IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 6, 2018, accepted October 24, 2018, date of publication November 9, 2018,
date of current version December 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879270

Real-World Railway Traffic Detection Based on Faster Better Network

JUAN LI[1], **FUQIANG ZHOU**[2], **AND TAO YE**[2]
[1]School of Instrumentation Science and Opto-electronics Engineering, Beihang University, Beijing 100083, China
[2]Second Research Institute, China Aerospace Science and Industry Group, Beijing 100083, China

Corresponding author: Fuqiang Zhou (zfq@buaa.edu.cn)

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant 61471123.

ABSTRACT Detection of railway shape and dangerous obstacles plays a critical role in the auxiliary driving of the train. Speed and accuracy are both of great significance to real-world railway traffic detection, which demands a higher efficiency and effectiveness. The goal of this paper is to design an architecture that achieves the right speed (for effectiveness)/accuracy (for effectiveness) balance for actual railway detection. Driven by this motivation and based on the advantages of some current algorithms, we propose FB-Net (faster better network), a robust end-to-end convolutional neural network. Detectors based on deep learning method are composed of feature extraction, candidate region generation,and classification. Specifically, our framework is focusing on with three embedded modules: 1) To improve efficiency, we replace standard convolutions with depthwise-pointwise convolutions in the feature extraction stage, aiming to red reduce model parameters; 2) To address the effectiveness, a priori module is added for candidate boxes to provide a coarse location for subsequent regressor and to reduce the searching space of objects significantly; 3) Meanwhile, we design a feature fusion module to enhance the semantic context interaction of adjacent feature maps for better detection of small objects. Experiments for railway traffic datasets on both computer device and mobile device demonstrate that FB-Net achieves good results when the input size is 320 pixels × 320 pixels.

INDEX TERMS Railway traffic detection, efficiency and effectiveness, depthwise-pointwise convolution, priori module, feature fusion.

I. INTRODUCTION

Since the first successful demonstrations in [1] and [2], much attention has been put in the field of intelligent transportation. Usually, autonomous traffic system operates in a dynamic and complex environment, and final decision needs accurate perception to handle some unpredictable situations in a timely manner. Reliable detection of objects is essential in traffic surveillance and auto-driving. For example, in urban traffic, cars are sharing the road with other traffic participants, such as pedestrians, bicycles, and animals. Awareness of these life threatening factors can help to prevent accidents. In railway traffic, auxiliary driving requires advanced judgements of the railway shape, and obstacles need to be recognized in case of catastrophes.

In the recent years, with the emergence of Age of Big Data and the rapid development of GPU hardware technology, significant progress has been made on visual object detection [3]–[7]. Many convolutional neural network based detectors show their strength on public benchmarks neither in accuracy nor in speed. However, there are limited studies on analyzing the performance of object detectors on realistic scenarios, and existing off-the-shelf detectors still face significant challenges when deployed in real-world traffic surveillance systems because of the wide range of appearance variations. First, various weather conditions and lighting effects lead to differences of images. Second, different camera placements and object pose variations are another source of dramatic changes in object appearance. Third, limited computing capacity of mobile devices demands smaller model size and higher speed.

Different from urban traffic monitoring, railway detection needs to operate and make judgement timely in a high speed movement. In this paper, we address the problem of object detection in practical railway traffic situation by designing a deep convolutional architecture. Towards this goal, we first discuss two problems of the railway traffic, describing the demands of the each solution. Second, we propose a novel detector named FB-Net aiming at boosting the balance between accuracy and efficiency for real-world railway traffic detection, including railway shapes and other obstacles.

The two fundamental problems of railway traffic detectors are efficiency and effectiveness. On the one hand, detector system needs to process images in a timely fashion on a computationally limited platform because the processor is installed on the train and moves with the train. This needs a more efficient network architecture with less parameter. On the other hand, accuracy is of crucial importance as life and property loss caused by train catastrophes is larger than other accidents. So the detector should keep high standard accuracy metrics, such as average precision (AP). Driven by the two demands, we design an elegant framework which is both efficient and effective for practical railway traffic detection. Firstly, depthwise-pointwise convolution, a replacement for standard convolution, is introduced to make deep neural network light weight and to make the system 'faster'. Second, we propose a priori module to provide preliminary locations and to filter out negative proposals. Lastly, we design a feature fusion module to enhance the interaction of semantic information for the limitation with small instances. These two algorithms help detection 'better'. To the best of our knowledge, this framework is novel and has been proven successful in several real-world railway situations. In fact, by transplanting the FB-Net into TX2, an embedded AI computing platform, we achieved better results comparing with many common frameworks, such as SSD.

To summarize, our main contributions are as follows:

A.

We propose FB-Net, a novel convolution framework aiming at boosting a better balance between efficiency and effectiveness. Depthwise-pointwise convolutions stage, a priori module and a feature fusion module are embedded in this architecture.

B.

With low resolution input size of 320 pixels × 320 pixels and GeForce GTX1080Ti hardware on a computer device, FB-Net achieves state-of-the-art result on real-world railway dataset, with 87.6% mAP and 82 FPS.

C.

For NVIDIA Jetson TX2, the detection speed of FB-Net can reach 20 FPS, which can increase to 30FPS when the network channel is half the original.

The rest of this paper is organized as follows. Section 2 introduces the related work for current object detectors. Section 3 discusses our evolving framework in detail while Section 4 demonstrates the experiments and the results. The conclusion is drawn in Section 5.

II. RELATED WORK

A. TRAFFIC DETECTION

Traffic detection, as an essential part of traffic surveillance and auto-driving, has produced many classic vehicle detectors, which have achieved promising detection results. It has drawn considerable attention when the cascade methods [8] and the deformable part models (DPM) [9] detectors are introduced. And Viola and Jones [8] propose the initial cascaded vehicle detection with a set of weak classifiers to early filter image patches that are not target objects. Later studies make some extent on this basis and achieved good performance [10], [11]. These methods are based on traditional feature extraction and the information may not be sufficient. For commercial systems, most of them rely on background modeling techniques for detecting moving blobs as a proxy for objects in the scene, which is limited to low-activity scenarios [12]. Feris and Bobbitt [13]propose a large set of complementary and extremely efficient detector models to address the problem of object detection in urban surveillance videos. By exploring scenic consistency information, Zhang *et al.* [14]propose a view independent objection classification system for traffic scene surveillance. According to the achieved object tracking result, an improved background modeling and foreground segmentation approach based on the feedback of moving objects is proposed [15]. Gibert *et al.* propose a multiple tasks architecture for fasteners inspection, which included two branches for coarse-level classification and refined classification separately. The results show better accuracy in detecting defects on railway ties and fasteners. Jun *et al.* adopt a three stages cascade detector for defect detection of the fasteners. Both of them are in a coarse-to-fine manner. One limitation of this method is that the result of the former stage will affect the detection of the latter stage. In recent years, deep learning framework is introduced to vehicle detection.Wang *et al.* [16]design a light-weight proposal network with a fine-tuning network for traffic surveillance. However, these methods are focusing on urban traffic monitoring and are not suitable for railway traffic detection because of limited efficiency as new characteristics are present in this scene.

