Railway Traffic Object Detection Using Differential Feature Fusion Convolution Neural Network

Tao Ye, Xi Zhang, Yi Zhang, and Jie Liu

***Abstract*— Railway shunting accidents, in which trains collide with obstacles, often occur because of human error or fatigue. It is therefore necessary to detect traffic objects in front of the trains and inform the driver to take timely action. To detect these objects in railways, we proposed an object-detection method using a differential feature fusion convolutional neural network (DFF-Net). DFF-Net includes two modules: the prior object-detection module and the object-detection module. The prior module produces initial anchor boxes for the subsequent detection module. Taking the initial anchor boxes as input, the object-detection module applies a differential feature fusion sub-module to enrich the sematic information for object detection, enhancing the detection performance, particularly for small objects. In experiments conducted on a railway traffic dataset, compared with the current state-of-the-art detectors, the proposed method exhibited significant higher performance and was more effective and more efficient than the other methods for object detection in railway tracks. Additionally, evaluation results based on PASCAL VOC2007 and VOC2012 indicated that the proposed method was significantly better than the state-of-the-art methods.**

***Index Terms*— Railway traffic object detection, differential feature fusion convolutional neural network, prior module.**

I. Introduction

With the rapid growth of railway transport, an increasing amount of attention has been paid to the reliability and safety of railways. However, shunting accidents in railways often occur [1] for the following two reasons: (1) the train driver makes an erroneous judgement about the distance of obstacles in front of the train owing to human error or fatigue, causing the train to collide with an obstacle; (2) the train driver misoperates the signals near the tracks, causing the train to move at an inappropriate time [2]. With the development of artificial intelligence, machine vision is effective for avoiding railway shunting accidents. In this study, we focus on detecting traffic obstacles on the railway tracks in the shunting mode.

In recent years, many engineers have focused on the development of driver-assistance systems to improve railway safety, and numerous approaches and systems have been designed to ensure driving safety [3]–[7]. However, few driver-assistance systems are differentially designed to allow the detection of obstacles in front of moving trains [1], [2], [8]. Furthermore, a detection system is more suitable if it can be installed in front of the train as compared to the other systems, which can only be placed in a fixed position (for example, the crossing or railway sides). This is particularly important because obstacle collisions can occur anywhere along the railway.

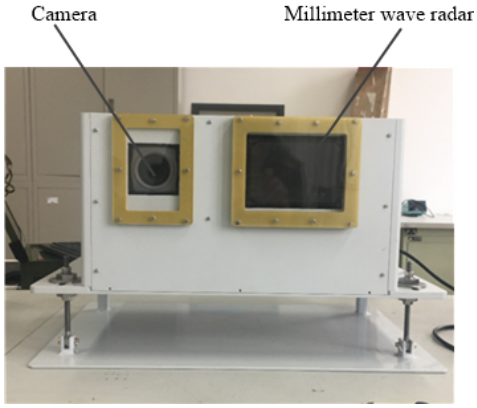


Fig. 1. Railway object-detection system.

In this study, using a graphics processing unit (GPU) and a deep convolutional neural network (CNN) [9], [10], we developed a novel railway object-detection system that can be installed in front of a train to automatically detect objects in front of the train in the shunting mode. As shown in Figure 1, a camera captures images for the proposed detection algorithm, and the distances between the detected object and train are measured using millimeter-wave radar. When the system detects obstacles on the railway track in front of the train, an alarm sends a voice to alert the driver of possible collisions. Seven different types of objects, i.e., the Bullet Train, Railway Straight, Railway Left, Railway Right, Pedestrian, Helmet, and Spanner, can be detected using the proposed method. The type of railway track (i.e., Railway Straight, Railway Left, or Railway Right) is detected to determine whether the trains are running on curved railway tracks. In the shunting mode, the driver mainly relies on the attendant to observe the railway conditions ahead. The objective of detecting railway tracks is to determine whether the train is running on a curved railway. When the train is running on a curved railway, the line of sight in front of the train driver is easily blocked. If our system detects a curved railway, it reminds the driver to drive safely by voice. The objective of detecting other types of objects is to allow the train drivers to detect objects in a timely manner. The motivation for detecting pedestrians and trains ahead is to allow the train attendant to identify danger in time. When our system detects a pedestrian or train on the railway

Manuscript received June 3, 2019; revised November 7, 2019 and December 27, 2019; accepted January 24, 2020. The Associate Editor for this article was D. F. Wolf. *(Corresponding author: Tao Ye.)*

The authors are with the School of Mechanical Electronic & Information Engineering, China University of Mining and Technology–Beijing, Beijing 100083, China (e-mail: ayetao198715@163.com).

Digital Object Identifier 10.1109/TITS.2020.2969993

ahead, the train attendants are informed by a voice prompt and take the appropriate measures to prevent a collision [1]. Additionally, for reducing unnecessary losses, the system can detect two types of items that workers often leave on railways: helmets and spanners.

In this study, we mainly focused on detecting traffic objects on railway tracks. Accordingly, we classified obstacles according to their hazard levels. We considered trains and pedestrians to be the first level of hazard. The curved railway and the remainder of the tools (the helmet and spanner) were regarded as the second and third levels of hazard, respectively. When the proposed method detects the crowed objects, the system sends a voice alerting the driver according to the highest level of hazard. The central point of the detection system is the design of the differential feature fusion neural network (DFF-Net) to detect obstacles on railway tracks. The DFF-Net combines the advantages of the two-stage method (i.e., Faster R-CNN [11]) and the one-stage method (i.e., single-shot detector (SSD) [12]). In contrast to the Faster R-CNN, we employ pyramid feature maps to detect objects of multiple scales and use differential feature fusion designs to construct robust features for object detection. Compared with the SSD, which utilizes tiled default boxes for detecting objects directly, the DFF-Net employs the prior detection module to produce the initial anchor boxes, and the object-detection module feeds the prior boxes as the input for further detection, enhancing the detection performance.

The contributions of our study are as follows.

(1) We proposed the DFF-Net, which is an end-to-end object detection network with a fully convolutional network architecture. The network comprises two modules: the prior objection-detection module and the object-detection module. The former generates the initial anchors for the subsequent modules. Taking the prior anchor boxes as input, the object-detection module obtains fully fused feature information by using sub-modules, e.g., the differential feature fusion module, for object detection. The proposed DFF-Net achieves a good tradeoff between accuracy and real-time performance.

(2) Effective training strategies such as negative mining and various data augmentation techniques tricks were employed to achieve a high detection accuracy. With a low resolution of 320 $\times$ 320 pixels and a GTX1080Ti GPU on a server platform, the DFF-Net achieved considerable performance on a real-world railway traffic dataset, with a mean average precision (mAP) of 90.12% and 54 (frames per second) FPS, which were 1.44% higher and 7 FPS faster than those of the SSD, respectively. For the small objects, i.e., the helmet and spanner, the mAP of the proposed method was 4.61% and 3.29% higher, respectively, than that of the SSD.

(3) The experimental results indicated that the DFF-Net effectively balances the real-time performance and accuracy. With an input size of 320 $\times$ 320 pixels, the forward inference speed was five times higher than that of Faster R-CNN, according to the railway benchmark. The DFF-Net can also be used for detecting objects in other scenarios.

