensive quantitative evaluation beyond qualitative visualization, we report detailed class-wise detection performance using multiple metrics, including BEV Average Precision (AP), 3D AP and AOS. In addition, ablation experiments and comparative evaluations are conducted to analyze the contribution of individual modules and to benchmark the proposed method against representative baseline approaches.

During the experiment, due to the fact that boxes and stones have similar topological structures in point clouds, misclassifications occurred when distinguishing between them. Therefore, in the subsequent experiments of this paper, boxes and stones were grouped as a single category.

Through experimentation, it was found that the point cloud network topology of Box and Stone is similar, and there is a certain degree of identification error during the annotation process. Therefore, it was considered to merge the Stone and Box in the point cloud into one category. However, because the Stone is relatively small in size in space, it is easily overlooked by the network, leading to missed judgments.

3.4.1 Ablation experiment. To better illustrate the role of the model in the components during the experimental process, the experimental results of the ablation experiments in this paper are shown in Tables 2 and 3. In these tables, "√" indicates that the option was selected, bold text denotes the optimal option and underlined text denotes the second-best option. In addition, it is important to note that the experiments on the detection head mainly focused on the selection of a 2D backbone network. Through the experimental results, the detection performance was compared and analyzed in different data set categories (2-class and 3-class) and object categories (Box, Person and Stone).

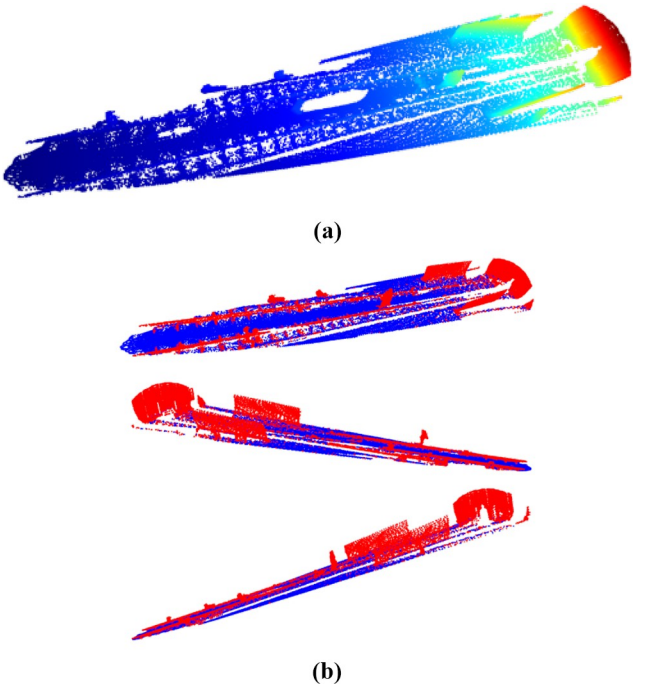


Figure 8. Intuitive comparison chart of point cloud data filtered by cloth simulation filter and unfiltered point cloud data
Note(s): (a) Point cloud data visualization diagram; and (b) CSF filter intuitive display diagram
Source(s): Authors' own work

Table 2. Ablation experiments of CSF filtering and detection heads on three types of data sets

CSF filtering	Sensor head	Foreign object category	0.5@bev	0.5@3d	0.5@aos
√	√	Box	88.93	78.46	78.12
		Person	79.92	79.18	57.36
		Stone	52.39	42.49	60.36
	√	Box	87.85	77.98	72.61
		Person	80.23	78.05	58.69
		Stone	46.19	40.93	62.56
√		Box	86.48	76.48	71.28
		Person	79.84	70.25	56.22
		Stone	43.25	39.33	58.47

Source(s): Authors' own work

Table 2 presents the ablation results on the three-class data set (box, pedestrian and stone), evaluating the impact of CSF and the detection head module. When both CSF filtering and the detection head are enabled, the proposed method achieves the best overall performance across most metrics. Specifically, for the box category, the BEV AP reaches 88.93%, the 3D

Table 3. Ablation experiments of CSF filtering and detection heads on two types of data sets

CSF filtering	Sensor head	Foreign object category	0.5@bev	0.5@3d	0.5@aos
√	√	Person	80.45	71.26	81.59
		Stone	68.18	66.21	89.67
	√	Person	71.42	70.90	81.27
		Stone	71.39	63.50	93.64
√		Person	70.39	59.43	73.18
		Stone	65.29	58.91	81.07

Source(s): Authors' own work

AP reaches 78.46% and the AOS reaches 78.12%. Pedestrian detection shows relatively stable performance, with a 3D AP of 79.18%, benefiting from its distinctive geometric structure in point clouds. Stone detection remains more challenging due to its small spatial size and sparse point distribution; nevertheless, enabling CSF filtering improves the BEV AP by effectively suppressing ground-level interference. Overall, these results demonstrate that CSF filtering and the detection head jointly contribute to improved multiclass detection accuracy.