B. OBJECT DETECTION BY CNN

Basically, CNN is a kind of network with many layers to extract feature based on invariance of regional statistics with respect to pixel location in an image. After CNNs are initially rekindled by their use for image classification [5], they are quickly applied to object detection. Comparing with classification task, one cares not only about classifying images, but also precisely estimating the class and location of objects contained within the images for detection task. OverFeat [17] is firstly proposed to predict the class label and the bounding box coordinates by applying a sliding window on the topmost feature map. Nowadays, CNN detectors of state-of-the-art can be divided into two categories: (1) the one-stage approach, including [18]–[20], detects objects by regular and dense samplings over locations, scales and aspect ratios. The bounding boxes are regressed directly with a confidence which represents the reliability of a predicted result. The main advantage of this method is high computational efficiency, but the detection accuracy of the one-stage approach is low and one major reason is the class imbalance problem.

(2) The two-stage approach, including [21]–[24], generates a sparse of candidate object boxes first and then they are further classified and regressed. This method has been achieving top performances on many benchmarks, such as PASCAL VOC and COCO. In my opinion, the reason for good performance is that the two-stage approach has following advantages over the one-stage approach: using former step with sampling heuristics to avoid class imbalance problem and using two cascade regressions for precise location. Unfortunately, two-stage approach is not suitable for some real-time situations.

Nowadays, some improvements and innovations have made to meet different needs of detectors. To address the class imbalance problem and to improve the accuracy performance for one-stage approach, Kong *et al.* [25], use objectness priori constraint for convolution stage aiming at reducing the search space significantly. Lin *et al.* [26] solve this problem by reshaping the standard cross entropy loss to focus training on a sparse set of hard examples. In order to improve the accuracy, DSSD [27] suggests enhancing one-stage approach with deconvolution layers for additional large-scale context. As for the model size and efficiency, there has been rising interest in designing small and low latency models in some recent literature. Depthwise separable convolutions are initially introduced in [28] and are subsequently used in Inception models [29]. Flatten networks [30] and Factorized networks [31] build factorized convolutions which show the extreme potential. MobileNet [32], which also adopts depthwise separable convolutions, achieves great performance on resource and accuracy tradeoff and can be used in many applications including object detection, classification, face attributes and large scale geo-localization.

This paper inherits the merits of the one-stage approach and the two-stage approach as we design a priori module similar to Kong. Besides, we have to take speed and accuracy into account at the same time, as well as the diversity and complexity of the image quality for real-world railway traffic. Some innovations have been made in FB-Net for better efficiency and effectiveness.

III. APPROACHES

This section describes the three core modules of our proposed framework FB-Net. We first introduce depthwise-pointwise convolution in section 3.1. Then in section 3.2, we present the priori module to guide the search of objects. Next, we explain how feature fusion module works, such that different feature maps have effective interaction. Finally, we integrate these strategies into the FB-Net in section 3.4.

A. DEPTHWISE -POINTWISE CONVOLUTION

As we all know, a standard convolution operates on both space and channel which leads to a great amount of calculations. In this study, we factorize a standard convolution into two convolution layers: a depthwise convolution layer for filtering and a pointwise convolution layer for combining as [32]. To be explained in detail, the depthwise convolution applies a single filter on each input channel while the point convolution operates on outputs of depthwise convolution. In Figure 1, top row presents how a standard convolution filter works and bottom row shows how a standard convolution is factorized into a depthwise convolution and a point wise convolution. Figure 2 presents the structure changes.

Assuming that the size of the input feature map is $F_{in\_w} \times F_{in\_h} \times m$ (F_{in_w} and F_{in_h} is the width and height of the feature map, respectively), and the kernel size of the filters is $K_w \times K_h$, the convolution operation produces a $F_{out\_w} \times F_{out\_h} \times n$ feature map. Comparison of a standard convolution and a depthwise-pointwise convolution from the number of parameters and the computational cost are as follows. Equation (1) and (2) are the functions to calculate the number of parameters (P), and (5) means that depthwise-pointwise convolution uses about nine times fewer parameters than standard convolution with the kernel size 3. As for the computational cost(C), we pay attention to multiplication while the add operation cost is ignored, and the results are shown as (3) and (4). Equation (6) presents the reduction of computation in the form of proportion.

$$P_{s\tan dard} = K_w \times K_h \times m \times n \tag{1}$$

$$P_{D-P} = K_w \times K_h \times m + 1 \times 1 \times m \times n \tag{2}$$

$$C_{s\tan dard} = K_w \times K_h \times m \times F_{out\_w} \times F_{out\_h} \times n \tag{3}$$

$$\begin{aligned} C_{D-P} &= K_w \times K_h \times F_{out\_w} \times F_{out\_h} \times m \\ &\quad + 1 \times 1 \times m \times F_{out\_w} \times F_{out\_h} \times n \end{aligned} \tag{4}$$

$$\begin{aligned} \frac{P_{D-P}}{P_{s\tan dard}} &= \frac{K_w \times K_h \times m + 1 \times 1 \times m \times n}{K_w \times K_h \times m \times n} \\ &= \frac{1}{n} + \frac{1}{K_w \times K_h} \end{aligned} \tag{5}$$

$$\begin{aligned} \frac{C_{D-P}}{C_{s\tan dard}} &= \frac{K_w \times K_h \times F_{out\_w} \times F_{out\_h} \times m}{K_w \times K_h \times m \times F_{out\_w} \times F_{out\_h} \times n} \\ &\quad + \frac{1 \times 1 \times m \times F_{out\_w} \times F_{out\_h} \times n}{K_w \times K_h \times m \times F_{out\_w} \times F_{out\_h} \times n} \\ &= \frac{1}{n} + \frac{1}{K_w \times K_h} \end{aligned} \tag{6}$$

B. PRIORI MODULE

Usually, the two-stage approaches, such as Faster R-CNN, achieve higher accuracy because of the mechanism that picks out effective samples and produces coarse locations for proposals. One-stage approaches, such as SSD, rely on one-step regression to predict the locations and sizes of objects, which are inaccurate in some challenging scenes. In order improve the accuracy of one-stage approach, this study design a priori module to handle the class imbalance problem and to use two step cascade regressor for object proposal parameters.