II. Related Works

Studies on obstacle avoidance systems are important for ensuring train safety in rail lines. The Anti-Collision Device Network (ACDN) [13], which was produced by the Indian Railway Co., Ltd., locates a train via a global positioning system and identifies the trains in front of this train through radio. García *et al.* [14] proposed a multisensory system that can be fixed on the opposite sides of the railway line. The system can notify the monitoring system of detected obstacles. McCall [3] proposed an intelligent security system to detect and track moving objects in level crossing environments. Sinha and Feroz [5] utilized an accelerometer to measure obstacles on a railway track. Šilar and Dobrovolný [15] and Pu *et al.* [16] used matching vision to detect railway-crossing objects. The aforementioned systems must be placed at a fixed position and therefore are not suitable for detecting obstacles anywhere along the railway track. Detection systems installed in front of the trains based on vison-based approaches have been proposed. Nakasone *et al.* [17] detected obstacles using a monocular camera combined with image processing. Kaleli and Akgul [18] proposed a vison-based algorithm using dynamic programing to identify railway tracks. Nassu and Ukai [19] presented an approach to perform rail extraction via edge feature matching. Qi *et al.* [2] utilized the histogram of the oriented gradient features to detect railway tracks and recognize turnouts. The aforementioned methods are based on image processing or hand-draft engineered features, which require considerable prior knowledge and engineering skills for designing the appropriate algorithm. Moreover, it is difficult to design a unified method for the detection of diverse objects in extreme scenarios. However, CNNs can adaptively extract image features; therefore, the manual designing of features is not required. Ye *et al.* [1] and Li *et al.* [8] proposed two real-time railway object-detection algorithms based on CNNs. The aforementioned methods have room for improvement with regard to the object-detection accuracy. This accuracy is important for railway object detection because it is related to the driving safety. In this study, we developed an object-detection system that can be mounted on a train. We used the previously proposed CNN architecture DFF-Net to automatically detect obstacles on railway tracks. The DFF-Net considers both accuracy and real-time performance.

In recent years, approaches based on CNNs have achieved success in the field of object detection [20]. Detectors based on CNNs can generally be divided into two categories: the one-stage method and the two-stage method. The two-stage method, i.e., the Fast RCNN [21], Faster RCNN [11], and SSPnet [22], first produces a sparse of default object boxes and then determines the accurate location and category of objects using additional CNNs. These two-stage approaches have achieved remarkable performances and several benchmarks. However, they produce many proposals in the first step, leading to computational burden in the second stage. Therefore, to improve the detection efficiency, scholars have focused on the one-stage method. The OverFeat [23] approach uses a ConveNet to extract features in a sliding window on a set of pyramid images. This network can be trained end-to-end, starting from pixels and ending with object classification.

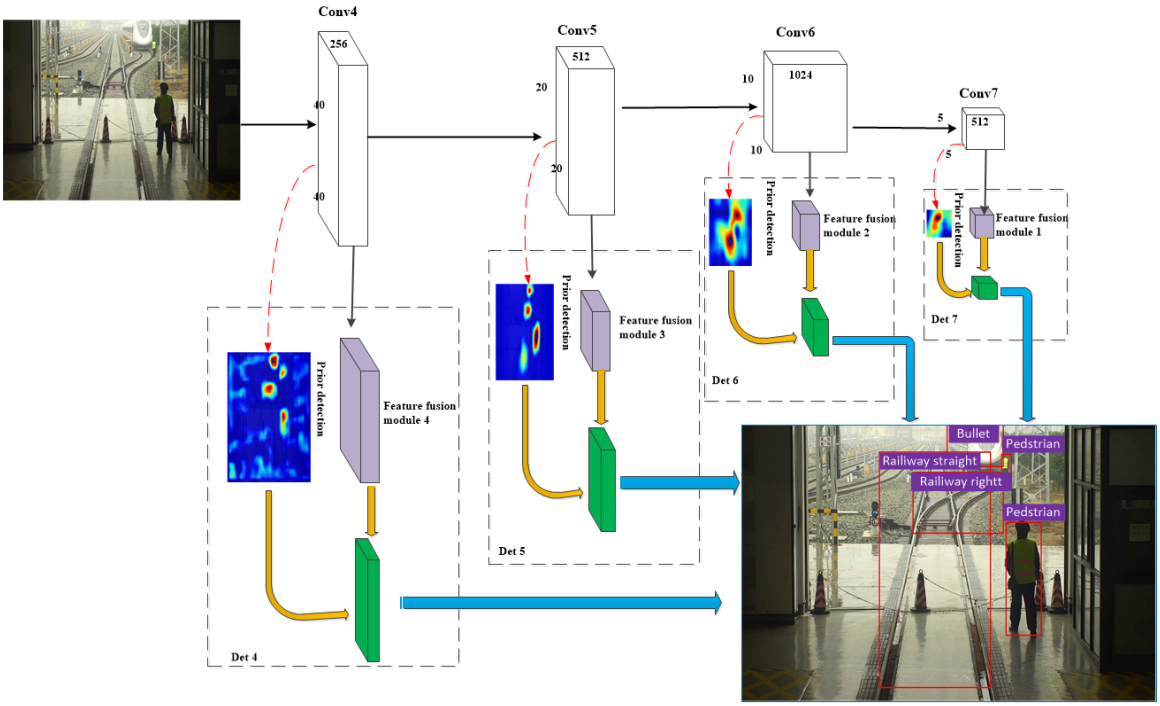


Fig. 2. Overview of the DFF-Net architecture, which includes a prior detection module and four object-detection modules. The object-detection modules consist of a differential set of feature fusion sub-modules.

SSD [12] and YOLO [24] are classical one-stage methods that directly use a single feedforward convolutional network to determine the locations and classes of objects. The main advantage of the one-stage method is speed; its detection accuracy is relatively low. In particular, YOLO is extremely efficient, even though it has a low detection accuracy. It is believed that one of the major reasons for the low accuracy is the class-imbalance problem. To improve the detection accuracy and address the imbalance problem, the DSSD [25] introduces additional context to the SSD by fusing feature maps. To significantly reduce the search space for the convolution stage, Kong *et al.* [26] used objectness priori to conduct negative sample mining. In this study, we designed a prior object-detection module to generate initial anchor boxes similar to RON [26] and employed differential convolution operations for different feature maps from the prior detection module to balance the computational complexity and the detection accuracy. Thus, the proposed method can ensure high accuracy and real-time performance.

III. Network Architecture

In this study, the DFF-Net was used for detecting real-world traffic objects on railways. For practical application, we focus on the accuracy and real-time performance of the DFF-Net. The network architecture of the DFF-Net is shown in Figure 2. The default input size of the network is 320 × 320 × 320. It consists of two modules: the prior detection module and the object-detection module. The prior detection module was constructed using VGG-16, which we pre-trained on ImageNet. The prior detection module is responsible for providing proper initialization, i.e., the initial locations and the size of prior boxes for the object-detection module, and it can reduce the search space for classification in the subsequent modules, whereas the object-detection module aims at regressing accurate object detections and predicting the class of the prior boxes. The process followed by the prior detection module is similar to that of the SSD [12]; it generates a fixed number of anchor boxes and predicts whether the classification scores of the objects belong to the objects. The object-detection module consists of a sub-module called the feature fusion module, which produces the scores for the object classes and location offsets corresponding to the prior anchor boxes. Subsequently, the final result is produced by non-maximum suppression (NMS). The procedures followed by the prior detection and the object detection are described in Sections 3.1 and 3.2, respectively.

A. Prior Detection Module

A two-stage detector, e.g., the faster R-CNN, has high accuracy because of the mechanism it follows, in which effective anchors are selected and initial locations for the anchor boxes are generated. However, one-stage methods, such as SSD and YOLO [23], predict the locations and classes of objects via one-step regression, which are less accurate in extreme scenarios, particularly for the detection of small objects. To improve the accuracy of the one-stage method, we designed a prior detection module to regress the locations and sizes of the object using two-step cascade regression.

