IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxix 00, 0000, date of current version xxxix 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.Doi Number

Autonomous Railway Traffic Object Detection Using Feature-Enhanced Single-Shot Detector

Tao Ye[1], Zhihao Zhang[1], Xi Zhang[1], Fuqiang Zhou[2]

[1] School of Mechanical Electronic & Information Engineering, China University of Mining & Technology (Beijing), Beijing 100083, China

[2] School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100083, China

Corresponding author: Tao Ye (e-mail: ayetao198715@163.com).

ABSTRACT With the high growth rates of railway transportation, it is extremely important to detect railway obstacles ahead of the train to ensure safety. Manual and traditional feature-extraction methods have been utilized in this scenario. There are also deep learning-based railway object detection approaches. However, in the case of a complex railway scene, these object detection approaches are either inefficient or have insufficient accuracy, particularly for small objects. To address this issue, we propose a feature-enhanced single-shot detector (FE-SSD). The proposed method inherits a prior detection module of RON [1] and a feature transfer block of FB-Net [2]. It also employs a novel receptive field-enhancement module. Through the integration of these three modules, the feature discrimination and robustness are significantly enhanced. Experimental results for a railway traffic dataset built by our team indicated that the proposed approach is superior to other SSD-derived models, particularly for small-object detection, while achieving real-time performance close to that of the SSD. The proposed method achieved a mean average precision of 0.895 and a frame rate of 38 frames per second on a railway traffic dataset with an input size of 320 × 320 pixels. The experimental results indicate that the proposed method can be used for real-world railway object detection.

INDEX TERMS feature transfer block, prior detection module, railway object detection, receptive field-enhancement module.

I. INTRODUCTION

In recent years, intelligent transportation systems (ITSs) have developed rapidly. The computer-vision perception unit is the key component of an ITS. In particular, in the field of automobile driving, computer vision is indispensable for identifying traffic signs, pedestrians, and vehicles [3–6]. Timely identification of obstacles on the road can reduce the frequency of accidents. Compared with road traffic accidents, train crashes can cause large-scale catastrophes resulting in considerable casualties and property losses [7, 8]. Therefore, it is necessary to construct an ITS for trains and detect obstacles in front of trains in time. In this study, we focus on detecting railway objects in a sequence of railway images for auxiliary driving of trains, providing timely warnings of existing threats, and preventing disasters. Our assistant driving system is mainly aimed at object detection in the shunting mode of the train (entry and exit station) with a speed of <45 km/h.

In this study, we designed an automatic railway object detection system using a graphics processing unit (GPU) and convolution neural networks [9, 10]. A prototype of the designed system is shown in Fig. 1. The system was installed in a train cap. As shown in Fig. 1, a camera acquires images for the proposed railway object detection method. A near-infrared laser is mainly used to supplement the illumination of the camera when the light is insufficient. When our system detects obstacles on the railway track in front of the train, a voice alerts the driver of possible collisions. According to the Railway Object Dataset [2] built by our team, the proposed method can detect seven different types of objects: bullet train, railway straight, railway left, railway right, pedestrian, helmet, and spanner. The purpose of detecting the type of railway track (i.e., railway straight, railway left, or railway right) is 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. Generally, the line of sight of the train driver is easily blocked in the case of a curved railway. The proposed system can remind the driver to operate the train safely when it detects the curved railway. The aim of detecting other types of objects is to allow the train drivers to find obstacles in front of the train in sufficient time to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

prevent collisions. In particular, when the proposed system detects pedestrians or trains on the railway ahead, the train attendants are informed by the voice prompt and take appropriate measures to prevent a collision. The helmet and spanner are two items that maintenance workers often lose on the railway. The motivation for detecting them is to reduce unnecessary losses.

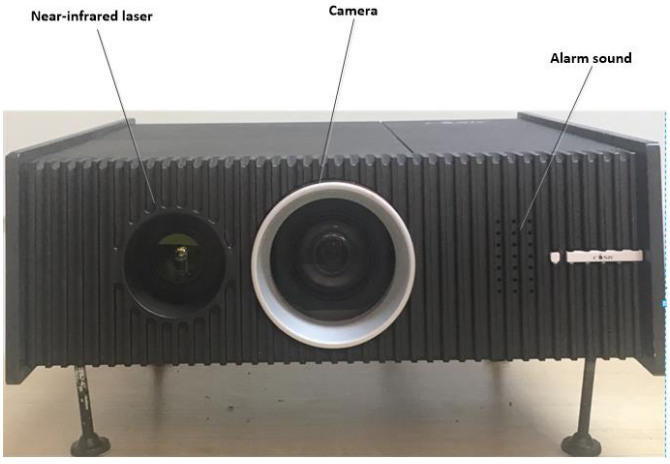


FIGURE 1. Prototype of the railway object detection system.

In this study, we focused on designing a railway object detection algorithm. We adopt a structure similar to that of the single-shot detector (SSD) [11] to ensure real-time object detection performance. The SSD can detect objects of different sizes by using multi-scale feature maps in real time. However, its shallow feature map does not contain high-level semantic information, leading to an inadequate discriminative ability for detecting small objects. This disadvantage makes the SSD unsuitable for detecting multi-scale railway objects. To achieve a high accuracy and real-time railway object detection, we exploit the advantages of RON [1] and FB-Net [2], enhance the receptive field, and propose a receptive field-enhancement neural network (FE-SSD). Inspired by the prior module detection mechanism of RON, the FE-SSD produces the initial anchor boxes to improve the accuracy of object detection. Similar to FB-Net, adjacent feature maps are fused to learn deeper semantic information by utilizing a feature transfer block (FTB). Finally, the FE-SSD achieves improved performance compared with the SSD [11] by enhancing the feature maps for detecting objects, particularly for small-object detection.

The main contributions of this work can be summarized as follows. First, we proposed the FE-SSD, which is a deep-learning-based railway traffic object detection method. The proposed method inherits the merits of the RON and FB-Net methods and involves the application of an enhanced receptive field, which can balance the accuracy and real-time performance for object detection and significantly improve the small-object detection performance. Third, with a resolution of 320 × 320 pixels and an NVIDIA GTX1080Ti GPU on a computer, the FE-SSD achieves a mean average precision (mAP) of 0.895 and a frame rate of 38 frames per second (FPS), which are superior to those of the other approaches mentioned in this paper.

This paper is organized as follows. We discuss previous research on railway object detection in Section 2. In Section 3, we present the detailed network of the FE-SSD. Section 4 presents the experimental results for the railway traffic dataset. Finally, we draw conclusions in Section 5.

