Real-time Vision Based System of Fault Detection for Freight Trains

Yang Zhang, Moyun Liu, Yunian Chen, Hongjie Zhang, and Yanwen Guo

***Abstract*—Real-time vision based system of fault detection (RVBS-FD) for freight trains aims to complete routine maintenance tasks efficiently for ensuring railway transportation security. However, most existing systems are designed to detect only one specific type of faults or even one fault, which fail to deal with multi-fault detection. Recently, the rapid development of deep learning techniques enables systems to provide a robust solution for the RVBS-FD of freight trains. But general convolutional neural networks (CNN) cannot fully meet the actual requirements in terms of the real-time, accuracy, and resource constrains for the RVBS-FD of freight trains. To solve these problems, we propose a CNN-based detector called Light FTI-FDet for the RVBS-FD of freight train. First, we use the multi-region proposal networks which extract a set of prior bounding boxes to achieve initial fault proposal generation. Then, a powerful multi-level region-of-interest pooling is presented for proposal classification and accurate detection. We finally design a reliable model reduction scheme to pursue fast speed with high detection accuracy in a simple manner. The experimental results on five typical fault benchmarks indicate that our Light FTI-FDet achieves higher accuracy and fast speed with about 17% model size of the well-known Faster R-CNN detector, substantially outperforming state-of-the-art methods.**

***Index Terms*—Real-time, vision based system, fault detection, freight train, convolutional neural network.**

I. Introduction

FAULT detection for railway and train is a seriously important task to ensure traffic safety [1], [2]. For a long time, inspecting mechanic parts of freight trains is carried out by well-trained professionals, which is labor-intensive with low efficiency and easily leads to miss detection. Due to continuing and rapid advances of both hardware and software in camera and computing systems, we can use cheaper, faster, higher quality, and smaller cameras and computing units [3]. As a result, vision-based methods supporting image processing and computational intelligence are implemented more easily than ever using a camera with associated operations [4]. Real-time vision-based system of fault detection (RVBS-FD) is one of the most important infrastructure in the intelligent transportation system. The RVBS-FD has been regarded as a very important technology in Instrumentation and Measurement (IM) field.

The vehicle braking and steering systems of a freight train contain cut-out cocks, dust collectors, bogie block keys, and fastening bolts, etc, all of which are crucial for the train operation safety. The RVBS-FD for freight trains is able to greatly reduce the labor cost and workload of the traditional detection systems by professionals. A typical RVBS-FD consists of an image acquisition hardware and an image analysis software. As shown in Fig. 1, the acquired images are obtained by image acquisition devices that include several monochrome cameras and auxiliary light device. To obtain high quality photos, the cameras are installed on both sides of the railway tracks and the auxiliary light sources are installed on the railway tracks ground.

Due to the complex application environment of railway, the RVBS-FD faces the following difficulties. First, the acquired images are subject to varying illumination, weather and other factors. Second, the backgrounds of freight train images often contain too much structural information with high complexity, leading to be hard to achieve fault detection through visual approaches. Third, the loss, damage and position changes of small parts are the most common faults, which are not obviously different observed from normal (non-fault) and defective (fault) images. Facing these difficulties, how to achieve high detection accuracy with real-time performance is extremely challenging under stringent resource requirements. Hence, fault detection is the most important research content for the RVBS-FD whose performance directly determines the working ability.

The proposed RVBS-FD for freight train in recent years have their own drawbacks and limitations [3], [5]–[10]. Most systems aim at only one specific type of faults or even one fault, which are incapable of dealing with multi-fault detection. For example, Liu et al. [3] present a hierarchical fault inspection system for detecting the missing of bogie block keys on freight trains. Due to the timeliness and accuracy, the existing systems based on conventional machine learning approaches cannot reach the required accuracy. Unlike conventional techniques, deep learning methods [11], [12] especially the convolutional neural networks (CNN), can deal with more difficult problems because deep networks actually implement functions of higher complexity. For example, Sun et al. [12] propose a CNN-based system for recognizing typical faults of freight trains. However, the system consisting of two complex CNN-based models can not achieve real-time fault detection under resource constrains environment, which is not sufficient

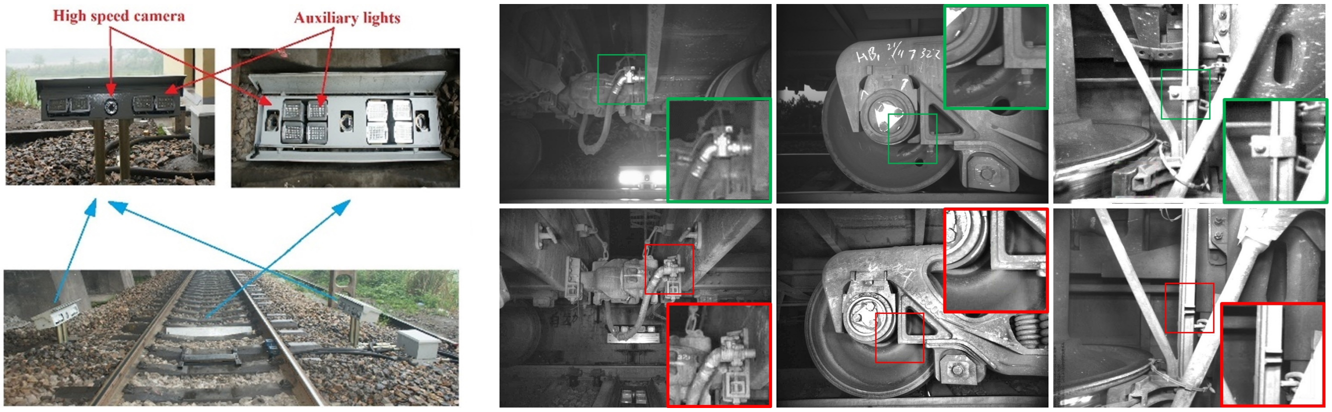
Manuscript received ** **, 2019; revised ** **, 2019. This work was supported in part by the National Natural Science Foundation of China (Grant 61772257 and 61672279), and the Fundamental Research Funds for the Central Universities 020214380058. *(Corresponding author: Yanwen Guo.)*

Y. Zhang, Y. Chen, H. Zhang, and Y. Guo are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: yzhangcst@smail.nju.edu.cn; narcissus-cyn@gmail.com; nju.zhanghongjie@gmail.com; ywguo@nju.edu.cn).