When both CSF filtering and the detection head were enabled, as shown in Table 2, the model achieved an increase of 1.08 in 0.5@bev for the Box category in the 3-class data set; 0.5@3d increased by 0.48; 0.5@aos increased by 5.51; in the Person category, 0.5@bev decreased by 0.31; 0.5@3d increased by 1.13; 0.5@aos decreased by 1.33; in the Stone category, 0.5@bev increased by 6.2; 0.5@3d increased by 1.56; 0.5@aos decreased by 2.2. Overall, CSF filtering has a positive effect on small object detection.

Compared to configurations that only use CSF filtering, when CSF filtering and the detection head module are enabled simultaneously, the model achieves improvements in the 0.5@bev metric of 2.45, the 0.5@3d metric of 1.98 and the 0.5@aos metric of 6.84 for the Box category in the 3-class data set; improvements in the 0.5@bev metric of 0.08, the 0.5@3d metric of 8.93 and the 0.5@aos metric of 1.14 for the Person category. The 0.5@bev metric for the Stone category improved by 9.14, the 0.5@3d metric improved by 3.16 and the 0.5@aos metric improved by 1.89. All metrics in all categories showed improvements, and some metrics showed significant improvements. This indicates that the detection head used in this paper is crucial for the detection accuracy (0.5@bev, 0.5@3d, etc.) of the three-category data, validating the effectiveness and necessity of these components.

Table 3 reports the ablation results on the two-class data set, where stones and boxes are merged into a single category to reflect practical deployment scenarios. Compared with configurations using only CSF filtering or only the detection head, the combination of both modules consistently improves detection performance. For the pedestrian category, the BEV AP increases to 80.45% and the 3D AP reaches 71.26%, indicating robust detection capability. For the merged stone–box category, the proposed configuration achieves a 3D AP of 66.21% and an AOS of 89.67%. These results suggest that merging geometrically similar small objects can effectively reduce misclassification errors and enhance detection robustness under sparse LiDAR sampling.

As shown in Table 3, on the second-category data set, compared to using only the detection head configuration, enabling both CSF filtering and the detection head simultaneously resulted in the following improvements in the Person category: an increase of 9.03 in the 0.5@bev metric; an increase of 0.36 in the @3d metric; and an increase of 0.32 in the 0.5@aos metric. In the Stone category, there was a decrease of 3.21 in the 0.5@bev

metric; the @3d metric increased by 2.71; and the 0.5@aos metric decreased by 3.97. Overall, for the Person category, allowing both CSF filtering and the detection head generally improved most metrics; for the Stone category, metrics showed mixed results, indicating that CSF filtering has varying effects on detection performance across different categories of foreign objects.

Compared to using CSF filtering alone, when CSF filtering and the detection head are used together, in the Person category, the 0.5@bev metric increased by 10.06; the 0.5@3d metric increased by 11.83; and the 0.5@aos metric increased by 8.41; in the Stone category, the 0.5@bev metric increased by 2.89; the 0.5@3d metric increased by 7.3, and the 0.5@aos metric increased by 8.6. In both foreign object categories, all metrics improved, with values significantly increasing after enabling the detection head, indicating that the detection head effectively enhances the model's accuracy in foreign object detection. In summary, the detection head enables the model to better adapt to detecting different types of foreign object, enhancing the model's generalization ability across various foreign object detection scenarios.

To further clarify the independent contributions of individual modules, we analyze the effects of CSF filtering and the detection head separately. CSF filtering primarily enhances small-object detection by suppressing ground-level clutter and strong reflections from sleepers and ballast. This effect is particularly evident in the stone category, where CSF improves BEV AP by effectively separating low-height objects from the track surface.

In contrast, the detection head module mainly contributes to improved localization accuracy and category discrimination by refining RoI features through voxel-level aggregation. This module yields consistent gains in 3D AP across all object categories, especially for pedestrians and boxes with relatively stable geometric structures.

When both modules are enabled, their effects become complementary: CSF filtering reduces false positives caused by ground interference, while the detection head improves spatial feature representation and classification confidence. This synergy explains the superior overall performance observed in the full configuration.