Similar to Faster-RCNN, n anchor boxes are associated with each cell of the feature map and each anchor box has a fixed initial position. First regression to predict four offsets of these boxes is made in anchor priori module. This operation provides a coarse location for sub-sequent regressior. Between all of these anchor boxes, only a tiny fraction covers objects and we call them positive anchors. In other words, the ratio between object and non-object boxes

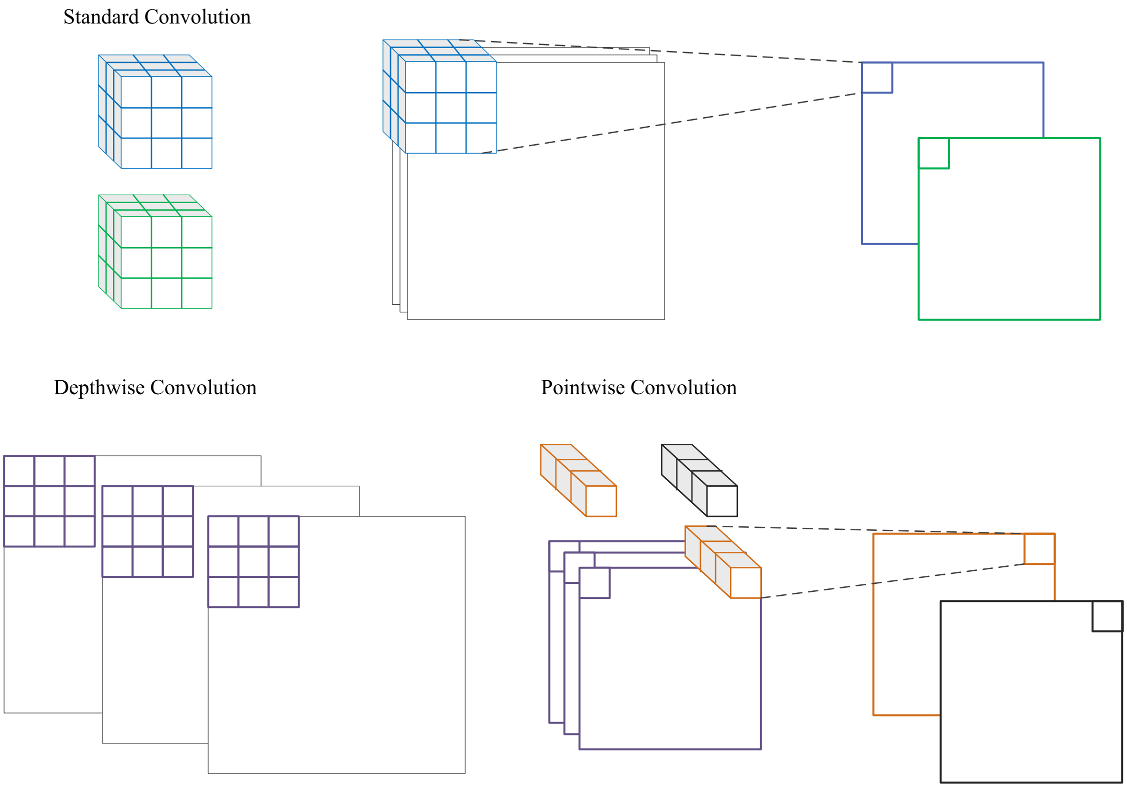


FIGURE 1. The convolution operation schematic. Standard convolution filers in top row are split into two steps: depthwise convolution in bottom row left and pointwise convolution in bottom row right.

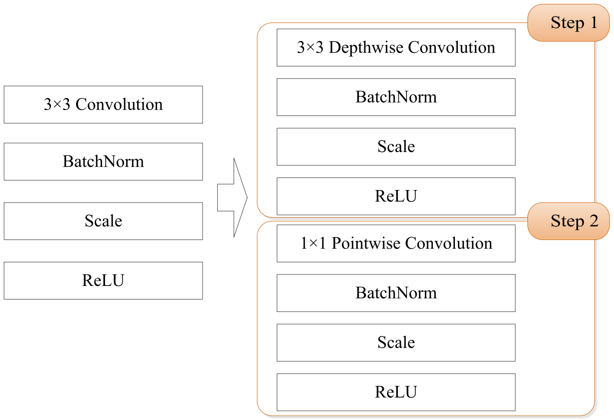


FIGURE 2. Left column: structure of standard convolution. Right column: structure of depthwise-pointwise convolution.

is seriously imbalanced. We propose a rule to filter lots of well classified negative anchors to mitigate the imbalance issue. Only refined positive anchors and negative anchors with confidence score less than 0.99 are passed for later detector. Region generation network (RPN) in Faster-RCNN generates anchor boxes only on the feature maps of the last convolution stage. For better detection of multi-scale objects, we use multi feature maps to generate anchors with different ratios unlike RPN. And feature maps, whose receptive fields are 8 × 8, 16 × 16, 32 × 32, 64 × 64, are selected to generate proposals. Besides, we can design the distribution of boxes so that specific feature map locations can be learned to be responsive to particular scales of objects. In conclusion, this module provides priori information which includes initial position for more accurate detection and reduces searching space of the objects.

C. FEATURE FUSION MODULE

Many studies have proved that highly-abstracted information helps object detection, particularly for small targets. We use different feature maps to detection objects with different scales. The main limitation of the feature pyramid is that it lacks fusion between different features maps. Particularly for small target, feature map with 8 × 8 receptive field is used to make predictions. However, the representation ability of this feature map is relatively weak compared with higher feature map. Inspired by the success of integrating context in DSSD [27], we design a feature fusion module to help to send high-level features back to former layers. This operation makes interaction between adjacent feature maps and enriches semantic information of former layers. We adopt element-wise summation to merge the two corresponding feature maps together. With the requirement that feature maps should have the same size and dimension, we firstly use deconvolution operation to enlarge the high-level feature maps. After element-wise summation, a standard convolution is needed to ensure the discriminability of features for object