M. Liu is with the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: lmomoy8@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier



(a) Image acquisition devices

(b) Typical samples

Fig. 1: Overview of real-time fault detection for freight train images. (a) Image acquisition devices contain high speed cameras and auxiliary lights, which are installed on both sides and in the middle of railway tracks. (b) Some typical samples of freight train images. The top row shows normal (non-fault) images, and bottom row shows fault images. The loss, damage and position changes of small parts are the most common faults, which are not obviously different observed from non-fault and fault images.

to meet the actual requirements of fault detection. In general, fault detection can be considered as a special type of object detection task in computer vision. For object detection, the region-based CNN (R-CNN) frameworks such as Faster R-CNN (FRCNN) [13] and the region-based fully convolutional networks (R-FCN) [14] are one of the mainstream detectors at present. However, the models of FRCNN and R-FCN are too large, making them difficult to be deployed to the platform and run in real-time under limited computation resource.

In this paper, our motivation is to design a real-time and light-weight fault detection system for freight trains. The core to our system is a deep learning based fault detection method, called Light FTI-FDNet. To avoid the sequential error accumulation of bottom-up fault candidate extraction strategies in deep learning, we propose a multi-region proposal network (MRPN) and introduce a set of prior anchors to achieve high-quality fault region proposals. Then, a powerful detection network is used to achieve accurate detection by incorporating multi-level region of interest (MRoI) pooling. We finally apply a model reduction scheme to pursue the light-weight and fast detection speed. To evaluate the performance of our Light FTI-FDNet, we conduct the experiments on a freight train detection database with five typical faults. The results show that our method can achieve high detection accuracy compared with state-of-the-art methods, and is thus suitable for real-time fault detection task.

A preliminary version of this paper was published as a conference version [15]. Compared with that version, this paper first proposes a real-time, light-weight and accurate detection algorithm named as Light FTI-FDet to achieve real-time fault detection. To this end, we propose a novel multi-level feature fusion strategy containing MRPN and MRoI pooling by connecting different levels of feature maps. As a new algorithmic enhancement, the multi-level feature fusion strategy further perfects the detection stage and improves the detection performance. Furthermore, we introduce a model reduction scheme to reduce model parameters and to improve detection speed in a simple manner. Finally, we perform more thorough experiments to validate the effectiveness of the multi-level feature fusion strategy and model reduction scheme. More extra experiments and theoretical analyses are also added to verify the robustness of the newly proposed RVBS-FD.

II. Related works

With the rapid development of artificial intelligence theory, the automated fault detection based on machine vision has been largely improved. Some of the recent research papers for RVBS-FD of freight trains and object detection are summarized as follows.

A. *Vision-based fault detection for freight trains*

Nan et al. [16] propose an automated and hierarchical inspection system for the detection of the condition of the brake beam bolt based on the fused feature generated by gradient information and basic feature descriptors. Cao et al. [17] propose weighted margin sparse embedded classifier for detecting brake cylinder of freight trains, which can achieve a much greater detection performance. Li et al. [18] propose an automatic fault recognition system for brake shoe key losing of freight train, containing the positioning of the feature region and recognition of the feature for the brake shoe key losing based on edge information and support vector machine. Some systems are extensively applied to inspect other components, such as angle cock [8], brake shoe [10], cut-out cock [19], dust collector [6], fastening bolt [9], and coupler yoke [7], etc.

However, most of these systems aim at only one type of faults. The reason is that the detection algorithms in the system mainly utilize conventional machine learning techniques, which have many weaknesses. Deep learning methods can solve more difficult problems because deep networks actually implement functions of higher complexity. A CNN-based system [12] is proposed for recognizing typical faults, which can greatly deal with low quality images. But this system achieves target region detection and fault recognition relying