3.4.2 Comparative experiment. To better illustrate the superiority of the method proposed in this paper, comparative experiments were conducted focusing on the method described in this chapter, 3DSSD, and TED. The detection performance across different data set categories and object categories (Box, Person and Stone) was analyzed, with the specific results shown in Table 4. In the three-category data set scenarios, the method proposed in this chapter performed exceptionally well. In the Box category, the 0.5@bev metric reached 88.93, far exceeding 3DSSD's 15.84 and TED's 62.19. Under the method proposed in this

Table 4. Comparative experiments conducted on three types of data sets

Method	Foreign object category	0.5@bev	0.5@3d	0.5@aos	FPS
Methods in this chapter	Box	88.93	78.46	78.12	21
	Person	79.92	79.18	57.36	
	Stone	52.39	40.93	60.36	
3DSSD	Box	15.84	6.52	15.22	14
	Person	59.38	36.86	60.79	
	Stone	0	0	0	
TED (Wu *et al.*, 2022)	Box	62.19	54.81	63.05	9
	Person	0	0	1.23	
	Stone	9.09	9.09	9.07	

paper, the 0.5@3d metric reached 78.46, achieving over a 10-fold improvement compared to 3DSSD's 6.52, and also showing a significant advantage over TED's 54.81. Especially in the detection of the Stone category, the proposed method outperforms 3DSSD and TED in multiple metrics. In the two-category data set, the proposed method achieves a 0.5@bev metric of 80.45 in the Person category and an 0.5@aos metric of 89.67 in the Stone category, both leading the results of 3DSSD and TED in the same scenarios. Experiments demonstrate that the proposed method achieves a higher detection accuracy across multiple categories compared to comparison methods, highlighting its effectiveness and superiority in detecting foreign objects in railways.

Table 4 compares the proposed method with representative point cloud–based baselines, including 3DSSD and TED, under the same data set settings. In the three-class evaluation, the proposed approach significantly outperforms the baseline methods across all categories. For example, in the box category, the proposed method achieves a BEV AP of 88.93% and a 3D AP of 78.46%, whereas 3DSSD and TED obtain substantially lower scores. Notably, for small objects such as stones, the baseline methods exhibit limited detection capability, while the proposed method maintains reasonable performance due to CSF-based ground filtering and voxel-level feature aggregation. These results demonstrate the effectiveness and robustness of the proposed framework for railway track intrusion detection.

A closer examination of the quantitative results reveals that performance gains vary across object categories and evaluation metrics. For example, improvements in BEV AP are more pronounced for medium- and large-sized objects, while gains for small objects remain relatively modest. This observation suggests that point density and object scale continue to play a critical role in detection performance, particularly under sparse LiDAR sampling conditions.

Although the proposed framework shows overall performance advantages, the experimental results also indicate certain limitations. In particular, detection performance for small objects remains sensitive to noise and sparsity in the point cloud data. These findings suggest that further improvements may require enhanced feature representation or multisensor fusion, rather than solely relying on architectural modifications.

The method proposed in this paper achieves an FPS of 21, which is the fastest among the three compared methods and fully meets the real-time requirement for on-site monitoring scenarios. 3DSSD follows with an FPS of 14, maintaining basic real-time performance, while the FPS of TED is only 9, failing to reach the threshold of real-time detection. Combined with the detection accuracy metrics (e.g. 0.5@bev, 0.5@3d, 0.5@aos), the proposed method outperforms the other two methods in both precision and speed, achieving a better balance between detection performance and inference efficiency.

4. Conclusion

In this paper, we present a cloud-based point-based approach to detect foreign objects on railways. Conventional image-based video methods are highly susceptible to adverse weather conditions such as variations in lighting, smoke, rain and fog. To address the need for precise localization and size estimation of track-level foreign objects, as well as the identification of potential risk factors associated with foreign object intrusion, we propose a detection framework based on point cloud data. Specifically, the CSF algorithm is used to remove ground points from railway scenes, effectively mitigating the impact of strong radar reflections from sleepers and ballast that typically hinder the detection of small objects. In addition, an improved Voxel R-CNN model is introduced, customized to the characteristics of the rail scenarios. The proposed model first voxelizes the point cloud and encodes voxel features, which are then processed by a backbone network composed of submanifold and

standard 3D convolutions for hierarchical feature extraction. The extracted features are projected into a BEV representation for the identification of the RoI. Subsequently, a voxel query mechanism and an accelerated feature aggregation module are applied to extract corresponding 3D features, which are passed into the detection head to generate 3D bounding boxes along with category predictions. The experimental results demonstrate that the proposed method achieves average detection accuracies of 80% for pedestrians, 80% for stones and 40% for boxes, confirming its effectiveness for foreign object detection in railway environments.