II. Related Works

Researchers have developed a variety of approaches for detecting obstacles in railways. Takeda [12] proposed an infrared detection unit for indicating ray-interrupting railway objects. Researchers [13–15] have utilized various radar systems for detecting vehicles at the crossing boundaries. Kim [16] and Silar [17] proposed different types of optical devices to detect crossing railway obstacles. Oorni [18] proposed an alarm system using a global positioning system to detect railway objects. Zhang [19] utilized Programmable Logic Controller (PLC) to design an intelligent alarm system for warning train drivers of passing objects. With the development of computer capabilities, methods based on image processing have been used to detect railway objects. Compared with the foregoing methods, the methods based on image processing are more intuitive, less expensive, and more accurate. Kim [20] and Mockel [21] mounted video cameras on the front of trains to detect possible obstacles using the camera-motion estimation method. González [22] and Wei [23] used background subtraction techniques to detect objects in a railway environment. Pu [24] developed an advanced safety system based on machine vision to detect moving obstacles at a railway crossing. Teng et al. [25] used a super-pixel-based railway object detection algorithm together with a support vector machine to classify obstacles. The aforementioned methods, which are based on image processing or traditional machine-learning techniques, were evaluated on small datasets with various limitations, e.g., background with little variation, lighting with little fluctuation, and limited object size. Hence, these methods can hardly satisfy the requirements of real-world railway object detection. However, the success of deep learning led to breakthroughs in the field of object detection. Several object detection methods [1, 2, 9, 10, 26, 27] have been proposed in recent years. Relatively few applications of deep convolution neural networks to railway object detection have been reported. Guo [28] proposed a railway pedestrian detection approach using AlexNet combined with an Histogram of Oriented Gradient (HOG) feature, which achieved a high accuracy and real-time performance. However, the algorithm cannot achieve end-to-end training and detection, and the detection performance for multi-class and multi-scale objects must be further investigated. Ye [29] proposed an FR-Net to detect railway objects in the shunting mode. Although FR-Net can achieve a high accuracy and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

real-time performance, there is considerable room to improve the performance for the railway traffic dataset, particularly for small-object detection. Furthermore, Li [2] designed an FB-Net to achieve the appropriate speed/accuracy for actual railway object detection. However, the detection performance of FB-Net for small objects needs to be improved. In this study, we designed a railway object detection model based on SSD that inherits a prior detection module (PDM) of RON and an FTB of FB-Net. Additionally, it employs a novel receptive field-enhancement module (RFEM). The proposed method can significantly enhance the feature discriminability and robustness and therefore ensure good accuracy and real-time performance.

III. Framework of FE-SSD

The objective of the FE-SSD is to autonomously detect railway objects in front of trains. As shown in Fig. 2, the FE-SSD introduces four FTBs, four PDMs, and eight RFEMs into the original SSD. The FTBs are designed to merge the adjacent feature maps via a deconvolution operation, making the semantic information of the current layer more abundant. The element-wise summation operation is adopted to fuse the corresponding feature maps, as indicated by Fig. 2. We implement FTB in a similar manner to FB-Net [2]. The PDM is responsible for providing better initialization, i.e., the initial locations and the size of prior boxes for the object detection stage, and can reduce the search space for classification in the subsequent module. We adopt the same mechanism as RON [1] in the PDM. In this study, the RFEM was designed to enhance the receptive field of feature maps and construct robust features that can easily detect small objects. We first assembled four RFEMs between the convolution layer of the Visual Geometry Group Network (VGG) backbone and conv4_3, between conv4_3 and conv5_3, between conv5_3 and fc7, and between fc7 and conv6_2. Additionally, we inserted the other four RFEMs into the detection branches corresponding to the original feature maps of conv4_3, conv5_3, fc7, and conv6_2. At the end of this section, the training strategies are briefly introduced.

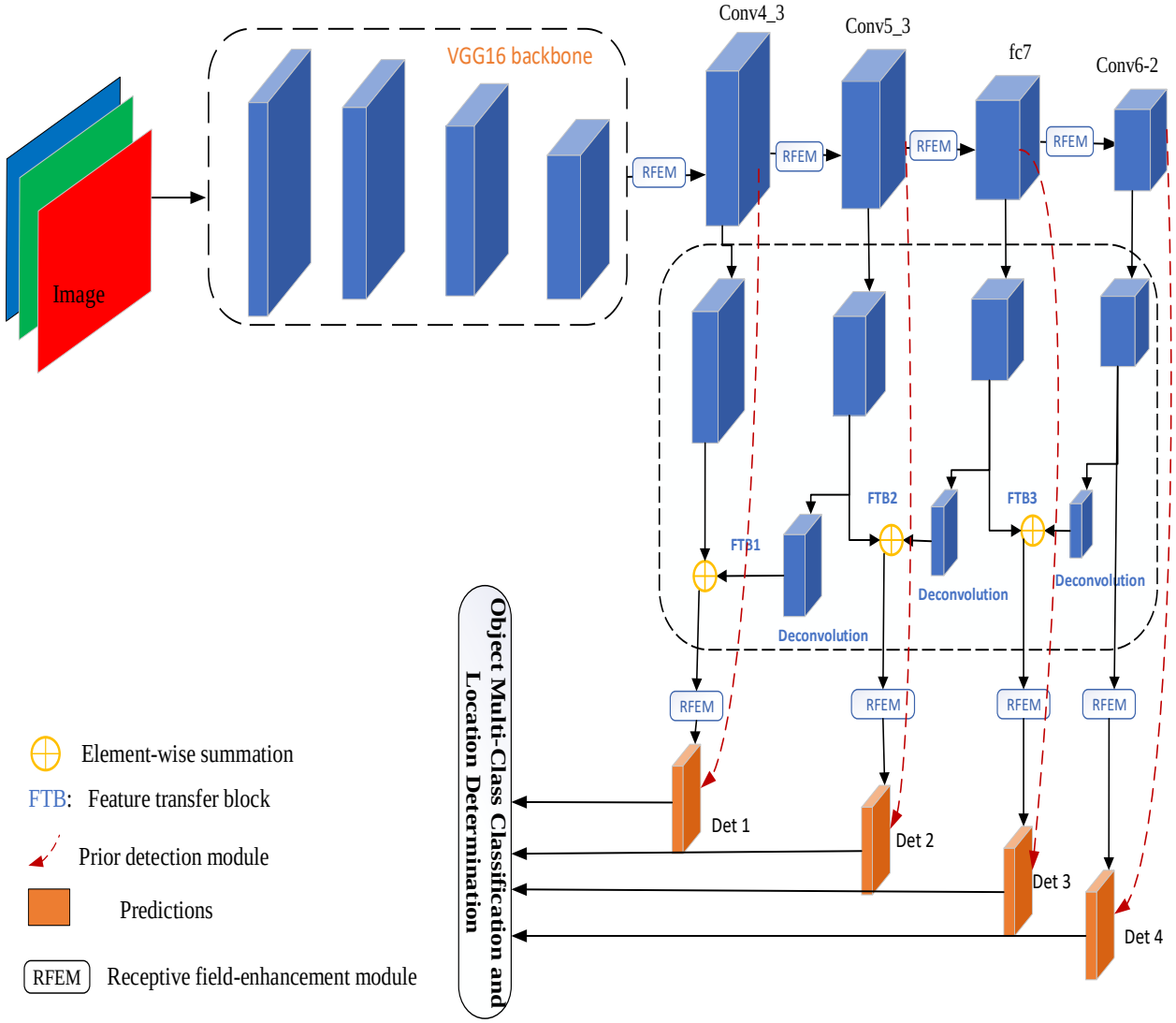


FIGURE 2. Architecture of the FE-SSD. The input size of the FE-SSD is 320 × 320 pixels.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

A. FTBS

Inspired by the feature integrating mechanism of FB-Net, we constructed the FTBs, as shown in Fig. 3. The FTBs transfer the information of the former adjacent feature maps into the current layer and enrich the semantic information of the current layers, where $3 \times 3 - s1$ indicates that the size of the kernel filter is 3×3, the number of filters is 256, and the step of the convolution is 1. We utilize the deconvolution operation to ensure that the adjacent feature maps have the same dimensions, and we adopt the elementwise summation operation of Caffe [30] to merge the corresponding two feature maps together. Then, the RFEM is used to enhance the discriminability of the features.