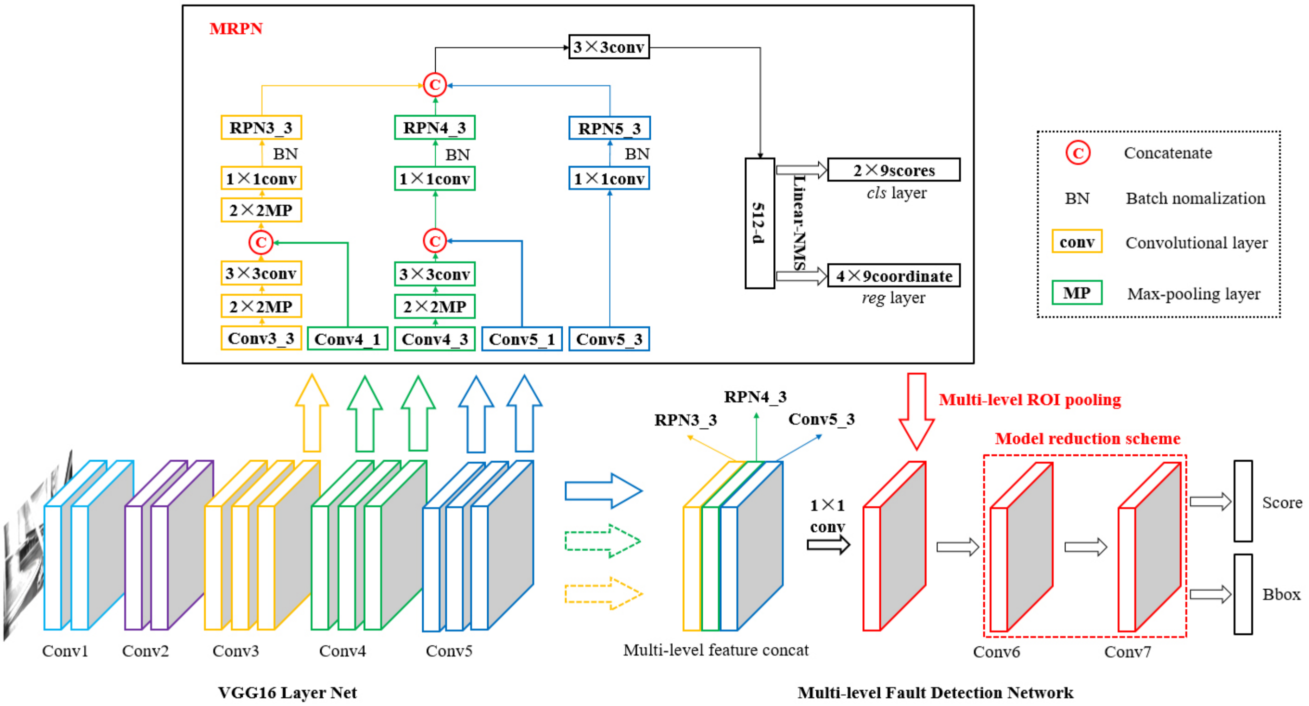


Fig. 2: The detailed architecture of the proposed Light FTI-FDet. Our approach takes an image as input, generates hundreds of fault region proposals via MRPN, and then scores and refines each proposal using the multi-level fault detection network with a MRoI pooling and a model reduction scheme.

on two complex CNN-based models, respectively. Such an approach is insufficient to meet actual requirements of fault detection, such as real-time and versatility.

B. Object detection

With the revival of CNN, CNN-based object detectors have been proposed consecutively. Broadly, these detectors can be organized into two main categories: one-stage and two-stage. The one-stage object detectors directly predict object classes and locations without region proposal generation, which are faster than two-stage object detectors and much more suitable under limited computation resource. For example, You Only Look Once (YOLO) [20], Single Shot multibox Detector (SSD) [21], RefineDet [22], Reverse connection with Objectness prior Networks (RON) [23], RetinaNet [24], CornetNet [25], Receptive Field Block Net (RFBNet) [26], and Deeply Supervised Object Detector (DSOD) [27] have persistently promoted detection accuracy and speed.

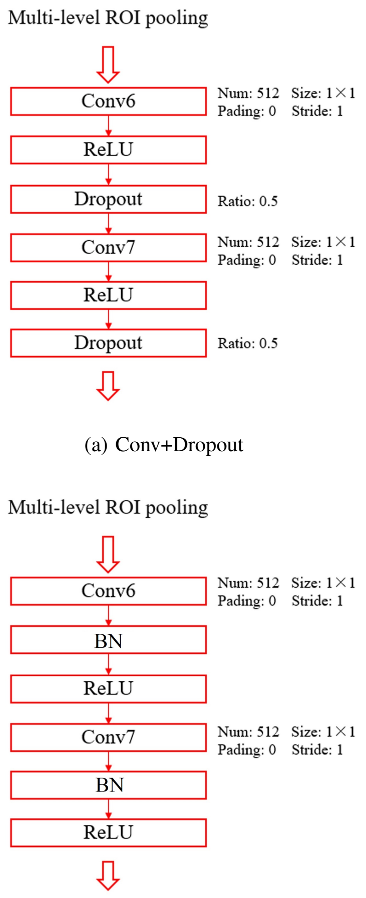
Two-stage object detectors include a pre-processing step for region proposal generation and another step for region classification. As the representative approaches, region-based CNN detectors such as RCNN [28], SPP-Net [29], Fast R-CNN [30], FRCNN [13], Online Hard Example Mining (OHEM) [31], HyperNet [32], Multi-scale CNN (MS-CNN) [33], Feature Pyramid Network (FPN) [34], R-FCN [14], Multi-scale Location-aware Kernel Presentation (MLKP) [35], Cascade RCNN [36], and CoupleNet [37] achieve accurate and effective object detection. In addition, some object detectors achieve the best tradeoff of speed and accuracy. For instance, Light-Head R-CNN [38] improves FRCNN [13] and R-FCN [14] to have high accuracy and speed by using a thin feature map and a R-CNN subnet.

As for RVBS-FD of freight trains, both accuracy and computation complexity are important considerations. To obtain a fine balance between accuracy and computational cost, we take inspiration by the powerful detection performance (high accuracy and low cost) of convolutional neural network demonstrated by the FRCNN [13]. Meanwhile, the multi-level feature fusion strategy in [15] can be used to boost the detection accuracy notably with a low computation.

III. System Overview

The proposed RVBS-FD comprises an image acquisition subsystem and an image analysis subsystem. As shown in Fig. 1(a), the image acquisition subsystem is composed of a device placed in the middle of the rails and two devices placed on one side of the rails. The acquisition devices usually contain several high-speed digital charge-coupled device (CCD) cameras and auxiliary lighting devices. The auxiliary light devices consist of four xenon bulbs installed at the sides of the CCD camera, to eliminate the light interference.

While freight trains pass through, freight train images are first collected by the image acquisition subsystem, and then directly transmitted to remote servers and processed by the image analysis subsystem to efficiently identify whether there are faults in them. The size of each image is fixed to 700×512 pixels. Some typical samples of the vehicle brake and steering system in freight trains are shown in Fig. 1(b).



(a) Conv+Dropout

(b) Conv+BN

Fig. 3: Model reduction scheme. (a) Convolutional layer to replace the FC. (b) Both the convolutional layer and BN layer to replace the FC and dropout.

An image analysis subsystem is a key part in real-time fault detection system. Moreover, a detection algorithm restricts the accuracy and detection speed of image analysis subsystems, as introduced next.

IV. Detection algorithm

The proposed detection algorithm Light FTI-FDet is mainly composed by baseline (*i.e.* a VGG16 model [39]), multi-region proposal generation network, and fault detection network. In this section, we first introduce the backbone architecture of our Light FTI-FDet, and then the multi-region proposal generation network, followed by the fault detection network containing multi-level RoI pooling and a model reduction scheme.