5. Limitations and future work

Despite the promising performance of the proposed method, several limitations should be acknowledged. First, the detection accuracy of small objects (e.g. stones) remains susceptible to point cloud sparsity and LiDAR resolution, especially under long-range sensing conditions. Second, the current model is trained and validated exclusively on data collected from a controlled railway test environment, and its generalization capability to diverse track structures, sensor configurations and environmental conditions requires further verification. Notably, this study does not include a direct experimental comparison with video-based detection methods under adverse weather conditions. Although prior studies have demonstrated the superior robustness of LiDAR sensing in low-illumination and visually challenging scenarios, a rigorous cross-modality comparison necessitates synchronized multisensor data collected under identical weather conditions – an issue that will be addressed in future work.

In future work, we plan to incorporate multisensor fusion and domain adaptation strategies to enhance robustness across diverse railway scenarios. In addition, lightweight model optimization will be explored to facilitate real-time deployment on embedded railway monitoring platforms.

References

Charles, R.Q., Su, H., Kaichun, M. and Guibas, L.J. (2017), "PointNet: deep learning on point sets for 3D classification and segmentation[C]", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77-85.

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. (2013), "Vision meets robotics: the KITTI dataset[J]", *The International Journal of Robotics Research*, Vol. 32 No. 11, pp. 1231-1237.

Huang, Y., Zhang, Z., Tao, Y. and Hu, H. (2022), "Quantitative risk assessment of railway intrusions with text mining and fuzzy rule-based bow-tie model[J]", *Advanced Engineering Informatics*, Vol. 54, p. 101726.

Lang, A.H., Vora, S., Caesar, H., Zhou, L. and Beijbom, O., (2019), "PointPillars: fast encoders for object detection from point clouds[a]", *arXiv*.

Li, B., Zhang, T. and Xia, T. (2016), "Vehicle detection from 3D lidar using fully convolutional network [a]", *arXiv*.

Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P. (2017), "Focal loss for dense object detection [C]", 2017 IEEE International Conference on Computer Vision (ICCV). 2999-3007.

Qi, C.R., Yi, L., Su, H. and Guibas, L.J. (2017), "PointNet++: deep hierarchical feature learning on point sets in a metric space[C]", *Advances in Neural Information Processing Systems*, Vol. 30.

Shi, S., Guo, C., Jiang, L., Wang, Z. and Li, H. (2020), "PV-RCNN: oint-Voxel feature set abstraction for 3D object detection[C]", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). *Seattle, WA*: *IEEE*, pp. 10526-10535.

Shi, S., Wang, X. and Li, H. (2019), "PointRCNN: 3D object proposal generation and detection from point cloud[a]", *arXiv*.

Wu, H., Wen, C., Li, W., Li, X., Yang, R. and Wang, C. (2022), "Transformation-equivariant 3D object detection for autonomous driving[a]", *arXiv*.

Yan, Y., Mao, Y. and Li, B. (2018), "SECOND: sparsely embedded convolutional detection[J]", *Sensors*, Vol. 18 No. 10, p. 3337.

Yang, B., Luo, W. and Urtasun, R. (2019), "PIXOR: real-time 3D object detection from point clouds[A]", *arXiv*.

Yang, Z., Sun, Y., Liu, S. and Jia, J. (2020), "3DSSD: point-based 3D single stage object detector[C]", 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). *Seattle, WA*: *IEEE*, pp. 11037-11045.

Zhang, W., Qi, J., Wan, P., Wang, H. and Yan, G. (2016), "An easy-to-use airborne LiDAR data filtering method based on cloth simulation[J]", *Remote Sensing*, Vol. 8 No. 6, pp. 501-562.

Zhang, Z., Chen, P., Huang, Y., Dai, L., Xu, F. and Hu, H. (2024), "Railway obstacle intrusion warning mechanism integrating YOLO-based detection and risk assessment[J]", *Journal of Industrial Information Integration*, Vol. 38.

Zhou, Y. and Tuzel, O. (2018), "VoxelNet: end-to-end learning for point cloud based 3D object detection[C]", *2018 Ieee/Cvf Conference on Computer Vision and Pattern Recognition (Cvpr)*, IEEE, New York, pp. 4490-4499.

Corresponding author
Xiukun Wei can be contacted at: xkwei@bjtu.edu.cn