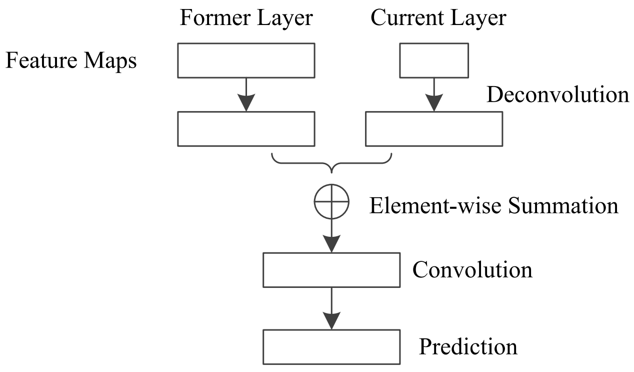


FIGURE 3. Feature fusion module.

detection. The feature fusion module is interpreted in Figure 3 and the mathematical definition will be given. Assuming that $\{X_i, i \in C\}$ are the source feature maps generated by the convolutional layer, the feature fusion module can be described as follows:

$$loc, class = D(\tau_n(\Phi_n), \ldots, \tau_{n-k}(\Phi_{n-k})), \quad n > k > 0 \tag{7}$$

$$\Phi_m = \phi_f(X_i, X'_{i+1}), \quad m \in [n-k, n-1] \tag{8}$$

$$X'_j = \phi_d(X_j), \quad j \in [n-k+1, n] \tag{9}$$

where ϕ_f is the feature fusion function. ϕ_d is the deconvolution function to generate feature maps twice of the original scale. D is the final operation to aggregate all the intermediate results to generate the final detection while $\tau_n(\cdot)$ is the function to transform the *nth* layer feature maps to the detection result of a certain range.

D. FB-NET ARCHITECTURE

Based on a feed-forward convolutional network, we design our framework called FB-Net, which is built on depthwise-pointwise convolution, priori module and feature fusion module as mentioned in the previous section. Our final purpose is to boost the efficiency and effectiveness of the detector for real-world railway traffic. In specific, depthwise-pointwise convolutions are used for efficiency while the other two modules are both for effectiveness. During the convolution stage for feature extracting, we replace all regular convolutions with depthwise-point convolutions except for the first layer. By defining the network in such simple ways, we can reduce model parameters and computation cost greatly. Particularly, down sampling is by the way of strided convolution. Similar to classic one-stage approach SSD, FB-Net produces a fixed number of bounding boxes of different scales with the corresponding inference which indicating the possibility of different classes, followed by non-maximum suppression for final detection. However, before making predictions, we add a priori module to remove most negative samples so as to reduce search space and also provide coarse locations for subsequent prediction module. The feature fusion module is also implemented between priori module and prediction module, which integrates semantic information of the adjacent feature maps. The sizes of the feature maps to make predictions are 40 × 40, 20 × 20, 10 × 10, and 5 × 5. The channels of the layers that are feed into element-wise summation are 256. In conclusion, the FB-Net architecture is defined in Figure 4.

IV. EXPERIMENTS AND RESULTS

In this section, we first introduce the datasets used in the experiments and then methods of data augmentation are presented. Finally, many experiments are carried out on railway datasets to explain the superiority of our model. For better comparison, we also make experiments on classical one-stage approach SSD and two-stage approach Faster-RCNN as well as DSSD. Results from three aspects: mAP performance, model size, and speed performance are presented. We obtain all of these results on caffe platform.

A. DATASETS

A camera is placed on the train to capture the real-world railway-traffic images as seen in Figure 5. To ensure the diversity of the data, pictures are obtained from various scenes including different weather conditions, different lighting conditions and also different speed conditions. For training, we sample spatially evenly from each raw video sequence from the training dataset, generating roughly 7342 images and the size of original frames is 640 pixels × 512 pixels. With the consideration of railway shapes and possible obstacles in assisted driving, we labeled the images with seven classes, including Bullet Train, Pedestrian, Railway Straight, Railway Left, Railway Right, Helmet and Spanner. 83% of these images are used for train-val dataset while the rest are for test.

B. DATA AUGMENTATIONS

In order to expand existing datasets and construct a more robust model, we augment the training data in an online manner with several random transformation, including:

- *Scale*: images are scaled by a random number $s \in [0.3, 1.0]$.
- *Ratio*: image aspect ratios are changed between 0.5 and 2.
- *Color Jitter*: the brightness, contrast, and saturation of images are each scaled by $k_i \in [0.5, 1.5]$.
- *Color Normalization*: RGB is normalized through mean subtraction.
- *Flips*: images are flipped with a 50% chance.

We randomly select one patch of the above options so that the minimum jaccard overlap with objects is 0.1, 0.3, 0.5, 0.7 or 0.9. After the aforementioned sampling step, each sampled patch is resize to 320 pixels × 320 pixels so that the input size to the network is consistent.

C. PERFORMANCE

The baseline of our experiments is VGG16 [33] and is pre-trained on the ILSVRC CLS-LOC dataset. The dimension of the input image is 320 pixels × 320 pixels. We replace fc6 and fc7 of VGG16 with depthwise-pointwise convolution