A. Backbone Architecture

As shown in Fig. 2, the detailed architecture of the proposed detection algorithm is inspired by FRCNN and R-FCN. In traditional FRCNN, only on the final feature map layer in the VGG16 model operates the RPN and RoI pooling for feature prior regions generation and detection. But these operations may omit some effective features which can contribute to a better result, because the features in deeper convolutional layer correspond to wider reception fields, leading to a grosser granularity. To solve these problems, we combine the feature maps of multiple layers to enhance the RPN and RoI pooling, specifically by combining lower-level and higher-level features. In addition, we propose two strategies to replace the fully connected layers, achieving model reduction to satisfy the actual needs of fault detection including high accuracy and fast speed.

To enrich features, we introduce a MRPN to combine different level feature maps, as introduced in subsection IV-B. Then, a MRoI pooling is proposed to apply concatenation on each feature maps and encode the concatenated feature, as introduced in subsection IV-C. The convolutional (Conv) block in the concatenated feature is 576-d (RPN3_3 is 192-d, RPN4_3 is 192-d, and RPN5_3 is 192-d), and we attach a randomly initialized 512-d 1×1 convolutional layer for reducing dimension. Last, we introduce a model reduction scheme with accurate detection, as introduced in subsection IV-D.

B. Multi region proposal generation

Designing deep classifier modules on top of the feature extractor is as important as the extractor itself. Small anchors may miss some important details in the lower layer. In order to solve this problem, we can upsample higher-level feature maps to complement its lower counterpart. Inspired by GoogLeNet [40] and HyperNet [41], we propose a novel MRPN based on a key method named inception module. In the MRPN, each sliding position all associates a set of prior anchors that can generate fault region proposals, and there are multi-scale sliding windows over multi-level feature maps.

Because of sub-sampling as pooling operations in CNN, these feature maps are not the same resolution. To combine multi-level feature maps at the same resolution in the VGG16 model [39], different sampling strategies are designed for different layers. For the lower feature maps (Conv3_3), a 2×2 max pooling layer is added to carry out subsampling. We apply the 512-d 1×1 Conv layer over this pooling layer and concatenate Conv4_1 layer. The concatenated layer is then applied to another 2×2 max pooling layer to extract local features. For the middle layer Conv4_3, a 2×2 max pooling layer is first added to carry out subsampling. And then, we apply the 512-d 1×1 Conv layer over this pooling layer and concatenate the Conv5_1 layer to extract features. In the MRPN, a 192-d 1×1 Conv layer is applied to extract local features over above two processed feature maps and Conv5_3 with an additional non-linearity [42].

Moreover, we normalize multiple feature maps by using batch normalization (BN) [43] to generate RPN3_3, RPN4_3, as well as RPN5_3 layers and concatenate them to one single output cube. An illustration of MRPN is shown in the top part of Fig. 2. The MRPN has the following advantages:

- Inspired by neuroscience, multi-level reasoning has been proved to be useful in some computer vision problems [39].
- Deep, intermediate and shallow CNN are characterized by a complementary fault detection system, as shown in the experiments.

- All local features can be pre-computed before MRPN and detection modules, leading to no redundant computation.

Next, we encode above concatenated feature maps using a 3×3 Conv layer which not only extracts more semantic features but also compresses them into a uniform space. Finally, the 512-d concatenated feature vector is fed into two output layers: 1) a classification layer to predict the fault region score; 2) a regression layer to refine the position of each kind of prior fault region. Moreover, there are three scales (128, 256, and 512) of our prior bounding boxes and three aspect ratios (0.5, 1, and 2), while each sliding position has $N =$ 9 prior anchors. In the learning stage, when the Intersection of Union (IoU) overlap is greater than 0.5 with a ground-truth, we will set a positive label for the prior box, and set a background label when IoU overlap is less than 0.3. And then, we retain the highest score anchor by applying a linear Non-Maximum Suppression (NMS) [44] with a threshold 0.7, and rapidly suppress the lower scoring boxes in the neighborhood. In the end, the top-2000 candidate fault regions are selected to achieve multi-level detection network.

C. Multi-level RoI pooling

The past object detectors all only perform RoI pooling over the last Conv layer, such as SPP-Net [45] and FRCNN [13]. However, to make full use of the multi-level Conv features and enrich the discriminative information of each prior fault region, we implement MRoI pooling over the RPN3_3, RPN4_3, and Conv5_3 feature maps, and obtain two $512 \times H \times W$ pooled features (both H and W are set to 7 in practice). Next, we apply concatenation on each feature and encode concatenated feature with a 512×1×1 Conv layer to combine the multi-level features. The 1×1 Conv layer can learn fusion weights in the training process and reduce the dimensions to match the first Conv layer (*i.e.* Conv6) in the model reduction scheme (see following subsection IV-D for details). The multi-level weighted fusion feature is then accessed to the follow-up bounding box classification and regression model. An illustration of multi-level fault detection network is depicted in the bottom half of Fig. 2(a).

D. Model reduction scheme

For classification, the feature maps of the last Conv layer are vectorized and fed into fully connected (FC) layers followed by a softmax logistic regression layer [42]. This structure treats the Conv layers as feature extractors, and the resulting feature is classified in a traditional way. The simplest way to implement fault detection is to take the FC-Dropout-FC-Dropout pipeline. However, it is known that the FC may cause overfitting, and it requires millions of parameters, thus hampering the generalization ability of the overall network [39].

In this paper, we propose two strategies to replace FC layers, achieving model reduction with accurate detection. As shown in Fig. 3(a), one idea is to use the Conv layer to replace the FC. Instead of adding FC layers, we take the average of the feature maps of MRoI pooling with the 1×1 Conv layer. The incorporation of the 1×1 Conv with ReLU layer is a way to increase the nonlinearity of the decision function without affecting the receptive fields of the Conv layers. The dropout [46] ratio is set as 0.5, which randomly sets half of the activations of the Conv6/Conv7 to zero during training. It has improved the generalization ability and largely prevents overfitting.