Similar to the Faster R-CNN, we associate n anchor boxes corresponding to the cells of the feature map. The cells are regularly distributed in the feature maps, and each prior anchor box has a fixed initial location relative to its corresponding cell. Each cell in the feature map has n prior anchor boxes. We design a strategy in which the specific

feature-map locations are associated with a specific scale of anchors. A large feature map is responsible for small-scale object detection, whereas a small feature map corresponds to large-scale object detection. In this study, we select three aspect ratios (length–width ratios of the anchors): 0.5, 1.0, and 2.0. For DFF-Net320, the feature layers of conv4_3 (size of 40 × 40), conv5_3 (size of 20 × 20), conv6_2 (size of 10 × 10), and fc7 (size of 5 × 5) are considered to be basic feature maps to implement object detection. Each cell of the feature map corresponds to three default anchor boxes, and there are 6375 anchor boxes in total. The first regression is utilized to estimate the four initial coordinates of the prior boxes, and the second regression is responsible for determining whether there are objects in these anchor boxes. To handle the class-imbalance problem, we devise a mechanism to filter several well-classified negative prior anchors. Only the negative anchors with confidence of <0.99 pass to the next detection module for training. The confidence of the anchors in the prior detection module is the output of the cross-entropy. The proposed method produces prior anchor boxes with multi-scales as well as a conventional SSD, in contrast to the Faster R-CNN, which uses a region proposal network. In this study, feature layers of conv4_3 (size of 40 × 40 with 256 channels), conv5_3 (size of 20 × 20 with 512 channels), conv6_2 (size of 10 × 10 with 1024 channels), and fc7 (size of 5 × 5 with 512 channels) are considered to be basic feature maps for implementing object detection. The prior detection module provides prior location information for more accurate detection in the next module. Moreover, this narrows the search space, improving the object-detection accuracy.

B. Object-Detection Module

Scholars [26], [27] have employed highly integrated information to improve the small-object detection performance. As shown in Figure 2, the object-detection module and the prior detection module share features. The object-detection module consists of a differential set of feature fusion sub-modules, as shown in Figure 3. Incorporation of context by enlarging the window around the candidate proposal anchor boxes is performed in two-stage detection methods. DFF-Net simulates this strategy through simple convolution layers. Considering the size and computational complexity of different feature maps, we adopt different convolution operations for different feature maps from the prior detection module. The differential fusion features can provide sufficient feature information and allow efficient detection of objects. Figure 3 shows the differential fusion feature mechanism, which is integrated into the detection modules. We use filters with sizes of 7 × 7, 5 × 5, and 3 × 3 in the object-detection module, where conv 3 _ 3-s1, 256 indicates that the size of the filter is 3 _ 3, the number of filters is 256, and the step of the convolution is 1. To reduce the number of parameters, we adopt sequential 3 × 3 filters to replace the larger convolution filters, as in [28]. Three 3 × 3 filters replace a 7 × 7 filter, and two 3 × 3 filters replace a 5 × 5 filter.

The feature maps corresponding to the receptive fields and the kernel sizes of the feature fusion filter groups are presented in Table I.

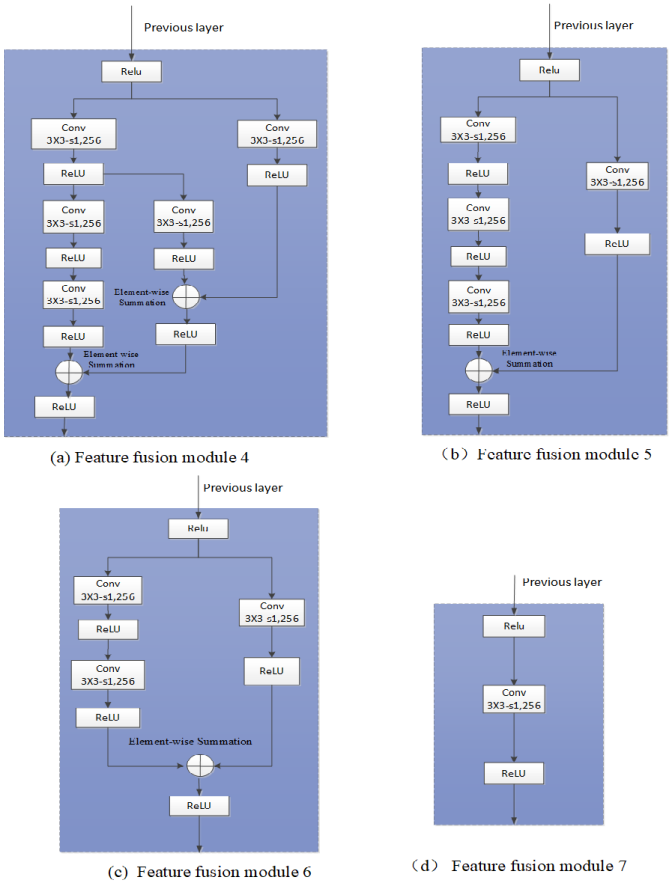


Fig. 3. Object-detection module overview (four differential feature fusion sub-modules).

TABLE I
FEATURE MAPS CORRESPONDING TO RECEPTIVE FIELD AND KERNEL SIZE OF FEATURE FUSION FILTER GROUPS

Size of feature map	Receptive field	Kernel size of feature fusion filter group
40 × 40	8 × 8	7 × 7,5 × 5,3 × 3
20 × 20	16 × 16	7 × 7,3 × 3
10 × 10	32 × 32	5 × 5,3 × 3
5 × 5	64 × 64	3 × 3

Taking a feature map with a size of 40 × 40 as an example, whose corresponding receptive fields are 8 × 8, we utilize 7 × 7, 5 × 5, and 3 × 3 filter groups to obtain the fusion feature for enhancing the detection of smaller objects. According to the feature fusion filter groups presented in Table I, we can obtain the fusion feature corresponding to the feature maps with sizes of 20 × 20, 10 × 10, and 5 × 5 in a similar manner.

Assuming that $\{F_i, i \in C\}$ represents the source feature maps produced by the series of convolutional layers, where $C = \{C_1, C_2, C_3, C_4\} = \{\text{conv4\_3, conv5\_3, conv6\_2, fc7}\}$ in this study, we can obtain the following:

$$F_{ki} = \phi_k(F_i), \quad k \in \{3, 5, 7\}, \tag{1}$$

where i represents a convolution layer from C, and ϕ_k denotes the convolutional operation with the size of filters. For feature maps with sizes of 40×40, 20×20, 10×10, 5×5, we can obtain the following equations:

$$\Phi_{40} = \phi_f(F_{7i}, F_{5i}, F_{3i}), \quad i = C_1 \tag{2}$$

$$\Phi_{20} = \phi_f(F_{7i}, F_{3i}), \quad i = C_2 \tag{3}$$

$$\Phi_{10} = \phi_f(F_{5i}, F_{3i}), \quad i = C_3 \tag{4}$$

$$\Phi_5 = F_{3i}, \quad i = C_5, \tag{5}$$

where ϕ_f represents the feature fusion function, and Φ_{40}, Φ_{20}, and Φ_{10} represent the fusion results for the feature maps, which have sizes of 40×40, 20×20, and 10×10, respectively. ϕ_f corresponds to the element-wise summation operation in the Caffe [1] platform. The feature map with a size of 5×5 does not conduct feature fusion, as it has sufficient feature information to detect large objects.

On the basis of Eqs. (1)–(5), the detection equation can be expressed as follows:

$$loc, class = D(\phi_d(\Phi_{40}), \phi_d(\Phi_{20}), \phi_d(\Phi_{10}), \phi_d(\Phi_5)), \tag{6}$$

where D predicts the final object detection results from the previous feature maps, and ϕ_d refers to the transformation function to generate multi-scale feature maps for object detection. Therefore, ϕ_d calculates the SoftMax probabilities per class and regresses the four offsets of the bounding box for each anchor, and the operation D includes the concatenation operation in the Caffe platform and NMS to eliminate redundancy.