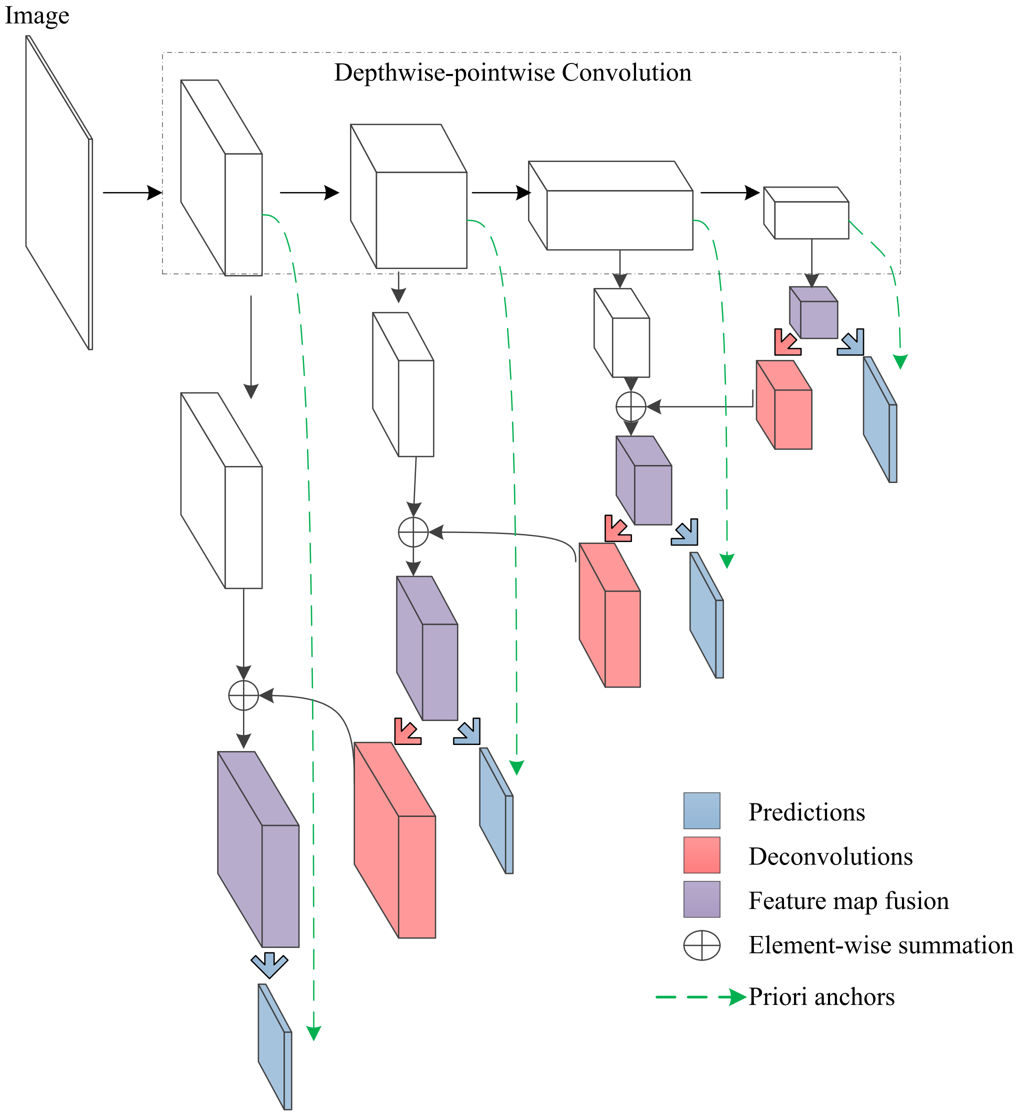
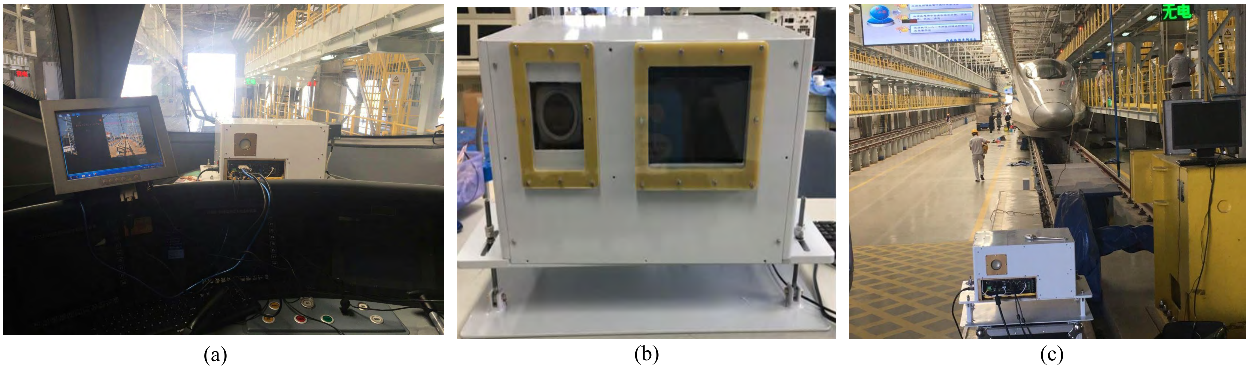


FIGURE 4. Feature fusion module.



(a) (b) (c)

FIGURE 5. Configuration of image collection system.

layers via subsampling parameters as well as DeepLab-LargeFOV [34]. In order to capture high-level semantic information and detect objects at multiple scales, two extra layers are also added to the truncated VGG16. In conclusion, feature maps of sizes 40 × 40, 20 × 20, 10 × 10, and 5 × 5 are used to make predictions. Some relevant and important parameters are set as follows. We set the default batch size to 32, optimization method to SGD with 0.9 momentum and 0.0005 weight decay, and initial learning rate to 0.001. The maximum number of iterations of all experiments is 120000.

1) EFFECTIVENESS PERFORMANCE

Average Precision (AP) is a critical evaluation index for model effectiveness. With recall rate as x axis, precision rate as y axis, AP calculates the area under the curve which combines both of the two indexes. The results are presented in Table 2. To illustrate the good use of the anchor priori module and the feature fusion module for effectiveness, we introduce B-Net, which is FB-Net without depthwise-pointwise convolution module and is present in the second row in Table 1. The result presents that B-Net achieves the best result of five classes. In particular, the AP value for small targets helmet and spanner is increased by 8.1% and 10.2% respectively compared with F-Net(Detailed in effiviency performance). In the fourth column, FB-Net produces 87.6% mAP without bells and whistles, and just decreases by 1.7% compared with B-Net. For small targets, FB-Net also achieves excellent performance with little reduction.

TABLE 1. Models of various designs.

Component	Depthwise-pointwise convolution?	Priori module?	Feature Fusion module?
F-Net	√		
B-Net		√	√
FB-Net	√	√	√

TABLE 2. Detection results with different methods. All methods are trained on the same trainval sets and tests on the same test set. Bold fonts indicate the best result.

Method	F-Net	B-Net	FB-Net	FB-Net_0.5
Bullet Train	0.8950	**0.9082**	0.9062	0.8951
Pedestrain	0.8285	**0.8853**	0.8294	0.7326
Railway Straight	0.9005	**0.9073**	0.9057	0.9067
Railway Left	0.8596	0.8659	**0.8825**	0.8543
Railway Right	**0.9075**	0.9056	0.9053	0.9046
Helmet	0.8187	**0.8999**	0.8785	0.7488
Spanner	0.7792	**0.8809**	0.8260	0.6328
mAP	0.8556	**0.8933**	0.8762	0.8107

2) EFFICIENCY PERFORMANCE

Both memory utilization and time inference are critical for railway traffic detection system. We use model size and Frame Per Second (FPS) as the evaluation index for efficiency. The results are presented in Table 3. Similar to effectiveness evaluation, F-Net, which is the base model with replacement of depthwise-pointwise convolution, is introduced to demonstrate the speed performance. At test phrase, the speed is evaluated on a machine with GeForce GTX1080Ti, CUDA 8.0 and cuDNN v6. As we can see, the F-Net processes an image in 9.43*ms*(106 FPS) with the input size 320 pixels × 320 pixels, while FB-Net uses 12.5*ms* for a same image. This is 2× faster than the B-Net counterpart. To further reduce the model size, we carry out experiment on the model FB-Net_0.5, whose channels are half of the original FB-Net. The model size and FPS is 13.9M and 115FPS, which are the best results.