Algorithm 1 A detection algorithm for real-time RVBS-FD of freight train (Light FTI-FDet).

```
1: Pre-train a deep CNN model (i.e. pre-trained VGG16 model on the ImageNet) for initializing basic layers in Step 2 and Step 3.
2: Train MRPN and use a linear NMS to remove the redundant region proposals.
3: Train a separate fault detection network using the region proposals obtained from Step 2.
4: Fine-tune the Conv layers unique to MRPN sharing the layers trained in Step 3.
5: Fine-tune the Conv layers for the fault detection network using region proposals obtained from Step 4, with shared the unique layers fixed.
6: Output the unified framework jointly trained in Step 4 and Step 5 as the final model.
```

Another idea is to utilize both the Conv layer and BN layer to replace the FC and dropout. In Fig. 3(b), the BN layer takes a step towards reducing internal covariate shift, and in doing so dramatically accelerates the training of our framework. It also acts as a regularizer, in some cases eliminating the need for dropout. Moreover, the BN makes it possible to use saturating nonlinearities by preventing the network from getting stuck in the saturated modes.

The MRPN and fault detection network are trained via back-propagation and stochastic gradient descent (SGD), as shown in **Algorithm 1**. We use a basic learning rate of 0.001 and it is divided by 10 for each 40K mini-batch until convergence. The batch sizes of MRPN and MRoI are set to 256 and 512, respectively. A pre-trained VGG16 model on the ImageNet [39] is first used to initialize shared Conv layers of our model, and then the new layers are initialized with a zero mean and a standard deviation of 0.01 Gaussian distribution. The momentum and weight decay are set as 0.9 and 0.0005, respectively. The confidence score in the detecting stage is 0.9.

V. Experiments

To evaluate the capability of our Light FTI-FDet in RVBS-FD of freight trains, we introduce a database containing five typical faults and evaluation metrics. And then, we analyze detection performance on different feature maps, model reduction scheme, and different modules connection. Finally, the experimental results are presented compared with state-of-the-art methods. All experiments are conducted on the PC condition of 3.60GHz of Intel Core i7-7700 processor, 16G RAM, and a single GTX1080Ti GPU.

A. Databases and evaluation metrics

A freight train image database [15], [19], [47] is used in this study, including five typical faults: angle cock, bogie block key, cut-out cock, dust collector, and fastening bolt on brake beam, as shown in Table I. The size of all images are fixed to 700×512 pixels. We label each training set image according to the format of the PASCAL VOC dataset [48] using a specialized label tool.

- *Angle cock*: As a key component of the air brake system in freight trains, the angle cock is used to keep air flowing smoothly in the main pipeline. If the angle cock is closed, it will cause a serious accident when the freight train is driving. When an angle cock has faults, its handle is usually damaged or lost.
- *Bogie block key*: It is a very small part, which is used to keep wheel sets from separating out of the bogie. We can detect the bogie block key by checking whether it is missing or not. However, it is difficult to find the triangle contour of the bogie block key, because it is relatively unique structure in complex background.
- *Cut-out cock*: It is a key part that cuts off the air from main reservoir to the brake pipe, which is used to shut down the brake pipe. Normally, the handle of cut-out cock is opened, which is not visible. When train stops or the brake breaks down, the handle of cut-out cock is closed. The handle is perpendicular to the vertical bar which is visible.
- *Dust collector*: It is often located beside the cut-out cock, and its function is to filter the impurities of compressed air. When a dust collector has faults, its end cover is usually damaged or lost.
- *Fastening bolt on brake beam*: It is an essential component of train brake device. If train brakes, the braking beam will create a great horizontal force, which may break fastening bolt or make it fall off.

In view of mechanical failure, these typical faults are caused by non-rigid mechanical structure. So the fault status is difficult to be described based on exact feature model, which becomes the primary difficulty of fault detection.

To evaluate the effectiveness of detectors, there are five indexes [15], [19]: correct detection rate (CDR), missing detection rate (MDR), false detection rate (FDR), detection speed, and model size (parameters). Taking an example to explain the above indexes, there is a testing set that contains m fault images and n non-fault images, through the work of the proposed method, a images are inspected as fault, among them b images are inspected by error, meanwhile, c images are inspected as non-fault, among them d images are inspected by error. Under the above situation, the above indexes will be defined as: MDR = $b/(m+n)$, FDR = $d/(m+n)$ and CDR = 1-MDR-FDR.

When we evaluate the proposed RVBS-FD, the index of FDR is less important than MDR, because it has little impact to detect non-fault region as error. The indexes of CDR, MDR, and FDR are all used to measure the accuracy of the detection algorithms. The model size and the accuracy report the impact of CNN architectural designs. We use the testing time for each image as the testing speed to reflect the dependence of a detector on hardware.

TABLE I: Databases using in experiments, including five typical faults: angle cock, bogie block key, brake shoe key, cut-out cock, dust collector, and fastening bolt on brake beam.

Databases	Training set	Testing set		
		Non-fault	Fault	Total
Angle cock	2002	1049	975	2024
Bogie block key	5440	2530	367	2897
Cut-out cock	815	671	179	850
Dust collector	815	798	52	850
Fastening bolt	1724	445	1257	1902

TABLE II: Detection results on different feature maps on the bogie block key.