The object-detection module calculates c (e.g., $c = 7$ for the railway traffic dataset) classification scores and four accurate offsets of objects corresponding to the initial anchor boxes obtained by the prior object-detection module, yielding $c + 4$ outputs (including c confidence scores and four location offsets) for each prior box to fulfill the detection process. Similar to the Faster R-CNN, the sub-network outputs the scores of each class. Subsequently, we predict the offsets relative to the prior anchor boxes using bounding-box regression.

IV. Experiments and Results

We present the railway traffic datasets in Section 4.1. The multi-task loss function for training the DFF-Net is presented in Section 4.2. Several experiments were conducted on railway traffic datasets to confirm the superiority of the proposed method. We compared the DFF-Net with the classical one-stage methods (SSD and DSSD), and the two-stage methods Faster R-CNN and RON to evaluate the performance of the proposed approach. We conducted all the experiments on the Caffe platform and examined the effects of different structural designs of the DFF-Net in an ablation study. The DFF-Net presented below represents DFF-Net320 without special instructions.

A. Datasets

We placed the equipment, which was developed by our team, in front of a train to capture real-world railway traffic videos, as shown in Figure 4. Additionally, we transplanted the proposed method into the equipment to detect the objects. We annotated all the samples according to human visual habits. Examples of the annotations are presented in Figures 5(a) and (b), which include Bullet Train, Pedestrian, Railway Straight, Railway Left, Railway Right, Helmet, and Spanner. To ensure diversity in the data, we acquired videos under different lighting and weather conditions. Subsequently, we sampled images every five frames in a series of railway traffic videos. We collected 7342 sample images with a resolution of 640×512 pixels. Considering the rail shape and the possible objects for trains in the shunting mode, we labeled the sample images according to seven categories: Bullet Train, Pedestrian, Railway Straight, Railway Right, Railway Left, Helmet, and Spanner. We labeled Railway Straight, Railway Right, and Railway Left at the center of the field of view associated with normal human vision. To investigate the detection effect of the DFF-Net on targets of different sizes, we calculated the mean size and mean area ratio for each category in the railway traffic dataset, as shown in Table II. The mean area ratio of Helmet and Spanner were 0.31% and 0.32%, respectively, indicating that the helmet and spanner were typical small objects. Row 2 of Table II indicates the number of each class in the dataset. We shuffled 70% of these images for training and validation, and the remainder of the images were used for testing.

Fig. 4. Railway traffic video collection system.

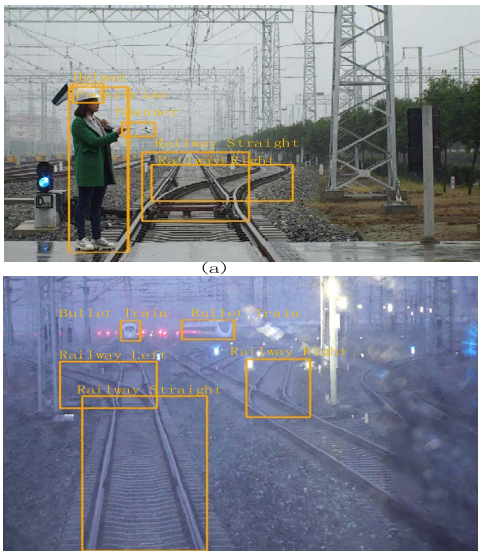


Fig. 5. Annotation examples.

TABLE II
MEAN SIZE AND MEAN AREA RATIO OF EACH CATEGORY IN RAILWAY TRAFFIC DATASETS

Category	Bullet Train	Pedestrian	Railway Straight	Railway Right	Railway Left	Helmet	Spanner
Occurrences (number)	3671	9371	3863	652	1804	3089	761
Mean size (pixels)	248 × 248	171 × 171	535 × 535	259 × 259	237 × 237	34 × 34	35 × 35
Mean area ratio	3.75%	2.23%	18.9%	4.38%	3.61%	0.31%	0.32%

TABLE III
HYPERPARAMETERS OF TRAINING PROCESS

Hyperparameter	Value
Input size	320 pixels × 320 pixels
Batch size	32
Stochastic gradient descent	0.9 momentum with 0.0005 weight decay
Initial learning rate	0.001
Iterations	20000

B. Training

To enlarge the railway traffic datasets and build a robust model, we adopted several data augmentation strategies as SSD [12]. The hyperparameters of the training process are presented in Table III. More details can be found in [12].

The loss function for the DFF-Net consists of two parts: the prior detection module and the object-detection module. We can formulate the loss function as follows:

$$\begin{aligned} &L(\{p_i\},\{q_i\},\{d_i\},\{t_i\}) \\ &= \frac{1}{N_{obj}}\left(\sum\nolimits_i L_{obj}(p_i,l_i)+\sum\nolimits_i I(l_i\geq 1)L_r(q_i,g_i)\right) \\ &\quad+\frac{1}{N_{\text{det}}}\left(\sum\nolimits_i L_{\text{det}}(d_i,l_i)+\sum\nolimits_i I(l_i\geq 1)L_r(t_i,g_i)\right), \quad (7) \end{aligned}$$

where i represents the i^{th} anchor in a minibatch, l_i represents the ground-truth category label of the i^{th} anchor, and g_i represents the ground-truth location of the i^{th} anchor. p_i and q_i represent the probabilities of the i^{th} anchor being an object and the initial locations of the anchor in the prior detection module, respectively. d_i and t_i represent the predicted object category and the location offsets of the initial anchor boxes in the object-detection module, respectively. N_{obj} and N_{det} represent the corresponding numbers of positive anchors in the prior detection module and the object-detection module, respectively. The object judging loss function L_{obj} is the cross-entropy loss over two classes (being an object or not), and the classification loss function L_{det} is the softmax loss over the confidence scores of multiple classes. L_r represents the regression loss of the bounding box. Similar to [11], [12], we applied a log-space shift in the box dimensions and allowed scale-invariant translation to parameterize the regression. Furthermore, we used the smooth regression loss l_1 as L_r. $I(\cdot)$ is an indicator function that limits the regression loss to the positive distribution anchors; i.e., $l_i \geq 1$ indicates that the regression loss of the negative anchors is ignored.