TABLE 3. Efficiency performance. Bold fonts indicate the best results.

Method	F-Net	B-Net	FB-Net	FB-Net_0.5
Model size(M)	22.5	127	54.4	**13.9**
FPS	106	44	82	**115**

3) COMPARISION WITH THE *STATE-OF-THE-ART*

In this section, we compare with the existing methods, including one-stage based method SSD, two-stage based method Faster-RCNN and DSSD. The quantitative results are lists in Table 4. Particularly, the backbone of DSSD is ResNet-101 as suggested in its respective paper.

TABLE 4. Comparision with state-of-the-art on the railway traffic dataset. Bold fonts indicate the best results.

Methods	SSD	Faster-RCNN	DSSD	FB-Net
Bullet Train	0.9020	0.9034	0.8977	**0.9062**
Pedestrain	**0.8708**	0.7940	0.8094	0.8294
Railway Straight	0.9013	0.9017	0.8887	**0.9057**
Railway Left	0.8611	0.8712	0.8361	**0.8825**
Railway Right	**0.9079**	0.9061	0.9051	0.9053
Helmet	0.8699	0.7059	0.8318	**0.8785**
Spanner	**0.8368**	0.3467	0.8231	0.8260
mAP	**0.8785**	0.7756	0.8560	0.8762
Model size(M)	98.6	521	623.4	**54.4**
FPS	47	10	13	**82**

Our first observation from column 2 and 5 is that, with the same basic architecture, FB-Net achieves an almost same result while the time reference is 1.7× faster than SSD. There are four classes whose AP result exceeds SSD. In particular, the AP value for small targets helmet and spanner is 88% and 83% respectively. Taking the model size into account, FB-Net is 54.4M and this is acceptable for many mobile devices. As for Faster-RCNN, the experimental results are not good. In my opinion, the reason for this may be that Faster-RCNN may be more suitable for large images as the input size of Faster-RCNN is 1000 pixels × 600 pixels. But the actual image size of our railway is 640 pixels × 512 pixels, and the reverse interpolation may result in inaccurate feature extraction. The model of DSSD also presents good performance. However, the model is too large for many limited platform.

Figure 6 is a scatter plot visualizing the mAP of each of model configurations, with circle radius representing model size, and colors representing models proposed by this paper or not. Running time per image ranges from nine milliseconds to almost 100 milliseconds.The closer to the upper left corner, the better the model. The smaller the radius of the circle, the less parameter the model has. Generally we observe that FB-Net is better than SSD since FB-Net is on the left of SSD, requiring only 12ms per image.

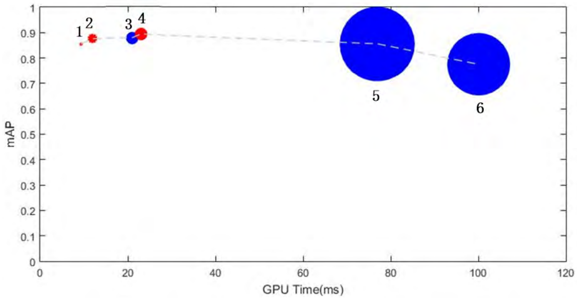


FIGURE 6. GPU Time vsmAP. 1:F-Net; 2:FB-Net; 3:SSD; 4:B-Net; 5:DSSD; 6:Faster-RCNN.

4) SMALL OBJECT DETECTION

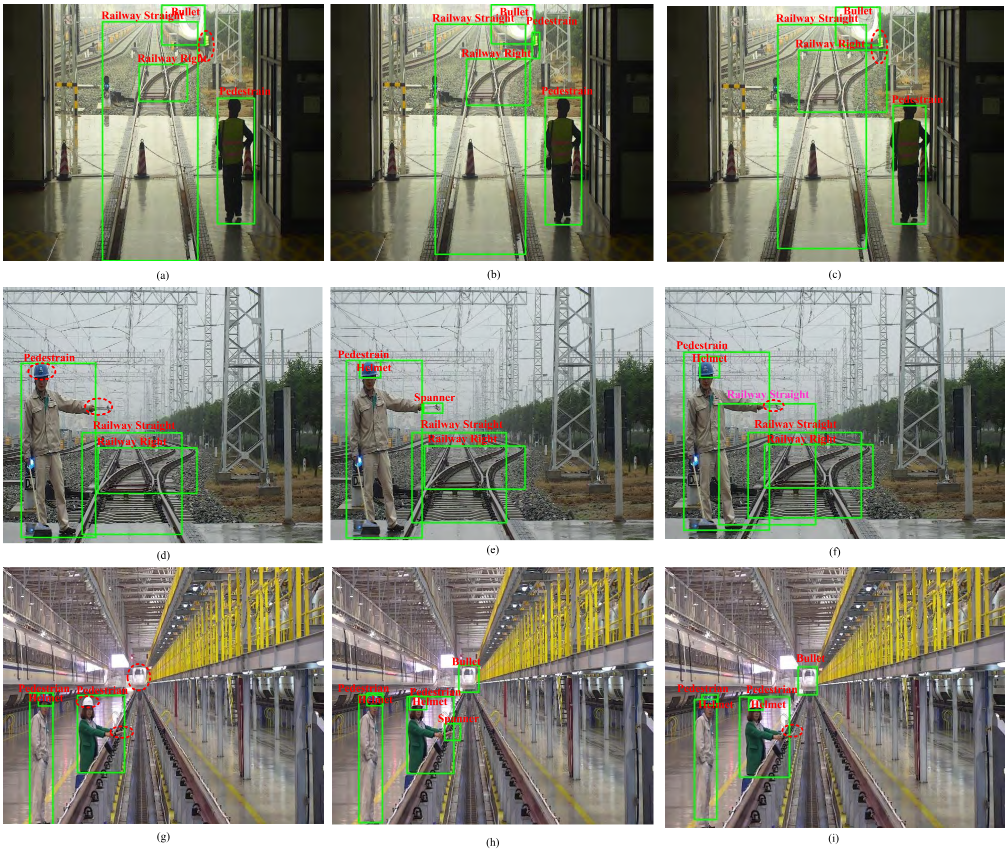
As for the feature fusion module, it helps FB-Net on small objects. On the one hand, small objects can only occupy smaller regions comparing with large objects and the location information is easy to be lost in the convolution process. On the other hand, recognition of small objects relies more on the context around it. Because of detection on multi-layers for different scales, small objects are only detected on 40×40 feature map, whose receptive field is too small to observe the larger context information. Feature fusion model enables former features to have more semantic information. Figure 7 is a good illustration of our models for detection of small objects with the conference threshold 0.85. The three columns are the detection results of SSD, B-Net and FB-Net respectively. Red circles represent the missing detection of small target such as helmet and spanner. Pink part is wrong detection of FB-Net. Compared with SSD, B-Net and FB-Net both achieve good performance on small objects.