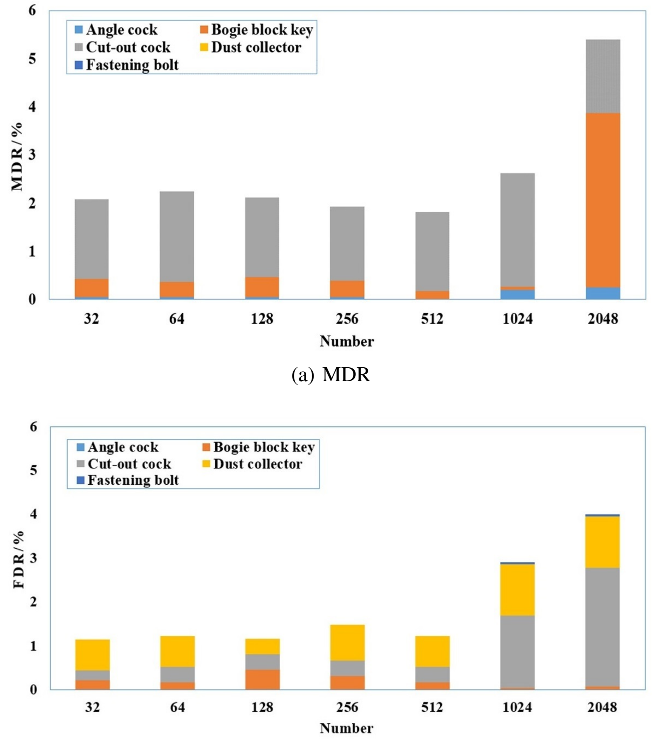
Feature maps	Channel	Bogie block key/%			Detection
		CDR↑	MDR↓	FDR↓	speed/s
5_3	512×1	98.96	0.59	0.45	0.051
5_3+3_3	256×2	99.04	0.72	0.24	0.053
5_3+4_3	256×2	99.35	0.10	0.55	0.054
5_3+4_3+3_3	192×3	98.90	0.72	0.38	0.057
5_3/2+4_3/2+3_3/2	192×6	98.45	0.76	0.79	0.060
5_3+5_1+4_1	192×3	99.66	0.17	0.17	0.057
Our method	192×6	99.86	0.07	0.07	0.058

B. Performance analysis

Different convolutional feature maps. An important property of our Light FTI-FDet is that it combines coarse-to-fine information across deep CNN models. We compare different feature maps assembled to obtain results on the bogie block key to illustrate the superiority of the MRPN and MRoI pooling modules. The detection results for combining different feature maps are shown in Table II. The "5_3" and "512×1" mean that we apply a $512\times1\times1$ Conv layer to extract features over Conv5_3 layer for the MRPN and MRoI pooling. The "5_3/2+4_3/2+3_3/2" and "192×6" mean that we first concatenate adjacent Conv layers such as Conv5_3 and Conv5_2, and then we use the same sampling strategies as introduced in subsection IV-B to match the same resolution, followed by applying six $192\times1\times1$ Conv layer to extract features for the MRPN and MRoI pooling. For fairness, the feature maps are normalized to the same resolution and all networks are trained with the same configuration.

Table II shows the detection results for combining different feature maps. Unsurprisingly, the results represent that the proposed MRPN and MRoI pooling work best (CDR = 99.86%, MDR = 0.07%, and FDR = 0.07%). This result indicates two keys. Multi-layer combination performs toughly better than a single layer, both for proposal and detection. There is no redundant computation on the MRPN and MRoI pooling. These results also demonstrate the effectiveness of the low-to-high combination strategy.

Model reduction scheme. In the proposed model reduction scheme, the performance of our Light FTI-FDet is greatly



(a) MDR

(b) FDR

Fig. 4: Sensitivity analysis on five faults across different number of channels in model reduction scheme. The optimal number of convolutional layers in model reduction scheme is set as 512, which is full in consideration of precision and complexity of our method.

affected by the width of Conv layer (*i.e.* the number of channels). Moreover, the number of channels is generally different for varieties of faults in practice. To achieve good results, we compare different channel number (32-2048) on above five faults to find an optimal number. For fairness, we just simply change the number of channels. And all networks are trained with the same configuration.

For the convenience of comparative analysis, we simply add up MDR or FDR to indicate the impact of the channel numbers on accuracy. As shown in Fig. 4, our method is robust to the change of the channel number for the faults of angle cock, dust collector and fastening bolts. However, there is a sharp difference within these results of cut-out cock and bogie block key. For the cut-out cock, its FDR increases slowly with the increase of the channel number, while its MDR rises abruptly when the channel number exceeds 1024. And for the bogie block key, its MDR and FDR are lowest when the channel number is 1024. Moreover, the sum of the MDR is minimal when the channel number is 512. Therefore, the optimal channel number of Conv layers in model reduction scheme is set as 512, which is full in consideration of precision and complexity of our Light FTI-FDet.

Different modules. We analyze each module used in our Light FTI-FDet by making experiments on bogie block key. In Table III, the "Baseline" means MRPN with MRoI pooling modules, and the "MRS" means model reduction scheme. Our method uses MRPN and MRoI pooling to learn more effective and comprehensive features than the Conv5_3 for distinguishing faults from complex backgrounds. The Baseline+MR can achieve the best performance. The model reduction scheme can significantly reduce model size without loss of performance. For example, the model size is reduced from 569.3MB to 89.7MB, and the CDR is increased from 98.07% to 99.86%. In addition, it can be seen from the results that using BN layer has higher accuracy than using Dropout layer with the same model size. So, we apply the Conv layer to replace the FC in the model reduction scheme.

TABLE III: Detection results of connecting different modules. The "Baseline" means MRPN with MRoI pooling modules, the "FC" means fully connected layer and the "MRS" means model reduction module.

Modules	Bogie block key/%			Model
	CDR↑	MDR↓	FDR↓	size/MB
Conv5_3+FC	95.66	4.24	0.10	521
Conv5_3+MRS(BN)	96.76	1.29	1.95	63.9
Conv5_3+MRS(Dropout)	98.96	0.59	0.45	63.9
Baseline+FC	98.07	0.10	1.83	569.3
Baseline+MRS(BN)	99.45	0.24	0.31	89.7
Baseline+MRS(Dropout)	99.86	0.07	0.07	89.7

C. Comparison with state-of-the-art methods

To further show the advantages of our Light FTI-FDet, we conduct experiments using hand-crafted feature descriptors and conventional machine learning tools. We compare the performance of the proposed Light FTI-FDet with current state-of-the-art methods and our previous research FTI-FDet [15]. In these methods, FAMRF+EHF [19], cascade detector based on local binary patterns (LBP) [19], HOG+Adaboost+SVM [49] are traditional methods. The SSD [21], YOLOv3 [20], RefineDet [22], RON [23], and DSOD [27] are one-stage object detectors in deep learning. The FRCNN [13], MLKP [35], CoupleNet [37], R-FCN [14], and Cascade RCNN [36] are two-stage object detectors. The same images are trained for fault detection. We conduct our experiments using the publicly available Caffe [50], which is well-known in this community. The same images are trained for fault detection. The parameters in each detector are tuned to be the best performance.