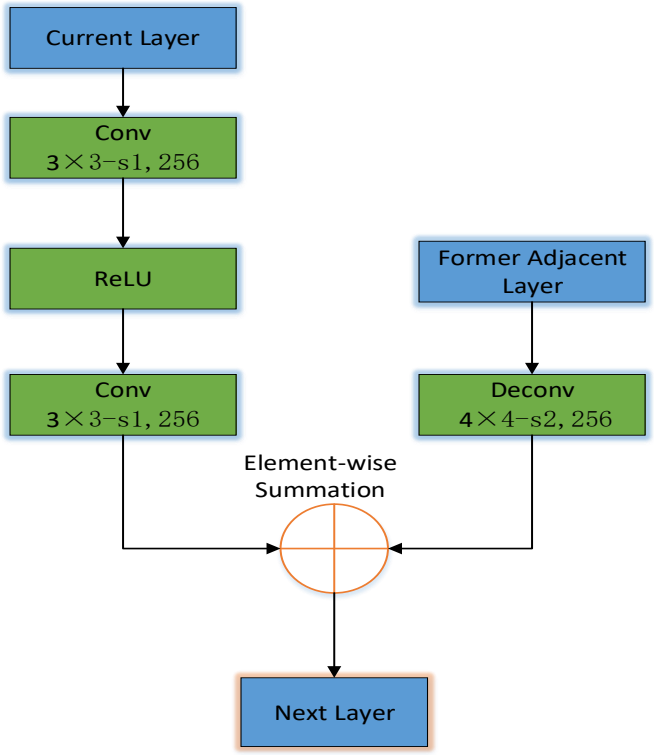


FIGURE 3. Structure of a FTB.

B. PDM

To improve the accuracy of the proposed method, we adopt a PDM to regress the locations and classes of the object using two-step cascade regression. Similar to RON [1], we associate each cell of the feature map with n anchor boxes. Each cell is 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 to associate a specific map location with a specific scale of anchors. In this study, we selected three aspect ratios: 0.5, 1.0, and 2.0. Each cell of the feature map corresponds to three default anchor boxes, and there are 6375 anchor boxes in total. The first regression is used to estimate and predict the four initial coordinates of the prior boxes, and the second is responsible for determining whether these anchor boxes contain objects. To solve the class imbalance problem, we design a mechanism to filter many well-classified negative prior anchors. Only the negative anchors with confidence of <0.99 pass to the next module for training. The proposed method produces prior anchor boxes with multiple scales, as well as the conventional SSD, in contrast to Faster-RCNN, which uses the region proposal network. As shown in Fig. 2, feature layers of conv4_3 (size of 40 × 40 with 256 channels), conv5_3 (size of 20 × 20 with 512 channels), fc7 (size of 10 × 10 with 1024 channels), and conv6_2 (size of 5 × 5 with 512 channels) are considered as basic feature maps for object detection. The PDM provides prior location information and is combined with the RFEM for more accurate detection in the next module. Moreover, it narrows the search space for better object detection.

C. RFEM

As shown in Fig. 2, the feature layers of conv4_3, conv5_3, fc7, and conv6_2 are taken as base feature maps to conduct object detection. In the present study, these four feature layers correspond to four feature maps with sizes of 40×40, 20×20, 10×10, and 5×5, respectively. The upper four RFEMs are embedded between the four aforementioned feature layers (conv4_3, conv5_3, fc7, and conv6_2) to enlarge the receptive field of the input features and provide richer features for follow-up object detection. Inserting the four bottom RFEMs into the corresponding FTBs can further enhance the performance of the features, particularly with regard to the robustness of the features, facilitating the detection of small objects. Fig. 3 presents our design for the RFEM. As shown, the RFEM is a multi-branch convolution layer. These convolution branches have a multi-scale design and can enlarge receptive fields of various sizes. The RFEM has five branches.

As shown in Fig. 4, we abbreviate the batchnorm [31], scale, and rectified linear unit (ReLU) nonlinearity as B-S-R. The input size of the previous layer is $C \times H \times W$, where C represents the channel of the previous layer, and H and W represent the height and width, respectively, of the previous layers. A 1×1 convolution operation is used to reduce the number of channels of the first four branches to one-fourth of the previous layer.

For the first branches, we use a max pooling operation followed by a 1×1 convolution and BSR. The output of the first branch is denoted as b_1. For the second and third branches, we replace a 3×3 convolution layer with two parallel 1×3 and 3×1 convolution layers to reduce the number of parameters and obtain deeper nonlinear layers. We define the outputs of the second and third branches as b_2 and b_3, respectively. For the fourth branch, we utilize two stacked 3×3 convolution layers to replace a 5×5 convolution layer. As demonstrated above, we replace a 3×3 convolution layer with two parallel 1×3 and 3×1 convolution layers to reduce the number of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

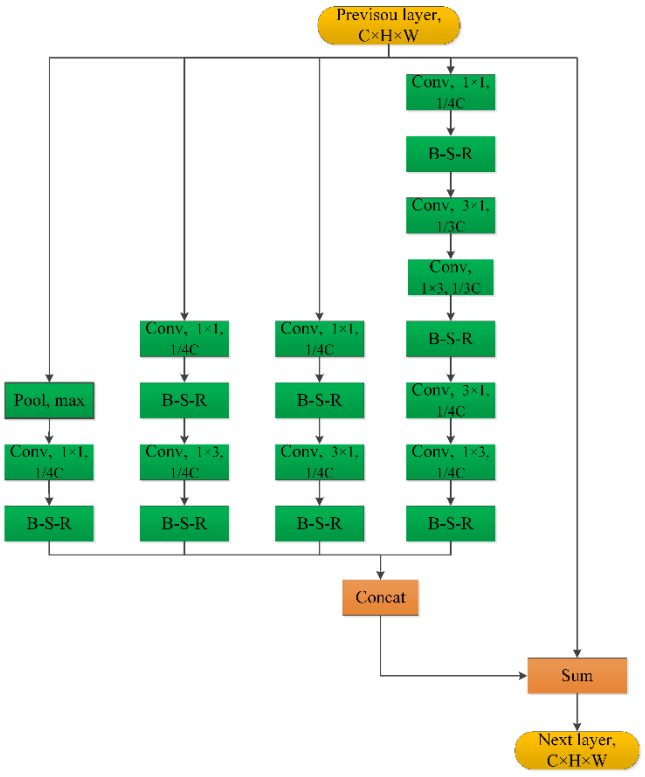


FIGURE 4. Structure of the RFEM. B-S-R represents the batchnorm, scale, and ReLU operations.

parameters. These substitution operations can improve the real-time performance of the network. We denote the output of the fourth branch as b_4. Finally, we apply the shortcut operation using ResNet [32]. The output feature map can be obtained as follows:

$$out = cat(b_1, b_2, b_3, b_4) + \alpha b_5, \quad (1)$$

where *cat* represents the concatenation operation in the Caffe framework [30], α is a scale factor that is usually set to a constant value of 0.5, and b_3 is used to retain the receptive field of the previous layer.