5) ROBUSTNESS TESTS

We collect images of all kinds of situation, including day, night, sunny day, rainy day, still train and moving train. Then FB-Net models are tested on these different scenes to verify the robustness. The results are shown in Figure 8 with the conference threshold 0.65. Although some images are in low quality, the proposed model still obtains considerable detection results.

6) MOBILE DEVICE DETECTION

Jetson TX2 is the fastest, most power-efficient embedded AI computing device. However, it's still insufficient compared with computing capability of computer GPU. Table 5 presents the performance of all models in Table 3 and Table 4 on TX2. Faster-RCNN and DSSD contain too many parameters to run

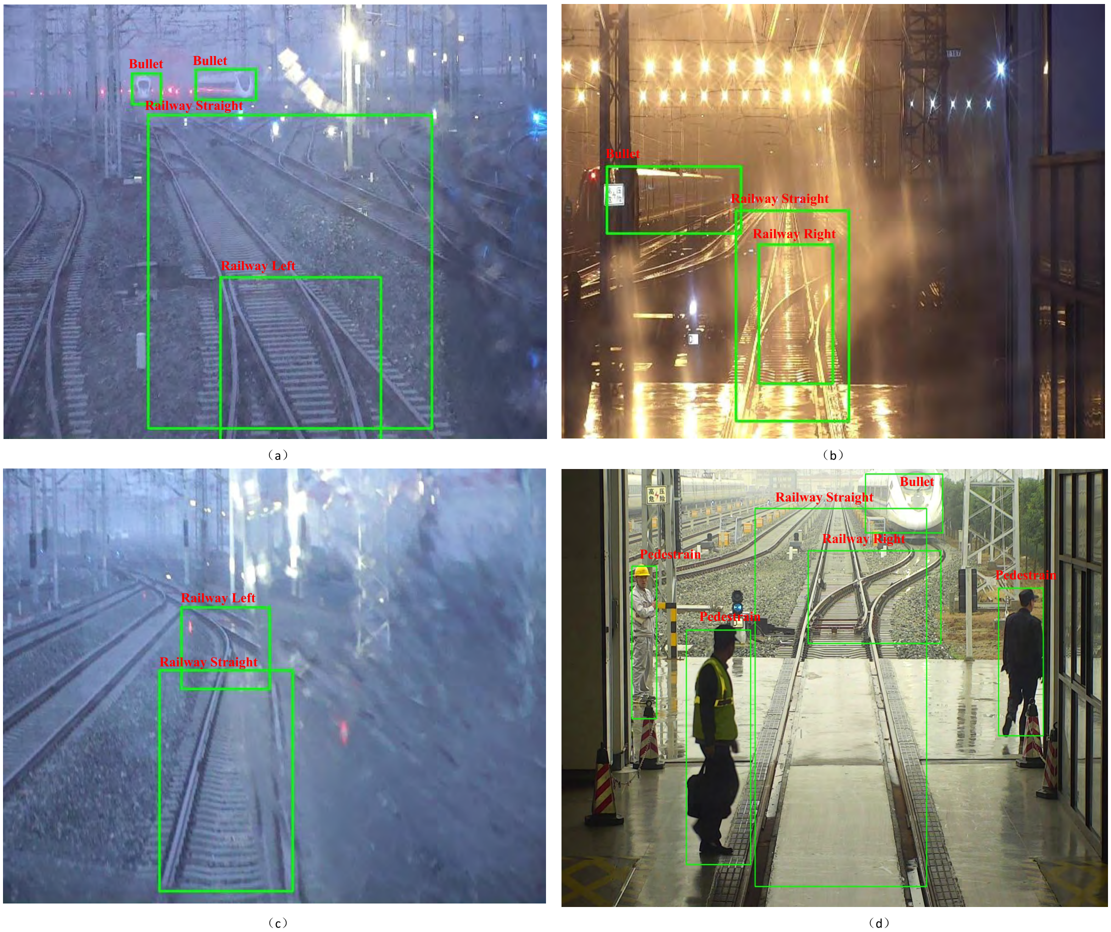


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 7. Display of good performance for small targets. Left column: SSD results. Middle column: B-Net results. Right column: FB-Net result. Red circles represent the missing detection of small targets. Pink part is wrong detection of FB-Net.



(a) (b) (c) (d)

FIGURE 8. Robustness tests for different real-world scenes. (a) Moving train and day. (b) Still train and night. (c) Moving train and rainy day. (d) Still train and sunny day.

TABLE 5. Speed perfomance of different models on TX2.

Methods	FPS
SSD	6
B-Net	5
F-Net	24
FB-Net	21
FB-Net_0.5	33

on this platform and are out of memory. The test speed of SSD model with good AP is only 6FPS while FB-Net can reach 21FPS. The speed difference is of great importance since their APs are nearly similar. It is worth mentioning that FB-Net_0.5 can achieve 33FPS. It provides the evidence that we could reduce the number of channels appropriately to achieve real-time detection on mobile device.

V. CONCLUSION

In this paper, focusing on real-world railway traffic detection, we present FB-Network based on a feed-forward convolutional network. In order to boost both effectiveness and efficiency, three novel portions are embedded in FB-Net. Firstly, we investigate the effectiveness of the depthwise separable convolutions leading to an efficient model. Secondly, a priori module, which removes most negative proposals and provides initial locations, is designed to guide the search of objects. Finally, we apply a feature fusion module to fuse adjacent features together and this enriches semantic information of former layers. Several experiments on railway traffic dataset are carried out. And the results show that FB-Net achieves 87.6 % mAP with 82FPS performance on computer. Experiments with mobile device TX2 verify the engineering potential of FB-Net. In the future, we plan to employ FB-Net to detect objects in some other specific situations. Besides, it is worth exploring the use of attention mechanism or inverted residual structure to further optimize the architecture.