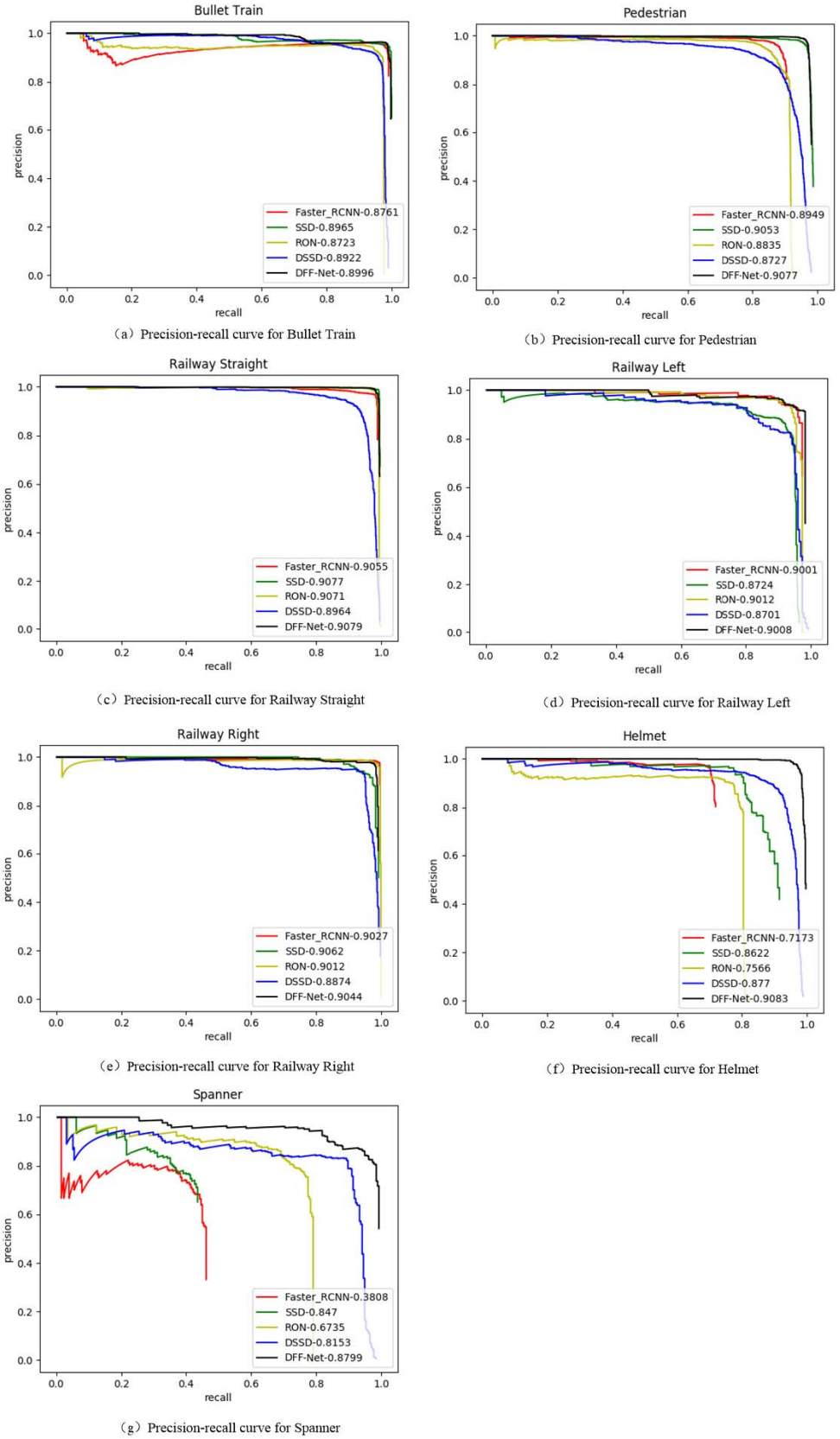
C. Comparison With State-of-the-Art

In our experiments, we considered VGG16 [11] as the backbone network of the DFF-Net. VGG16 is pre-trained on the ILSVRC-LOC dataset [29]. The average precision (AP) was utilized to evaluate the performance of the models, which was computed according to the area of a curve consisting of the precision and recall rates. To detect multi-class objects, we measured the model performance using the mAP.

Because the DFF-Net inherits the advantages of the SSD—Faster R-CNN and RON—and incorporates the concept of the DSSD with feature fusion, we compared the DFF-Net with the following classical object detectors: the SSD, Faster R-CNN, RON, and the DSSD. With the exception of the DSSD, the experiments on the other four detectors were conducted with the VGG16 backbone on a Caffe platform. For the DSSD, the backbone was ResNet-101 [30], and the procedure was taken from [25]. For fair comparisons, the maximum number of iterations of the six detectors was set as 200000. The comparison results are presented in Table IV. The DFF-Net achieved an mAP of 0.9012 with an input size of 320 × 320, which was far better than those of the other four classical detection methods. The experimental results for Faster R-CNN were the poorest among the methods tested. We deemed that Faster R-CNN may be more suitable for detecting objects with large images [1, 10]. We collected railway images with a size of 640 × 512. However, the input size of the regular Faster R-CNN is 1000 × 600, and the interpolation operation may lead to a loss of image information. The DSSD exhibited acceptable performance with the largest model size, and the SSD exhibited the second-best performance with regard to accuracy and speed. Overall, the DFF-Net was superior to the other approaches with regard to both accuracy and real-time performance.

Figure 6 shows the precision–recall curves of the five methods for the railway traffic datasets. The precision–recall curves for the typical railway traffic objects, i.e., Bullet Train, Pedestrian, Railway Straight, Railway Left, Railway Right, Helmet, and Spanner, are presented in Figures 6(a)–(g), respectively. DFF-Net was superior to the other four approaches and achieved the highest AP value among the seven categories. For small objects such as the helmet and spanner, the proposed method achieved AP values of 90.83% and 87.99%, respectively. The results indicate that the detection performance of the DFF-Net for small targets was acceptable. Moreover, the experimental results clearly indicate that Faster R-CNN performed the poorest for small-object detection.

The fifth column of Table IV presents the inference time of the DFF-Net and the other four methods. The inference times of these methods were calculated using a GeForce GTX1080Ti GPU for fair comparison. The proposed DFF-Net achieved 54 FPS with an input resolution of 320 × 320 pixels on a



(a) Precision-recall curve for Bullet Train

(b) Precision-recall curve for Pedestrian

(c) Precision-recall curve for Railway Straight

(d) Precision-recall curve for Railway Left

(e) Precision-recall curve for Railway Right

(f) Precision-recall curve for Helmet

(g) Precision-recall curve for Spanner

Fig. 6. Precision–recall curves of different methods for different railway traffic objects. The red line indicates the results of Faster R-CNN. The green line indicates the results of the SSD method. The blue line indicates the results of the DSSD. The black line indicates the results of the DFF-Net.

single GTX1080Ti GPU, and the SSD achieved 47 FPS with 300×300 pixels. The proposed method was five times faster than Faster R-CNN and approximately four times faster than RON when the prior detection module was used. Moreover, the DFF-Net was approximately four times faster than the DSSD with feature fusion design.