Hence, considering the PDM and the four bottom RFEMs, we can describe the transformation function as follows:

$$Det1 = \varphi_d(Pri_{40}, out_1) \quad (2)$$
$$Det2 = \varphi_d(Pri_{20}, out_2) \quad (3)$$
$$Det3 = \varphi_d(Pri_{10}, out_3) \quad (4)$$
$$Det4 = \varphi_d(Pri_5, out_4), \quad (5)$$

where φ_d represents the transformation function to generate multi-scale feature maps for object detection. Specifically, φ_d computes the SoftMax probabilities for each class and regresses the four offsets of the bounding box for each anchor. The outputs of φ_d are Det1 to Det4, as shown in Fig. 2. Pri_{40}, Pri_{20}, Pri_{10}, and Pri_5 correspond to the i^{th} original feature maps, i.e., conv4_3, conv5_3, fc7, and conv6_2. out_i represents the i^{th} output within the four bottom RFEMs, where $1 \le i \le 4$. The localization loss of FE-SSD is the Smooth L_1 Loss between the ground truth box (g) and predicted box (l), the calculation function can be described as follows:

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{\ p}\, smooth_{L1}\left(l_i^{\ m} - \hat{g}_j^{\ m}\right), \quad (6)$$

where N represents the number of positive samples, $x_{ij}^{\ p}$ represents the indicator when matching the i^{th} default boxes with the j^{th} ground-truth boxes of category p. Similar to SSD, we regress to offsets for the center (cx, cy) , width (w) and height (h) of the default bounding box (d), and $l_i^{\ m}$, $\hat{g}_j^{\ m}$ respectively represent the offsets of the predicted box and ground truth box, the calculation function of offsets can be described as follows:

$$\hat{g}_j^{cx} = \left(g_j^{cx} - d_i^{cx}\right)/d_i^{w}, \hat{g}_j^{cy} = \left(g_j^{cy} - d_i^{cy}\right)/d_i^{h},$$
$$\hat{g}_j^{w} = log\left(\frac{g_j^{w}}{d_i^{w}}\right), \hat{g}_j^{h} = log\left(\frac{g_j^{h}}{d_i^{h}}\right), \quad (7)$$

The confidence loss uses SoftMax multi-class loss, the calculation function can be described as follows:

$$L_{conf}(x, c) = \sum_{i \in Pos}^{N} x_{ij}^{\ p}\, log(\hat{c}_i^{\ p}) - \sum_{i \in Neg} log(\hat{c}_i^{\ 0})$$
$$where\ \hat{c}_i^{\ p} = \frac{exp(c_i^{\ p})}{\sum p\, exp(c_i^{\ p})}, \quad (8)$$

where $\hat{c}_i^{\ p}$ is the confidence prediction value of category p in the i-th prediction bounding box, obtained by the SoftMax function, and p = 0 refers to the background.

According to (2)–(5), the detection equation can be expressed as follows:

$$loc, cls = D(Det1, Det2, Det3, Det4), \quad (9)$$

where D represents the function to conduct final object detection using the receptive field-enhanced feature maps. Additionally, *loc* and *cls* represent the location and class, respectively, of bounding boxes. The final detection boxes from Det1 to Det4 are gathered using non-maximum suppression (NMS). The function D concatenates Det1 to Det4 by utilizing the concatenation operation in the Caffe platform and NMS to eliminate redundancy. Operation D calculates c (i.e., c = 7 for the railway traffic dataset of this study) classification scores and four accurate offsets of objects corresponding to the prior anchor boxes obtained by the PDM, yielding c+4 outputs (including c confidence scores and four location offsets) for each initial anchor box to implement the object detection process. The assembled RFEMs can improve the detection accuracy of the original network. Considering the tradeoff between accuracy and real-time performance, we performed experiments to demonstrate the superiority of our design, as described in Section 4.

D. TRAINING

We apply data augmentation strategies similar to those of SSD [11] and adopt the training skills reported in Ref. [11]. The FE-SSD is designed according to the framework of Caffe [30]. The proposed network is trained end-to-end and is optimized via stochastic gradient descent (SGD). We adopted SGD with momentum of 0.9 and weight decay of 0.0005 in this study. We set the initial learning rate to 0.001 for the first 100000 iterations and reduced the learning rate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

to 0.0001 for the next 100000 iterations. The batch size of the training samples was set as 32. We adopted the center code type to encode the bounding boxes and utilized the same hard mining strategy, negative anchor filtering method, and matching strategy as FB-Net. As shown in Fig. 2, VGG16 is taken as the backbone, which is pre-trained on ImageNet for initialization. We initialize the weights of all the new layers by utilizing a zero-mean Gaussian distribution with a standard deviation of 0.01. Additional details are presented in Ref. [11].

IV. EXPERIMENTS

In this section, we first introduce the railway traffic dataset in detail. Second, we compare the SSD, RON, FR-Net, FB-Net, and FE-SSD to evaluate their performance with regard to the overall accuracy. Then, extensive experiments to evaluate the effectiveness of the FE-SSD via an ablation analysis are discussed. In these experiments, we used the mAP as the evaluation metric. All five detectors used VGG-16 [32] as the backbone, for fair comparison. Because the foregoing detectors were all derived from the SSD, we set the maximum number of iterations of the five detectors to 200000 in this study.

A. Dataset

The real-word railway traffic dataset was identical to that used in Refs. [2, 29]. The data were acquired from various scenarios, including different weather and lighting conditions. The training images were sampled from the raw video sequence. However, we collected approximately 1400 images to expand the formal dataset in Refs. [2, 29] and obtain 8706 images with a total size of 640 × 480 pixels. Although many public datasets (VOC, COCO) are used in the field of object detection, there are no suitable public datasets for railway object detection. To better cater to the scenarios of railway object detection, we built the railway benchmark. Considering the railway shape and potential obstacles in the shunting mode, we labeled the images with seven classes: bullet train, pedestrian, railway straight, railway left, railway right, helmet, and spanner. The number of each class in the dataset is presented in Table I. Moreover, Fig. 5 presents the mean area ratio (MAR) of each class. The MAR was obtained by dividing the pixel size of the object by the resolution of the image. As shown in Fig. 3, the "railway straight" and "bullet train" categories exhibited the highest MARs: 18.34% and 13.13%, respectively. These two categories represent typical large objects in the railway dataset. The MAR values for "helmet" and "spanner" were 0.86% and 0.31%, respectively, indicating that these two categories were representative of typical small objects. Thus, 70% of these images were shuffled for training and validation, and the remaining images were used for testing.

B. EXPERIMENTS ON RAILWAY TRAFFIC OBJECT DATASET

1) COMPARISON WITH STATE-OF-THE-ART METHODS

Because the proposed method is an improved version of the SSD, we compared the FE-SSD with SSD variant algorithms (e.g., SSD, RON, FR-Net, and FB-Net) to evaluate its performance for the railway traffic dataset. The comparisons were performed on a Caffe platform. The experimental results are presented in Table II. With an input size of 320 × 320 pixels, the FE-SSD achieved an mAP of 0.895, which was higher than those of the other four models. Among the methods tested, RON exhibited the lowest accuracy. We consider that RON generated too many anchor boxes, which increased the degree of redundant matching, reducing the detection accuracy. FR-Net320 had an mAP of 0.8798. Compared with RON, FR-Net320 focuses more on the improvement of the real-time performance. Although FB-Net exhibited the fastest real-time performance, it had a lower accuracy than the FE-SSD; i.e., the AP values for the "spanner" category were 7.18% lower than those for the FE-SSD. Owing to the combination of the PDM, FTB, and RFEM, the FE-SSD exhibited high real-time performance and the highest detection accuracy among the five detectors.