Detection results of five typical faults in comparison with state-of-the-art methods are detailed in Table IV. For the convenience of comparative analysis, we simply calculate the mean values of CDR, MDR, and FDR to indicate the accuracy of different methods, denoted as mCDR, mMDR, and mFDR, respectively. As shown in Table IV, for the angle cock and fastening bolt, our method has 100% CDR, 0% MDR, and 0% FDR. For the bogie block key, our method has 99.86% CDR, 0.07% MDR, and 0.07% FDR. The proposed Light FTI-FDet outperforms other related methods and FTI-FDet [15] in the above three faults.

TABLE IV: Detection results of five typical faults in comparison with state-of-the-art methods using the indexes of CDR, MDR, and FDR. We compare the proposed Light FTI-FDet with some widely used detectors such as traditional detectors, one-stage object detectors, and two-stage object detectors. The same images are trained for fault detection.

Methods	Angle cock/%			Bogie block key/%			Cut-out cock/%			Dust collector/%			Fastening bolt/%		
	CDR↑	MDR↓	FDR↓	CDR↑	MDR↓	FDR↓	CDR↑	MDR↓	FDR↓	CDR↑	MDR↓	FDR↓	CDR↑	MDR↓	FDR↓
Cascade detector(LBP)	82.96	16.56	0.48	96.58	2.11	1.31	76.83	7.88	15.29	89.30	1.88	8.82	92.06	3.21	4.73
HOG+Adaboost+SVM	89.02	10.88	0.10	96.96	0.90	2.14	88.00	2.59	9.41	96.94	0.47	2.59	95.69	1.42	2.89
FAMRF+EHF	–	–	–	97.72	0.76	1.52	93.30	1.29	5.41	96.12	1.06	2.82	92.70	0.89	6.41
SSD(VGG16)	100	0	0	99.28	0.55	0.17	90.82	0.94	8.24	91.65	0	8.35	96.27	3.73	0
SSD(ResNet101)	96.19	3.41	0.40	91.81	0.35	7.84	95.53	4.47	0	84.47	0	15.53	98.68	0.58	0.74
YOLOv3	99.65	0.31	0.05	89.09	10.80	0.10	82.12	0.35	17.53	68.23	0	31.76	96.32	3.68	0
RefineDet(VGG16)	100	0	0	97.48	2.31	0.21	94.24	1.41	4.35	85.65	0	14.35	99.32	0.68	0
RON(VGG16)	99.93	0.07	0	98.31	1.62	0.07	92.47	1.06	6.47	98.59	0	1.41	99.95	0	0.05
DSOD(DenseNet)	96.59	1.24	2.17	98.66	0.79	0.55	90.12	9.88	0	99.76	0.12	0.12	97.95	0.47	1.58
MLKP(VGG16)	100	0	0	96.44	3.49	0.07	95.88	0.47	3.65	97.29	0	2.71	99.95	0.05	0
CascadeRCNN(VGG16)	99.95	0.05	0	94.34	4.56	1.10	98.12	1.41	0.47	89.65	0	10.35	99.84	0.16	0
FRCNN(VGG16)	99.75	0.25	0	95.66	4.24	0.10	97.65	0.94	1.41	96.35	0	3.65	99.95	0.05	0
+Soft NMS	99.75	0.25	0	77.98	22.02	0	98.35	0.83	0.82	95.88	0	4.12	99.90	0.05	0.05
R-FCN(ResNet-50)	99.85	0.15	0	96.20	3.59	0.21	96.47	0.82	2.71	80.59	0	19.41	99.95	0	0.05
+Soft NMS	99.75	0.25	0	64.42	35.55	0.03	70.00	0.12	29.88	73.18	0	26.82	99.74	0.26	0
R-FCN(ResNet101)	99.71	0.29	0	90.99	9.01	0	86.00	0.71	13.29	91.65	0	8.35	99.97	0	0.03
MLKP(ResNet101)	100	0	0	93.41	6.42	0.17	94.59	1.06	4.35	81.18	0	18.82	99.79	0.05	0.16
CascadeRCNN(ResNet101)	99.95	0.05	0	97.51	2.28	0.21	95.41	2.94	1.65	91.88	0	8.12	99.74	0.05	0.21
FRCNN(ResNet-101)	100	0	0	98.76	1.14	0.10	96.83	0.82	2.35	87.41	0	12.59	99.95	0	0.05
CoupleNet(ResNet101)	100	0	0	98.03	1.24	0.73	99.41	0	0.59	87.88	0	12.12	99.87	0.13	0
+Soft NMS	100	0	0	98.17	0.90	0.93	98.82	0	1.18	86.59	0	13.41	99.92	0	0.08
FTI-FDet(VGG16)	99.26	0.25	0.49	98.76	1.24	0	98.71	0.82	0.47	99.65	0	0.35	100	0	0
Light FTI-FDet	100	0	0	99.86	0.07	0.07	96.24	1.88	1.88	99.53	0	0.47	100	0	0

For the cut-out cock, our Light FTI-FDet has 96.24% CDR, 1.88% MDR, and 1.88% FDR. For the dust collector, our Light FTI-FDet has 99.53% CDR, 0% MDR, and 0.47% FDR. The accuracy of our method is lower than FTI-FDet in these two faults. In Table V, the mFDR of our method is higher than traditional methods and object detectors, and is slightly lower than FTI-FDet. The main reason is that the processing of replacing FC by Conv layer may be insensitive to noise. Some typical detection results are shown in Fig. 5. However, it can be seen from the results that our method is unsatisfactory when the images are over-imposed or the contaminated by noises. The reason is that these types of training images are limited in our training set. We plan to solve this problem by expanding the database through adding more images of such kinds through data augmentation in the future. This will also improve the generalization ability of the CNN network.