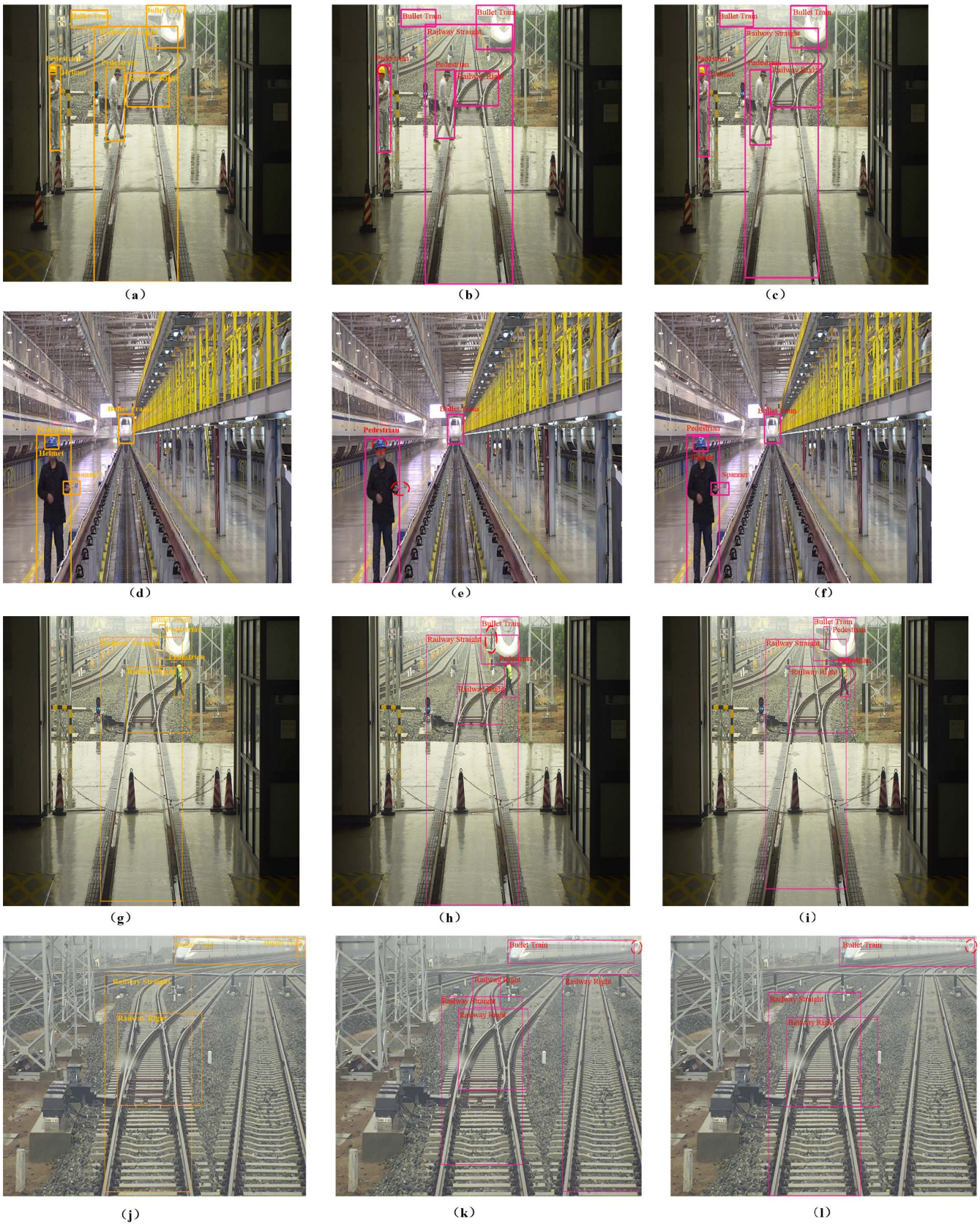


Fig. 7. DFF-Net vs. SSD300. We trained both methods using the railway traffic dataset that was previously described. The left column shows the original detection images, and the yellow rectangles indicate the ground truth of the objects. The middle column shows the detection results for SSD300, and the right column shows the detection results for the proposed DFF-Net. In rows 1–4, we present four detection scenarios for the railway traffic objects in the experiment: outside train garage, train garage, pedestrian, and running train, respectively. Bounding boxes with a classification score of $\geq$0.8 were drawn. The red dotted line indicates an undetected object.

TABLE IV

RESULTS OF COMPARISON WITH CLASSICAL DETECTORS FOR RAILWAY TRAFFIC DATASETS

Method	Backbone	Input size	#Boxes	FPS	Model size (M)	mAP
SSD	VGG-16	~ 300 × 300	8732	47	98.6	0.8868
Faster R-CNN	VGG-16	~ 1000 × 600	300	10	521	0.7968
RON	VGG-16	~ 320 × 320	21250	15	161.5	0.8417
DSSD	ResNet-101	~ 321 × 321	17,080	13	623.4	0.873
DFF-Net-320	VGG-16	~ 320 × 320	6375	54	109.5	0. 9012

TABLE V

MODELS WITH VARIOUS DESIGNS

Component	SSD	DFF-Net/o	DFF-Net
Prior object-detection module?	-	-	√
Feature fusion sub-modules?	-	√	√

TABLE VI

PERFORMANCE WITH DIVERSITY DESIGNS (ALL MODELS WERE EVALUATED USING RAILWAY TRAFFIC DATASETS)

Method	mAP	FPS	Bullet Train	Pedestrian	Railway Straight	Railway Left	Railway Right	Helmet	Spanner
SSD	0.8868	47	0.8965	0.9053	0.9077	0.8724	**0.9062**	0.8622	0.847
DFF-Net/O	0.8901	53	0.8972	0.9055	0.9063	0.8936	0.9039	0.8647	0.8592
DFF-Net	0.9012	54	**0.8996**	**0.9077**	**0.9079**	**0.9008**	0.9044	**0.9083**	**0.8799**

TABLE VII

PERFORMANCE OF VARIOUS FEATURE FUSION DESIGNS (ALL DESIGNED MODELS WERE EVALUATED USING RAILWAY TRAFFIC DATASETS; BOLD FONT INDICATES BEST RESULT)

Method	mAP	FPS	Bullet Train	Pedestrian	Railway Straight	Railway Left	Railway Right	Helmet	Spanner
DFF-Net-full	**0.9019**	45	0.8968	**0.9079**	**0.908**	0.9003	0.904	0.9083	**0.8881**
DFF-Net-reduce	0.9009	50	0.8995	0.9077	0.9079	0.8994	**0.9051**	**0.9084**	0.8781
DFF-Net	0.9012	**54**	**0.8996**	0.9077	0.9079	0.9008	**0.9044**	**0.9083**	0.8799

TABLE VIII

PERFORMANCE COMPARISONS FOR DIFFERENT INPUT SIZES BASED ON RAILWAY TRAFFIC DATASETS (BOLD FONT INDICATES BEST RESULT)

Method	mAP	FPS	Bullet Train	Pedestrian	Railway Straight	Railway Left	Railway Right	Helmet	Spanner
DFF-Net-512	0.9045	16	**0.9057**	**0.9086**	0.9079	**0.9038**	0.8999	0.9081	**0.8979**
DFF-Net-320	0.9012	54	0.8996	0.9077	0.9079	0.9008	**0.9044**	**0.9083**	0.8799

D. Visual Results for Railway Traffic Object Detection

We present the visual detection results for railway traffic objects in this section. We set the classification threshold score as 0.8 to evaluate the detection performance of the DFF-Net; i.e., bounding boxes were drawn for objects with classification scores of $\geqslant$0.8. We re-edited the class of the object manually and ignored the classification scores of each object for a better appearance. Moreover, we compared the proposed method with the SSD300 approach. In Figures 7(a)–(l), the left column contains the original detection images, and the yellow rectangles indicate the ground truth of the objects. The middle column presents the detection results for SSD300, and the right column corresponds to the proposed DFF-Net. As shown in Figures 7(b), (e), (h), and (k), SSD300 incurred detection failure. As shown in Figures 7(b) and (e), the helmets and a spanner were not detected. As shown in Figures 7(h) and (k), a pedestrian and a train were not detected. The proposed DFF-Net detected most of the objects well, except for a train, as shown in Figure 7(l). The experimental results indicate that the proposed method is superior to SSD300 for small-object detection. DFF-Net first employs the prior detection module to produce the initial anchor boxes; then, the object-detection module based on differential feature fusion sub-modules inputs the initial anchor boxes for further detection, resulting in more accurate detection. The differential feature fusion sub-modules enriched the sematic information for object detection, which improved the performance, particularly for small-object detection. However, the object-detection performance of DFF-Net has considerable room for improvement.

E. Robustness Test

As described in this section, we evaluated the robustness of the DFF-Net in different scenarios. The yellow boxes denote the ground truth of the multiple objects, and the pink boxes represent the objects detected by SSD300 and the proposed method. Figures 8(a)–(f) show the robustness-test results. Figure 8(a) indicates that the DFF-Net can detect obstacles on the railway, as well as pedestrians walking across the railway. For Figures 8(b) and (e), the train was driving in poor weather, and the acquired images were low-quality. However, DFF-Net exhibited high scores for detecting straight and curved railways. The proposed method can accurately detect the railway turning intersection ahead of time to inform the train driver to drive cautiously, as shown in Figure 8(c). As shown in Figure 8(e), the DFF-Net can check the railway working conditions in the tunnel well and ensure safe driving conditions in the case of low illumination. Figure 8(f) indicates that the proposed method can detect multiple objects on the railway with good accuracy, particularly small objects, i.e., the spanner and helmet. The experimental results indicate that the DFF-Net can achieve high detection performance even with low-quality images. Furthermore, the robustness-test results indicate that the DFF-Net can satisfy the requirements of real-world railway object detection in the shunting mode.