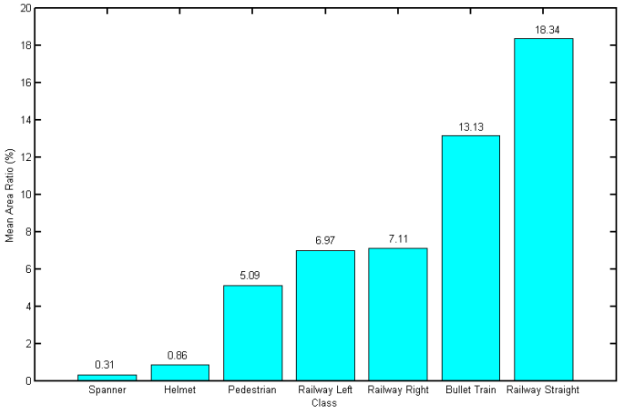


FIGURE 5. MAR for each class in the railway traffic dataset.

The fifth column in Table II presents the inference times of the FE-SSD and the other four detectors. The inference time was calculated with a single image on a machine with an NVIDIA 1080TI GPU. As shown in Table II, the FE-SSD processed a single image in 26.3 ms (FPS) with an input size of 320 × 320 pixels, whereas the SSD processed a single image in 21.3 ms with an input size of 300 × 300 pixels. Compared with the SSD and RON, the FE-SSD, FB-Net, and FR-Net generated fewer anchor boxes on the multi-scale feature maps, i.e., 6375 anchor boxes for the FE-SSD vs. 8732 anchor boxes for SSD300. RON produced a large number of anchor boxes; hence, it processed a single image in 66.7 ms (FPS). FR-Net processed a single image in 13.9 ms with an input size of 300 × 300 pixels, exhibiting the best real-time performance among the methods tested. However, its accuracy was lower than that of the FE-SSD. Because railway traffic object detection involves safety issues, the accuracy must be carefully considered while improving the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

real-time performance. The FE-SSD was slightly slower than SSD300, as shown in Table I. However, it achieved the highest accuracy among the tested models, as well as a good tradeoff between the speed and the accuracy.

The accuracy for each category is presented in Table III. The FE-SSD outperformed the other four detectors and achieved the highest AP value for five categories. For the helmet and spanner (typical small objects), the FE-SSD achieved AP values of 0.8986 and 0.8768, respectively. The results indicate that the proposed method is superior to the other approaches for small-object detection and significantly enhances the feature discriminability of small objects.

2) VISUALIZATION OF DETECTION RESULTS FOR RAILWAY TRAFFIC DATASET

Fig. 6 shows typical detection results for SSD300 and FE-SSD320 under different scenarios of the railway traffic dataset. To evaluate the performance of these two methods, we set the confidence threshold score to 0.6. Figs. 6(a)–(f) present the detection results for SSD300, and Figs. 6(g)–(l) present the results for FE-SSD. SSD300 exhibited detection failure. As shown in Figs. 6(a)–(f), some small objects and partially occluded objects were not well detected. Two distant trains in Figs. 6(a) and (c) were not detected. A distant pedestrian and two helmets were lost, as shown in Fig. 6(b). As shown in Figs. 6(d) and (f), three helmets were not detected. The spanner was not detected, as shown in Fig. 6(e). The non-detected objects were almost exclusively small objects. This is mainly because SSD300 does not make full use of the multi-scale information of the feature maps. Owing to the PDMs, FTB, and RFEMs, the proposed method can detect small objects in cases of extreme occlusions. As shown in Figs. 6(g)–(l), some small objects were not detected by SSD300, whereas this situation was avoided by the proposed method.

3) ROBUSTNESS TEST

We evaluated the robustness of the FE-SSD in different scenarios. Figs. 7(a)–(f) present the robustness test results of the proposed method. As shown in Figs. 7(a) and (c), the proposed method judged the working conditions well and ensured safe driving under low-illumination conditions. For Figs. 7(b) and (d), the acquired images were low-quality because of the poor weather and the motion smear. However, the FE-SSD exhibited high scores for detecting railways and multiple objects. As shown in Fig. 7(e), the proposed method accurately detected the railway turning intersection ahead of the train for informing the driver to drive carefully. Fig. 7(f) shows that the FE-SSD detected multiple railway objects with good accuracy—particularly small objects, such as the helmet and spanner. According to the robustness test results, the FE-SSD exhibited high detection performance even with low-quality images. The experimental results indicate that the FE-SSD can satisfy the requirements of real-world railway traffic object detection in the shunting mode.

C. ABLATION STUDY

1) COMPARISON OF DIFFERENT COMPONENTS

To evaluate the efficiency of the PDM, FTB, and RFEM components, we developed and tested three variant models. The tests were performed using the railway traffic dataset. We first developed the FE-SSD(-4) model, which does not have PDM, FTB, or RFEM components. Then, we developed FE-SSD(-3), which has a PDM but no FTB or RFEM. Third, we developed FE-SSD(-2), which combines a PDM and three FTBs. For FE-SSD(-1), there are four RFEs on the SSD at the top positions, as shown in Fig. 1, on the basis of FE-SSD(-2). The FE-SSD comprised a PDM, FTBs, and two RFEM components (four RFEM components on the SSD at top positions and four RFEMs on the SSD at bottom positions). Table IV presents the different designs, and Table V presents the performance for the different designs. We first validated the effects of the PDM and FTB components. As shown in Table V, the mAP of FE-SSD(-2) was increased to 0.893 (compared with 0.8718 for the SSD) with the introduction of the two types of components. For FE-SSD(-1), the mAP was increased to 0.8943 by introducing the top four RFEMs. With the addition of the PDM, FTB, and two RFEM components, the mAP of the FE-SSD increased to 0.895. As indicated by the fourth and fifth columns of Table

TABLE I
NUMBER OF EACH CLASS IN THE DATASET.

Class	Bullet Train	Pedestrian	Railway Straight	Railway Left	Railway Right	Helmet	Spanner	Total
Number	4453	10447	4797	1231	2823	3271	1119	28141

TABLE II
COMPARISON WITH CLASSIC DETECTORS FOR THE RAILWAY TRAFFIC DATASET. BOLD FONT INDICATES THE BEST RESULT.

Method	Backbone	Input size	#Boxes	Model size (M)	FPS	mAP
SSD300	VGG-16	~ 300 × 300	8732	99	47	0.8788
FR-Net320	VGG-16	~ 320 × 320	6375	75	72	0.8798
FB-Net320	VGG-16	~ 320 × 320	6375	54	82	0.871
RON320	VGG-16	~ 320 × 320	21250	162	15	0.8426
FE-SSD-320	VGG-16	~ 320 × 320	6375	141	38	**0. 895**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

TABLE III
COMPARISON WITH STATE-OF-ART METHODS FOR THE RAILWAY TRAFFIC DATASET: DETAILED RESULTS. BOLD FONT INDICATES THE BEST RESULT.