REFERENCES

[1] E. D. Dickmanns and V. Graefe, "Dynamic monocular machine vision," *Mach. Vis. Appl.*, vol. 1, no. 4, pp. 223–240, Dec. 1988, doi: 10.1007/bf01212361.

[2] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-D road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199–213, Feb. 1992, doi: 10.1109/34.121789.

[3] 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, doi: 10.1016/j.neucom.2015.09.116.

[4] Y. LeCun, Y. Bengio, and G. Hinton, ''Deep learning,'' *Nature*, vol. 521, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[6] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, ''Scalable object detection using deep neural networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2155–2162, doi: 10.1109/cvpr.2014.276.

[7] Q. Geng and Z. Zhou, ''Survey on recent progresses of semantic image segmentation with CNNs,'' in *Proc. Int. Conf. Virtual Reality Vis. (ICVRV)*, Sep. 2016, pp. 158–163, doi: 10.1109/icvrv.2016.34.

[8] P. Viola and M. Jones, ''Rapid object detection using a boosted cascade of simple features,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, p. 1, doi: 10.1109/cvpr.2001.990517.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan, ''A discriminatively trained, multiscale, deformable part model,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8, doi: 10.1109/cvpr.2008.4587597.

[10] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, ''A convolutional neural network cascade for face detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5325–5334, doi: 10.1109/cvpr.2015.7299170.

[11] H. Qin, J. Yan, X. Li, and X. Hu, ''Joint training of cascaded CNN for face detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3456–3465, doi: 10.1109/cvpr.2016.376.

[12] S. Varadarajan, H. Wang, P. Miller, and H. Zhou, ''Fast convergence of regularised region-based mixture of Gaussians for dynamic background modelling,'' *Comput. Vis. Image Understand.*, vol. 136, pp. 45–58, Jul. 2015, doi: 10.1016/j.cviu.2014.12.004.

[13] R. Feris, R. Bobbitt, S. Pankanti, and M.-T. Sun, ''Efficient 24/7 object detection in surveillance videos,'' in *Proc. 12th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2015, pp. 1–6, doi: 10.1109/avss.2015.7301791.

[14] Z. Zhang, K. Huang, Y. Wang, and M. Li, ''View independent object classification by exploring scene consistency information for traffic scene surveillance,'' *Neurocomputing*, vol. 99, pp. 250–260, Jan. 2013, doi: 10.1016/j.neucom.2012.07.008.

[15] Q. Ling, J. Yan, F. Li, and Y. Zhang, ''A background modeling and foreground segmentation approach based on the feedback of moving objects in traffic surveillance systems,'' *Neurocomputing*, vol. 133, pp. 32–45, Jun. 2014, doi: 10.1016/j.neucom.2013.11.034.

[16] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, ''Evolving boxes for fast vehicle detection,'' in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 1135–1140, doi: 10.1109/icme.2017.8019461.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun. (Dec. 2013). ''Overfeat: Integrated recognition, localization and detection using convolutional networks.'' [Online]. Available: https://arxiv.org/abs/1312.6229

[18] W. Liu *et al.*, ''SSD: Single shot multibox detector,'' in *Computer Vision—ECCV*. Springer, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[19] 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, doi: 10.1109/cvpr.2016.91.

[20] J. Redmon and A. Farhadi, ''YOLO9000: Better, faster, stronger,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/cvpr.2017.690.

[21] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, ''A unified multi-scale deep convolutional neural network for fast object detection,'' in *Computer Vision—ECCV*. Springer, 2016, pp. 354–370, doi: 10.1007/978-3-319-46493-0_22.

[22] R. Girshick, ''Fast R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448, doi: 10.1109/iccv.2015.169.

[23] 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, doi: 10.1109/tpami.2016.2577031.

[24] A. Shrivastava, A. Gupta, and R. Girshick, ''Training region-based object detectors with online hard example mining,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769, doi: 10.1109/cvpr.2016.89.

[25] 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)*, Jun. 2017, pp. 5936–5944, doi: 10.1109/cvpr.2017.557.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ''Focal loss for dense object detection,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007, doi: 10.1109/iccv.2017.324.

[27] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. (2017). ''DSSD: Deconvolutional single shot detector.'' [Online]. Available: https://arxiv.org/abs/1701.06659

[28] L. Sifre, ''Rigid-motion scattering for image classification,'' Ph.D. dissertation, 2014.

[29] S. Ioffe and C. Szegedy, ''Batch normalization: Accelerating deep network training by reducing internal covariate shift,'' in *Proc. Int. Conf. Mach. Learn. (JMLR)*, 2015, pp. 448–456.

[30] J. Jin, A. Dundar, and E. Culurciello, ''Flattened convolutional neural networks for feedforward acceleration.'' [Online]. Available: https://arxiv.org/abs/1412.5474

[31] M. Wang, B. Liu, and H. Foroosh, ''Factorized convolutional neural networks,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 545–553, doi: 10.1109/iccvw.2017.71.

[32] A. G. Howard *et al.* (2017). ''MobileNets: Efficient convolutional neural networks for mobile vision applications.'' [Online]. Available: https://arxiv.org/abs/1704.04861

[33] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition.'' [Online]. Available: https://arxiv.org/abs/1409.1556v6

[34] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, ''Semantic image segmentation with deep convolutional nets and fully connected CRFS,'' *Comput. Sci.*, vol. 40, no. 4, pp. 357–361, 2014.

JUAN LI received the B.S. degree in measurement–control technology and instrumentation from Tianjin University, Tianjin, China, in 2016. She is currently pursuing the M.S. degree in measurement technology and instruments with the Key Laboratory of Precision Opto-mechatronics Technology of Ministry of Education, Beihang University, Beijing, China. Her current study includes object detection.

FUQIANG ZHOU received the B.S., M.S., and Ph.D. degrees in instrument, measurement, and test technology from Tianjin University, Tianjin, China, in 1994, 1997, and 2000, respectively. He joined the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, as a Post-Doctoral Research Fellow, in 2000. He is currently a Professor with the School of Instrumentation Science and Opto-Electronics Engineering, Beihang University. His current research interests include computer vision and image processing.

TAO YE received the B.S. and M.S. degrees in measurement–control technology and instrumentation from the China University of Mining and Technology, Beijing, China, in 2009 and 2012, respectively, and the Ph.D. degree in measurement technology and instrumentsfromthe Key Laboratory of Precision Opto-mechatronics Technology of Ministry of Education, Beihang University, Beijing, in 2015. He is currently an Engineer withthe Second Research Institute, China Aerospace Science and Industry Group. His current research interests include deep learning and traffic detection.