F. Ablation Study

1) Comparison of Different Designs:

a) Designs of prior detection module and feature fusion sub-module: For a more intuitive understanding of the effects of the prior detection module and the feature fusion

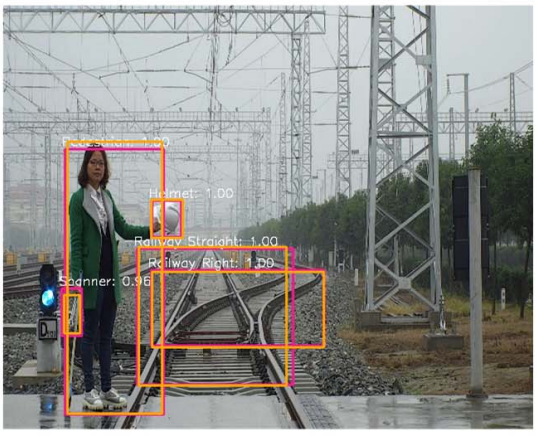
(a) Workers in railway tracks

(b) Railway in the night

(c) Driving in the fork

(d) Driving in tunnel

(e) Railway in the rain



(f) Multiple objects in the tracks

Fig. 8. Robustness test in different scenarios.

sub-module, we introduced DFF-Net/o, i.e., the DFF-Net without the prior detection module. Compared with the DFF-Net, the SSD was designed without the prior detection and feature fusion modules. Table V presents the designs of the three methods, and Table VI presents the performance of the three methods. In the experiment, the DFF-Net achieved

TABLE IX

PASCAL VOC2007 TEST DETECTION RESULTS (BOTH FAST RCNN AND FASTER RCNN WERE TESTED USING INPUT IMAGES WHOSE MINIMUM DIMENSION WAS 600; TWO SSD MODELS WITH INPUT SIZES OF 300 × 300 PIXELS AND 512 × 512 PIXELS WITH SAME SETTINGS WERE TESTED)

Method	mAP (%)	Airplane	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV
Fast [23]	70	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster [11]	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300[27]	77.5	79.5	83.9	76.0	69.6	50.5	87.0	85.7	88.1	60.3	81.5	**77.0**	86.1	**87.5**	83.97	79.4	52.3	77.9	**79.5**	**87.6**	76.8
SSD512[27]	79.5	84.8	85.1	81.5	73.0	57.8	**87.8**	88.3	87.4	**63.5**	**85.4**	73.2	86.2	86.7	83.9	82.5	**55.6**	**81.7**	79.0	86.6	**80.0**
RON320[28]	74.2	75.7	79.4	74.8	66.1	53.2	83.7	83.6	85.8	55.8	79.5	69.5	84.5	81.7	83.1	76.1	49.2	73.8	75.2	80.3	72.5
DFF-Net320	78.3	82.4	83.5	77.1	71.1	58.6	86.6	87.9	88.5	62.8	80.9	74.1	85	86.6	84.3	82.3	52.8	79	80.1	85.6	77.5
DFF-Net512	**79.89**	**86.0**	**86.39**	**81.53**	**74.17**	**65.85**	87.44	**88.39**	**88.74**	62.98	80.46	72.89	**86.25**	86.64	**84.99**	**83.96**	54.98	80.92	78.88	87.31	79.01

TABLE X

PASCAL VOC2012 TEST DETECTION RESULTS (BOTH FAST RCNN AND FASTER RCNN WERE TESTED USING INPUT IMAGES WHOSE MINIMUM DIMENSION WAS 600; TWO SSD MODELS WITH INPUT SIZES OF 300 × 300 PIXELS AND 512 × 512 PIXELS WITH SAME SETTINGS WERE TESTED)

Method	mAP(%)	Airplane	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Mbike	Person	Plant	Sheep	Sofa	Train	TV
Fast [23]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster[11]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
SSD300[27]	75.8	88.1	82.9	74.4	61.9	47.6	82.7	78.8	91.5	58.1	80.0	64.1	89.4	85.7	85.5	82.6	50.2	79.8	**73.6**	86.6	72.1
SSD512[27]	78.5	90.0	85.3	**77.7**	64.3	58.5	85.1	**84.3**	**92.6**	61.3	**83.4**	65.1	89.9	**88.5**	88.2	**85.5**	54.4	**82.4**	70.7	87.1	75.6
RON320[28]	71.7	84.1	78.1	71.0	56.8	46.9	79.0	74.7	87.5	52.5	75.9	60.2	84.8	79.9	82.9	78.6	47.0	75.7	66.9	82.6	68.4
DFF-Net320	76.2	90.3	80.6	76.2	62.2	52.1	83.6	77.9	90.1	59.4	81.6	62.0	**90.3**	84.5	85.0	84.5	51.9	80.6	69.6	**90.1**	71.7
DFF-Net512	**79.0**	**90.7**	**86.5**	77.3	**65.4**	**59.6**	**85.9**	84.1	90.9	**63.4**	82.4	**65.6**	90.1	88.2	**88.5**	85.0	56.1	81.8	73.5	88.7	**76.4**

the best results with regard to both accuracy and real-time performance. The SSD exhibited the poorest results among the three methods without the prior detection and the feature fusion modules. Without the prior detection module, DFF-Net/o exhibited poorer performance than the DFF-Net, but it exhibited better performance than the SSD with the feature fusion module. Without the prior detection module, DFF-Net/o produced more anchor boxes for object detection, which resulted in a lower FPS for detecting objects compared with the DFF-Net.

b) Various feature fusion designs: To investigate the use of differential feature fusion, we introduced DFF-Net-full and DFF-Net-reduce. "DFF-Net-full" indicates that feature modules 4–7 are all adopted to obtain fusion feature maps, as well as Figure 3(a), and "DFF-Net-reduce" indicates that feature modules 4–7 are all adopted to obtain fusion feature maps, as well as Figure 3(b). The results of a comparison of the three feature fusion designs are presented in Table VII. DFF-Net-full achieved the highest mAP and the lowest FPS among the three types of feature fusion methods. Clearly, DFF-Net-full is more complex than the other two approaches. However, the mAP value for the DFF-Net was approximately 0.7% lower than that of DFF-Net-full and 0.3% larger than that of DFF-Net-reduce. The DFF-Net achieved a higher FPS than the other two designs. The experimental results indicate that the DFF-Net achieves a good compromise between accuracy and real-time performance.

2) Comparison of Different Designs: As shown in Table VIII, the detection performance was significantly influenced by the input size. In this experiment, the original image was resized from 640 × 512 to 320 × 320 or 512 × 512 for input to the network. The mAP value of DFF-Net-512 was higher than that of DFF-Net-320 for detecting railway traffic objects. High-resolution input images can enlarge the information of the small objects, making the objects easier to detect. Although increasing the input size can improve the detection performance for small objects, it reduces the real-time performance of the algorithm. DFF-Net-512 achieved 16 FPS, and DFF-Net achieved 54 FPS. DFF-Net achieved a good balance between accuracy and real-time performance.

Experiments Using PSCAL VOC2007 and VOC2012: To verify the universal property of the proposed method, we trained our model on VOC2007 trainval and VOC2012 trainval and then compared it with other models, such as Fast-RCNN [21], Faster R-CNN [11], SSD300 [25], SSD512 [25], and RON320 [28]. Owing to our limited amount of time and computation resources, we adopted the VGG-16 backbone rather than utilizing more effective networks, e.g., the ResNet [31] backbone. All of the methods were fine-tuned on the same pre-trained VGG16 network. The goal of VOC is to recognize objects from many visual object classes (i.e., not pre-segmented objects) in real scenarios. It is fundamentally a supervised learning problem because it provides a set of labeled images as training sets. Twenty object categories were selected: person, animal (bird, cat, cow, dog, horse, sheep), vehicle (airplane, bicycle, boat, bus, car, motorbike, train), and indoor (bottle, chair, dining table, potted plant, sofa, and television (TV)/monitor). We set the hyper parameters as described in Section 4.2. For all experiments, 40000 training iterations were conducted. The comparison results for VOC2007 are

presented in Table IX. As shown, the DFF-Net achieved an mAP of 0.7989, which was the highest value among the models tested. In the table, the best results are presented in bold font. According to the experimental results and the results for VOC2007, the proposed method achieved better performance under the test conditions compared with the classical models (for the same datasets), particularly for small-object (i.e., bottle) detection. The results indicate that the proposed method has a good adaptability to different datasets and is of great significance.