Method	mAP	Bullet train	Pedestrian	Railway straight	Railway left	Railway right	Helmet	Spanner
SSD	0.8788	0.8982	0.8711	0.8971	0.8811	0.9028	0.8723	0.829
FR-Net320	0.8798	0.9055	**0.8888**	0.9025	0.8708	**0.9057**	0.8663	0.8192
FB-Net320	0.871	0.9052	0.8783	0.8991	0.8519	0.9022	0.8556	0.8050
RON320	0.8426	0.8928	0.8703	0.8985	0.8628	0.903	0.7316	0.7398
FE-SSD320	**0.895**	**0.9056**	0.8862	**0.9029**	**0.8934**	0.9018	**0.8986**	**0.8768**

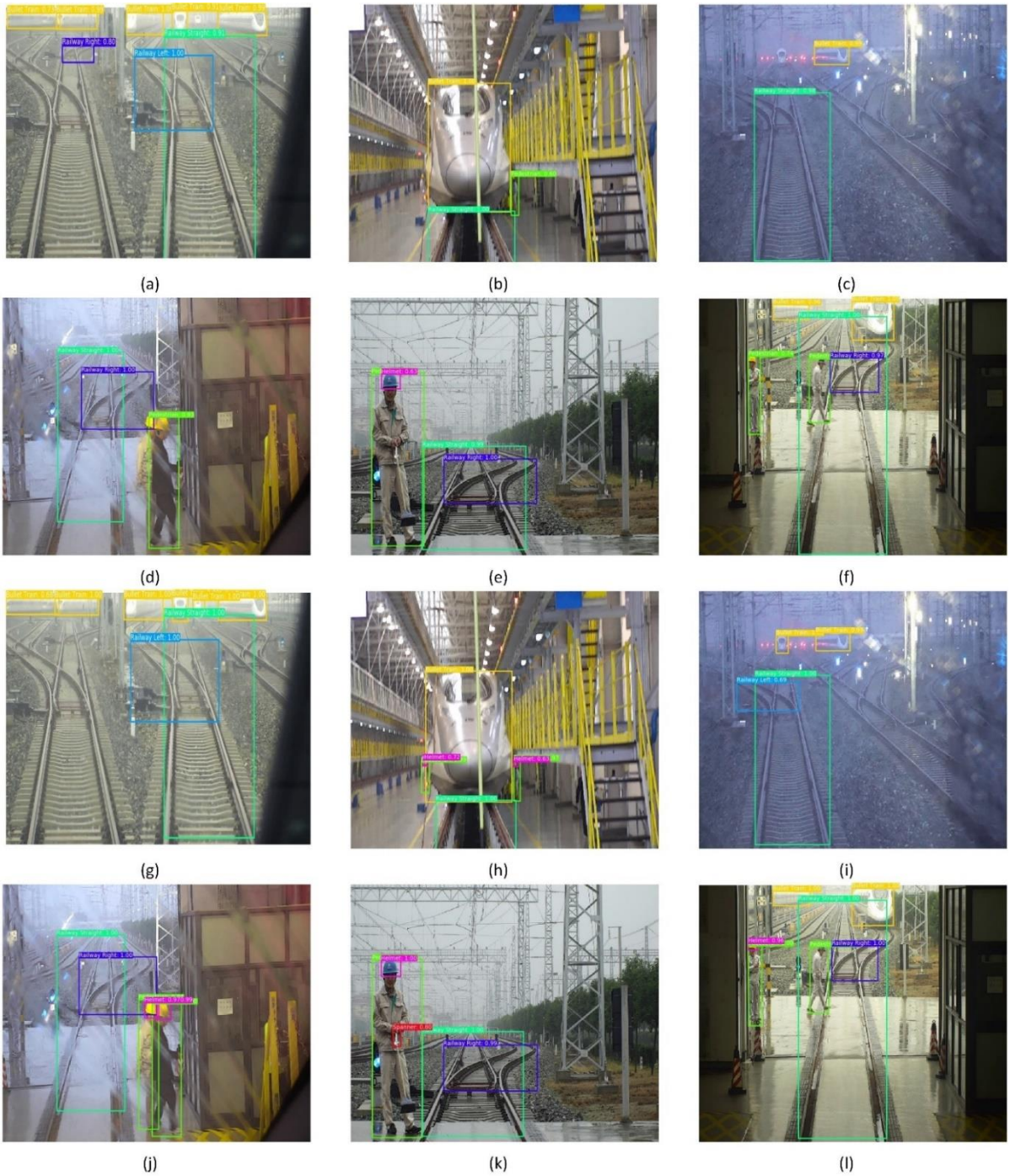
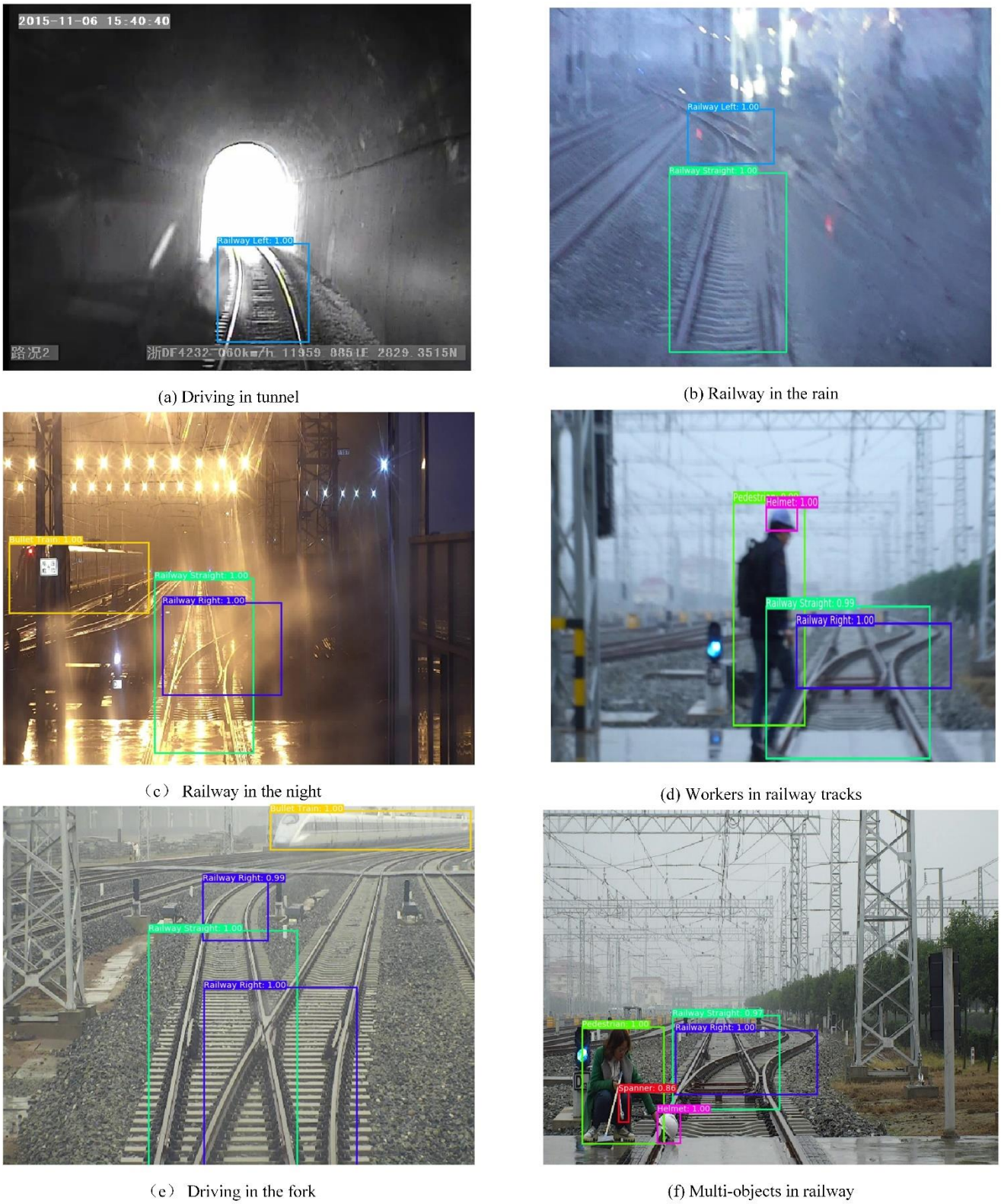


FIGURE 6. **Detection results for the SSD and FE-SSD in different scenarios. Each color corresponds to an object category.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access



(a) Driving in tunnel

(b) Railway in the rain

(c) Railway in the night

(d) Workers in railway tracks

(e) Driving in the fork

(f) Multi-objects in railway

FIGURE 7. **Robustness test in different scenarios.**

V, with the addition of the PDM, FTB, and RFEM components, the capacity of the models increased, while the real-time performance of the models decreased. However, the mAP value for the FE-SSD was approximately 2.3% higher than that of FE-SSD(-4). The test results indicate that the FE-SSD achieved higher accuracy than the SSD without FE-SSD/O with the backbone of MobileNet achieved the fastest detection but the lowest accuracy. When we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

TABLE IV
MODELS WITH VARIOUS DESIGNS.

Component	FE-SSD(-4)	FE-SSD(-3)	FE-SSD(-2)	FE-SSD(-1)	FE-SSD
Prior object detection module?		√	√	√	√
+3 FTB			√	√	√
+4 RFEM module(T)?				√	√
+4 RFEM module(B)?					√

TABLE V
PERFORMANCE WITH DIFFERENT COMPONENTS. ALL THE MODELS WERE EVALUATED USING THE RAILWAY TRAFFIC DATASET. BOLD FONT INDICATES THE BEST RESULT.

Method	mAP	FPS	Model size (M)	Bullet train	Pedestrian	Railway straight	Railway left	Railway right	Helmet	Spanner
FE-SSD(-4)	0.8718	52	92	0.8928	0.8598	0.8954	0.8806	0.8994	0.8498	0.8248
FE-SSD(-3)	0.8922	48	132	**0.9057**	0.8848	0.9033	0.8904	0.9017	0.8953	0.8641
FE-SSD(-2)	0.893	45	136	**0.9057**	0.886	**0.9043**	0.8907	**0.9038**	0.8959	0.8648
FE-SSD(-1)	0.8943	40	139	0.9056	**0.8881**	0.9038	0.8922	0.9012	0.8972	0.8732
FE-SSD	**0.895**	38	141	0.9056	0.8862	0.9029	**0.8934**	0.9018	**0.8986**	**0.8768**

TABLE VI
PERFORMANCE COMPARISON RESULTS FOR DIFFERENT INPUT SIZES BASED ON THE RAILWAY TRAFFIC DATASET. BOLD FONT INDICATES THE BEST RESULT.

Method	mAP	FPS	Bullet train	Pedestrian	Railway straight	Railway left	Railway right	Helmet	Spanner
FE-SSD-512	**0.8963**	15	**0.906**	**0.8879**	0.9014	**0.8938**	0.9014	**0.9031**	**0.8804**
FE-SSD-320	0.895	38	0.9056	0.8862	**0.9029**	0.8934	**0.9018**	0.8986	0.8768

TABLE VII
PERFORMANCE COMPARISON FOR DIFFERENT BACKBONES BASED ON THE RAILWAY TRAFFIC DATASET. BOLD FONT INDICATES THE BEST RESULT.

Method	Backbone	mAP	FPS	Bullet train	Pedestrian	Railway straight	Railway left	Railway right	Helmet	Spanner
FE-SSD/O	VGG16	0.8718	52	0.8928	0.8598	0.8954	0.8806	0.8994	0.8498	0.8248
FE-SSD/O	MobileNet	0.8532	102	0.8938	0.8291	0.8965	0.8588	0.9005	0.8155	0.7783
FE-SSD/O	ResNet-101	0.8722	10	0.8962	0.8574	0.8959	0.8793	0.9012	0.8522	0.8233
FE-SSD	VGG16	0.895	38	**0.9056**	0.8862	**0.9029**	0.8934	0.9018	**0.8986**	0.8768
FE-SSD	MobileNet	0.8814	72	0.9052	0.8859	0.9018	0.8821	**0.9049**	0.8672	0.8233
FE-SSD	ResNet-101	0.8956	6	0.9049	**0.8871**	0.9025	**0.8941**	0.9025	0.8966	**0.8812**

combined the design of the proposed method with different backbones, the accuracy of all the networks increased, as indicated by rows 5–7 of Table VII. With the backbone of ResNet-101, the proposed method achieved the highest mAP (0.8956) and the lowest detection speed (6 FPS). The backbone of MobileNet yielded the worst mAP (0.8814) and degrading the real-time performance. Moreover, the experimental results suggest that the combination of the PDM, FTB, and RFEM components significantly enhanced the object detection performance.

2) COMPARISON OF DIFFERENT INPUT SIZES

Table VI presents the comparison results for different input sizes. In this experiment, the original image was resized from 640 × 512 pixels to 320 × 320 or 512 ×512 pixels to be input to the network. FE-SSD-512 achieved better performance than FE-SSD-320 for detecting railway traffic objects. Increasing the input size can increase the amount of the feature information for small objects, facilitating their detection. However, FE-SSD-512 achieved a frame rate of 15 FPS, which was lower than that of FE-SSD-320. In practical applications, the tradeoff between the accuracy and the real-time performance must be considered.

3) COMPARISON OF DIFFERENT BACKBONES

We replaced the backbone VGG16 with MobileNet [33] and ResNet-101 [31] to verify the effectiveness of the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

method. For a more intuitive understanding of the effects of the proposed method, we introduced FE-SSD/O, that is, the FE-SSD without the PDM, FTB, and RFEM components. The experiments were conducted using the Railway Traffic Dataset with the aforementioned platform (NVIDIA GTX1080Ti GPU), and the original image was resized from 640 × 512 to 320 × 320 to be input to the networks. The comparison results for different backbones are presented in Table VII. As indicated by rows 2–4 of the table, the backbone of ResNet-101 yielded the highest mAP and the lowest FPS among the three backbones without our designed components. As indicated by row 3 of Table VII, the highest detection speed (72 FPS). Compared with the backbones of MobileNet and ResNet-101, VGG16 achieved a good tradeoff between the accuracy and the real-time performance. The experimental results indicate that the design comprising the PDM, FTB, and RFEM is effective for accurately detecting railway objects.

V. CONCLUSIONS

We proposed an FE-SSD based on a PDM, an FTB, and an RFEM for railway object detection. First, PDMs are used to filter out the most negative anchor boxes and provide initial boxes to narrow the search space of objects. Then, RFEs are used to expand the receptive field of the feature maps, yielding richer feature information, particularly for small objects. Finally, FTBs are employed to fuse feature-map information of adjacent layers, which is conducive to small-object detection. The FE-SSD achieved a mAP of 0.895 and a frame rate of 38 FPS with an input size of 320 × 320 pixels, outperforming four other detectors for the Railway Traffic Dataset, and had similar real-time performance to the SSD. The experimental results indicate that the FE-SSD is more suitable than the SSD for railway object detection. Owing to its advantages, the FE-SSD has potential for other applications requiring object detection.