Furthermore, the comparison results for VOC2012 are presented in Table X. For the VOC2012 task, we followed the settings of VOC2007, with the following changes. For the proposed model, we trained the first 60000 iterations with a learning rate of 10^{-3} and trained the next 20000 iterations with a learning rate of 10^{-4}.

The results in Table X exhibit the same performance trend observed for the VOC2007 test. The experimental results indicate that among the methods tested, the proposed method had the best performance for this dataset.

V. Conclusion

We proposed a real-time railway traffic object-detection system that is based on DFF-Net for preventing railway shunting accidents. To enhance both the accuracy and the real-time performance, we introduced two novel parts in the DFF-Net: a prior object-detection module and an object-detection module. First, the prior object-detection module generates prior anchor boxes for the next module. By using the prior anchor boxes as input, the object-detection module applies a feature fusion sub-module to enrich the sematic information, resulting in accurate object detection. Several experiments were performed using railway traffic datasets. The DFF-Net achieved an mAP of 90.12% with 54 FPS on a computer with an Nvidia GTX1080Ti GPU. Compared with the state-of-the-art SSD algorithm, the algorithm of the proposed method achieved an mAP 1.44% higher and an FPS 7 higher. In particular, for small objects, i.e., a helmet and spanner, the mAP of the DFF-Net was 4.61% and 3.29% higher, respectively, than that of the SSD, indicating that the DFF-Net is more suitable for small-object detection. The proposed DFF-Net achieves a good tradeoff between the accuracy and real-time performance. Moreover, experiments involving railway traffic datasets and PASCAL VOC2007 and VOC2012 indicated that the DFF-Net is more efficient for small-object detection than the other classical methods. In the future, we will consider compressing and optimizing the DFF-Net and transplanting the model into an embedded system, i.e., Nvidia Jeston TX2. Furthermore, we plan to extend the application of the proposed method to other scenes.

Acknowledgment

The authors would like to thank China University of Mining and Technology, Beijing, and the Beijing Institute of Remote Sensing Equipment for providing the experimental hardware platform. They would also like to thank the Railway Bureau for providing us with a field to collect experimental data. Moreover, we thank Editage (www.editage.cn) for English language editing.

References

[1] T. Ye, B. Wang, P. Song, and J. Li, "Automatic railway traffic object detection system using feature fusion refine neural network under shunting mode," *Sensors*, vol. 18, no. 6, p. 1916, 2018.

[2] Z. Qi, Y. Tian, and Y. Shi, "Efficient railway tracks detection and turnouts recognition method using HOG features," *Neural Comput. Appl.*, vol. 23, no. 1, pp. 245–254, Jul. 2013.

[3] J. Mccall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[4] T. Zhu and J. M. M. S. De Pedro, "Railway traffic conflict detection via a state transition prediction approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1268–1278, May 2017.

[5] D. Sinha and F. Feroz, "Obstacle detection on railway tracks using vibration sensors and signal filtering using Bayesian analysis," *IEEE Sensors J.*, vol. 16, no. 3, pp. 642–649, Feb. 2016.

[6] H. Salmane, L. Khoudour, and Y. Ruichek, "A video-analysis-based railway–road safety system for detecting hazard situations at level crossings=," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 596–609, Apr. 2015.

[7] R. Danescu and S. Nedevschi, "Probabilistic lane tracking in difficult road scenarios using stereovision," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 272–282, Jun. 2009.

[8] J. Li, F. Zhou, and T. Ye, "Real-world railway traffic detection based on faster better network," *IEEE Access*, vol. 6, pp. 68730–68739, 2018.

[9] Y. Lecun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[10] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[12] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, Oct. 2016, pp. 21–37.

[13] *Anti Collision Device Network [EB/OL]*. Accessed: Jun. 9, 2018. [Online]. Available: http://en.wikipedia.org/wiki/Anti_Collision_Device

[14] J. J. Garcia *et al.*, "Efficient multisensory barrier for obstacle detection on railways," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 702–713, Sep. 2010.

[15] Z. Šilar and M. Dobrovolný, "The obstacle detection on the railway crossing based on optical flow and clustering," in *Proc. 36th Int. Conf. Telecommun. Signal Process. (TSP)*, 2013, pp. 755–759.

[16] Y.-R. Pu, L.-W. Chen, and S.-H. Lee, "Study of moving obstacle detection at railway crossing by machine vision," *Inf. Technol. J.*, vol. 13, no. 16, pp. 2611–2618, Dec. 2014.

[17] R. Nakasone *et al.*, "Frontal obstacle detection using background subtraction and frame registration," *Q. Rep. RTRI*, vol. 58, no. 4, pp. 298–302, 2017.

[18] F. Kaleli and Y. S. Akgul, "Vision-based railroad track extraction using dynamic programming," in *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2009, pp. 1–6.

[19] B. T. Nassu and M. Ukai, "Rail extraction for driver support in railways," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 83–88.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[21] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*. [Online]. Available: https://arxiv.org/abs/1312.6229

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[25] C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: https://arxiv.org/abs/1701.06659

[26] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5936–5944.

[27] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[28] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2117–2125.

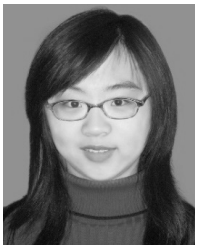
[29] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[31] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," 2016, *arXiv:1603.08029*. [Online]. Available: https://arxiv.org/abs/1603.08029

Tao Ye received the B.S. degree in measurement and control technology and instrumentation from the China University of Mining and Technology, Xuzhou, China, in 2009, the M.S. degree in mechanical and electronic engineering from the China University of Mining and Technology–Beijing, Beijing, China, in 2012, and the Ph.D. degree in measurement technology and instruments from the Key Laboratory of Precision Opto-mechatronics Technology of Ministry of Education, Beihang University, Beijing, in 2016. He was an Engineer with the Beijing Institute of Remote Sensing and Equipment from 2016 to March 2019. He is currently a Senior Engineer with the School of Mechanical Electronical and Information Engineering, China University of Mining and Technology–Beijing. His current research interests include deep learning and traffic detection.

Xi Zhang received the B.S. degree in mechanical design and manufacture from the Hefei Industry University, Anhui, China, in 1989, and the M.S. degree in hydraulic transmission and control and the Ph.D. degree in mine mechanical engineering from the China University of Mining and Technology–Beijing, Beijing, China, in 1991 and 1995, respectively, and the Ph.D. degree in mine mechanical engineering from Beijing Technology University, Beijing, in 1997. He has been with the School of Mechanical Electronic and Information Engineering, China University of Mining and Technology–Beijing, since 1997. As a Professor, his main research interests include mining machines, hydraulic transmission and control, measurement technology and instruments, machine learning, and object detection.



Yi Zhang received the B.S. degree in mechanical engineering and automation and the M.S. degree in mechanical manufacturing and automation from the China University of Mining and Technology–Beijing, Beijing, in 2009 and 2012, respectively, where she is currently pursuing the Ph.D. degree. She has been an Engineer with the School of Mechanical Electronic and Information Engineering, China University of Mining and Technology–Beijing since 2012. Her current research interests include deep learning and traffic detection.

Jie Liu received the B.S. degree in mechanical engineering and automation from the China University of Mining and Technology, Xuzhou, China, in 2008, and the M.S. degree in mechanical and electronic engineering from the China University of Mining and Technology–Beijing, Beijing, China, in 2011, where she is currently pursuing the Ph.D. degree. Her current research interests including hydraulic sealing, mining machines, and object detection.