ACKNOWLEDGMENT

We thank the China University of Mining and Technology and the Beijing and Beijing Institute of Remote Sensing Equipment for providing the experimental hardware platform. Additionally, we appreciate the Railway Bureau providing us with the field site for collecting experimental data. This work was also supported by the Fundamental Research Funds for the Central Universities (2020XJJD03).

REFERENCES

[1] 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. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 5936–5944.

[2] 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.

[3] Y. Chen, D. Zhao, L. Lv, and Q. Zhang, "Multi-task learning for dangerous object detection in autonomous driving," *Inf. Sci.*, vol. 432, pp. 559–571, 2018.

[4] A. Ucar, Y. Demir, and C. Güzelis, "Object recognition and detection with deep learning for autonomous driving applications," *Simulation*, vol. 93, no. 9, pp. 759–769, 2017.

[5] Y. Ye, L. Fu, and B. Li, "Object detection and tracking using multi-layer laser for autonomous urban driving," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, 2016, pp. 259–264

[6] Y. Hou, H. Zhang, S. Zhou, and H. Zou, "Efficient convnet feature extraction with multiple roi pooling for landmark based visual localization of autonomous vehicles," *Mob. Inf. Syst.*, 2017:8104386:1–8104386:14, 2017.

[7] Wikipedia contributors, "List of rail accidents (2000–2009)." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 2 Aug. 2019. Web. 5 Aug. 2019.

[8] Wikipedia contributors. "List of rail accidents (2010–present)." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 2 Aug. 2019. Web. 5 Aug. 2019.

[9] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 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, 2016.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision—ECCV 2016*, Springer, Berlin, Germany, 2016, pp. 21–37.

[12] T. Takeda, "Improvement of railroad crossing signals," in *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No. 99TH8383)*, Tokyo, Japan, 1999, pp. 139–141.

[13] S. P. Lohmeier, R. Rajaraman, and V. C. Ramasami, "An ultra-wideband radar for vehicle detection in railroad crossings," in *Sensors*, *2002 IEEE*, Orlando, Florida, USA, vol. 2, 2002, pp. 1762–1766.

[14] A. H. Narayanan, P. Benjamin, R. Benjamin, N. Mazzino, G. Bochetti, and A. Lancia, "Railway level crossing obstruction detection using MIMO radar," in *2011 8th European Radar Conference*, Manchester, UK, 2011, pp. 57–60.

[15] M. Watanabe, K. Okazaki, J. Fukae, N. Tamiya, N. Ueda, and M. Nagashima, "An obstacle sensing radar system for a railway crossing application: A 60 GHz millimeter wave spread spectrum radar," in *2002 IEEE MTT-S International Microwave Symposium Digest (Cat. No. 02CH37278)*, Seattle, WA, USA, vol. 2, 2002, pp. 791–794.

[16] G. Kim, J, Baek, H. Jo, K. Lee, and J. Lee, "Design of safety equipment for railroad level crossing using laser range finder," in *Proc. 9th Int. Conf. Fuzzy Systems and Knowledge Discovery*, Sichuan, China, 2012, pp. 2909–2913.

[17] Z. Silar and M. Dobrovolny, "Utilization of directional properties of optical flow for railway crossing occupancy monitoring," in *2013 International Conference on IT Convergence and Security (ICITCS)*, Macao, China, 2013, pp. 1–4.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

[18] R. Oorni, "Reliability of an in-vehicle warning system for railway level crossings-a user-oriented analysis," *IET Intel. Transport Syst.*, vol. 8, no. 1, pp. 9–20, 2014.

[19] Z.G. Zhang, X.F. Li, and Y.L. Gan, "Railway crossing intelligent safety alarming system design based on PLC," in *26th Chinese Control and Decision Conference (2014 CCDC)*, Changsha, China, 2014, pp. 5045–5048.

[20] Z. Kim and T. E. Cohn, "Pseudoreal-time activity detection for railroad grade-crossing safety," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 319–324, 2004.

[21] S. Mockel, F. Scherer and P. F. Schuster, "Multi-sensor obstacle detection on railway tracks," in *Proc. IEEE Intelligent Vehicles Symp.*, Columbus, OH., USA, 2003, pp. 42–46.

[22] J. F. González, J. L. Galilea, M. M. Quintas, and C. A. Vázquez, "Sensor for object detection in railway environment," *Sensor Letters*, vol. 6, no. 5, pp. 690–698, 2008.

[23] C. P. Wei, Y. M. Huang, Y. C. F. Wang, and M. Y. Shih, "Background recovery in railroad crossing videos via incremental low-rank matrix decomposition." in *Proc. 2013 2nd IAPR Asian Conf. Pattern Recognition*, 2013, pp. 702–706.

[24] 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, 2014.

[25] T. Zhu, F. Liu, and B. Zhang, "Visual railway detection by super pixel based intracellular decisions," *Multimed. Tools Appl.*, vol. 75, pp. 2473–2486, 2016.

[26] 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, pp. 1137–1149, 2017.

[27] 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, pp. 1904–1916, 2015.

[28] B. Q. Guo and N. Wang, "Pedestrian intruding railway clearance classification algorithm based on improved deep convolutional network," *Opt. Precis. Eng.*, vol. 26, no. 12, pp. 3040–3050, 2018.

[29] 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, pp. 1916, 2018.

[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM int. conf. Multimedia. ACM*, 2014, pp. 675–678.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," [Online]. Available: https://arxiv.org/abs/1409.1556, 2014.

[33] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," [Online]. Available: https://arxiv.org/abs/1704.04861.

TAO YE received B.S. degrees in Measurement and Control Technology and Instrumentation from the China University of Mining and Technology, Xuzhou, China, in 2009. He received an M.S. in Mechanical and Electronic Engineering from the China University of Mining and Technology, Beijing, China, in 2012, and a Ph.D. in Measurement Technology and Instruments from the Key Laboratory of Precision Opto-mechatronics Technology of Ministry of Education, Beihang University, Beijing, in 2016. He worked as an engineer at the Beijing Institute of Remote Sensing and Equipment from March 2016 to March 2019. He is currently a senior engineer at the School of Mechanical Electronic and Information Engineering, China University of Mining and Technology of Beijing. His current research interests include deep learning and traffic detection.

ZHIHAO ZHANG received B.S. degrees in Mechanical Engineering from the China University of Mining and Technology (Beijing), China, in 2014. He is currently a graduate student in the department of Mechanical Engineering at the School of Mechanical Electronic and Information Engineering, China University of Mining and Technology (Beijing). His current research interests include deep learning and railway fault detection.

Xi ZHANG received B.S. degrees in Mechanical Design and Manufacture from the Hefei Industry University, Anhui, China, in 1989. He received M.S. degrees in Hydraulic Transmission and Control from the China University of Mining and Technology, Beijing, China, in 1991, and a Ph.D. in Mine Mechanical Engineering from the China University of Mining and Technology, Beijing, China, in 1995. He received a post-doctorate degree in Mine Mechanical Engineering from Beijing Technology University, Beijing, China, in 1997. He has worked at 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 instrumentation, machine learning, and object detection.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3015251, IEEE Access

FUQIANG ZHOU received B.S., M.S., and Ph.D. degrees in Instrument, Measurement, and Test Technology from Tianjin University, China, in 1994, 1997, and 2000, respectively. He joined the School of Automation Science and Electrical Engineering at Beihang University as a postdoctoral research fellow in 2000. He is currently a professor at the School of Instrumentation and Opto-electronics Engineering at Beihang University, China. His research interests include precision vision measurement, 3D vision sensors, image recognition, and optical metrology.