The mMDR of RCNN-based method is high, especially FRCNN and R-FCN for the bogie block key, and the reason is that the bogie block key is hard to be compared with other components of the freight train because of its small size. Traditional FRCNN and R-FCN perform the RoI pooling only on the final layer to obtain features of the region, resulting in omitting some potentially useful features. But our method uses MRPN and MRoI pooling to also utilize the features from lower layer, which makes the obtained features more effective and comprehensive, and can better help detect faults from complex backgrounds.

D. Comparison of model size and detection speed

Table VI lists a comparison of the model size and detection speed in deep learning. The model size of the proposed Light FTI-FDet is reduced to 89.7 MB, achieving about 83% model size reduction based on the VGG16 compared with FRCNN. To perform a freight train image with the size of 700×512 pixels, the testing time of our method is 0.058s with a single GPU. The detection speeds of one-stage object detectors are almost faster than our Light FTI-FDet, but their model sizes are larger relatively. Especially, the model size of DSOD is smallest, but its speed is slower than ours. In addition, both speed and model sizes of two-stage object detectors are unsatisfactory compared with our Light FTI-FDet. Therefore, we can conclude from the experiment results that our method is more capable for real-time fault detection of freight train images, which has detection speed competitive with the one-stage object detectors, and its model size is much smaller than two-stage object detectors.

To sum up, there are three reasons that make the proposed Light FTI-FDet in RVBS-FD for freight trains achieve high accuracy and fast speed. First, the proposed MRPN is able to implement more accurate fault region proposals. Second, the MRoI pooling can help the fault detection network to synthesize more comprehensive information for distinguishing faults from backgrounds. Third, the model reduction scheme is used to reduce redundant parameters, improve the generalization ability and prevent overfitting.

TABLE V: Mean detection accuracy of five typical faults. The mean values of CDR, MDR, and FDR are calculated as mCDR, mMDR, and mFDR to indicate the accuracy of different methods.

Methods	mCDR/%	mMDR/%	mFDR/%
Cascade detector(LBP)	87.55	6.33	6.12
HOG+Adaboost+SVM	93.32	3.25	3.43
FAMRF+EHF	94.96	1.00	4.04
SSD(VGG16)	95.60	1.05	3.35
SSD(ResNet101)	93.34	1.76	4.90
YOLOv3	87.08	3.03	9.89
RefineDet(VGG16)	95.34	0.88	3.78
RON(VGG16)	97.85	0.56	1.59
DSOD(DenseNet)	96.62	2.50	0.88
MLKP(VGG16)	97.92	0.80	1.28
CascadeRCNN(VGG16)	96.37	1.24	2.39
FRCNN(VGG16)	97.89	1.10	1.01
+Soft NMS	94.37	4.63	1.00
R-FCN(ResNet-50)	94.62	0.91	4.47
+Soft NMS	81.41	7.24	11.35
R-FCN(ResNet101)	93.66	2.00	4.33
MLKP(ResNet101)	93.79	1.51	4.70
CascadeRCNN(ResNet101)	96.14	0.92	2.94
FRCNN(ResNet-101)	96.59	0.40	3.01
CoupleNet(ResNet101)	97.04	0.25	2.71
+Soft NMS	96.70	0.18	3.12
FTI-FDet(VGG16)	99.28	0.46	0.26
Light FTI-FDet	99.13	0.39	0.48

VI. Conclusion

The RVBS-FD for freight trains can greatly reduce labor burden and improve the efficiency, especially with the development of machine vision technology in the IM field. In this paper, we present a real-time, light-weight and accurate detector named as Light FTI-FDet in the RVBS-FD of freight trains. Our Light FTI-FDet has a powerful ability for high quality fault proposal generation by using MRPN with a set of characteristic prior anchors, and achieves fault classification and accurate localization via a fault detection network including a MRoI pooling and a model reduction scheme. Specially, the reliable model reduction scheme is used to obtain fast detection speed with high accuracy in a simple manner. Experiments on five benchmarks show that the our RVBS-FD has a low resource requirement with a fast speed at 0.058s per image, achieving only 17% model size of Faster R-CNN.

In the future, we will focus on the following three aspects to further improve accuracy and computation speed of the fault detection. First, the accuracy of RVBS-FD should be further improved especially on the cut-out cock. There are some methods such as increasing training samples and designing a better strategy for fusing multi-level features. Second, the detection speed of RVBS-FD can be further increased by simplifying the structure of the proposed MRPN and MRoI pooling. Third, the model size of our Light FTI-FDet can be further decreased using a light-weight network to replace the

TABLE VI: Comparison results of model size and detection speed in deep learning. By model size, we mean the number of bytes required to store all of the parameters in the trained model.

Methods	Model size/MB	Detection speed/s
SSD(VGG16)	95.5	0.047
SSD(ResNet101)	194.4	0.094
YOLOv3	246.3	0.026
RefineDet(VGG16)	135.8	0.056
RON(VGG16)	157.9	0.029
DSOD(DenseNet)	50.8	0.097
MLKP(VGG16)	596.1	0.147
CascadeRCNN(VGG16)	735.6	0.195
FRCNN(VGG16)	546.8	0.065
R-FCN(ResNet-50)	123.7	0.079
R-FCN(ResNet-101)	199.9	0.116
MLKP(ResNet-101)	253.5	0.271
CascadeRCNN(ResNet101)	220.8	0.203
FRCNN(ResNet-101)	179.7	0.166
CoupleNet(ResNet101)	409.6	0.112
FTI-FDet(VGG16)	557.3	0.071
Light FTI-FDet	89.7	0.058

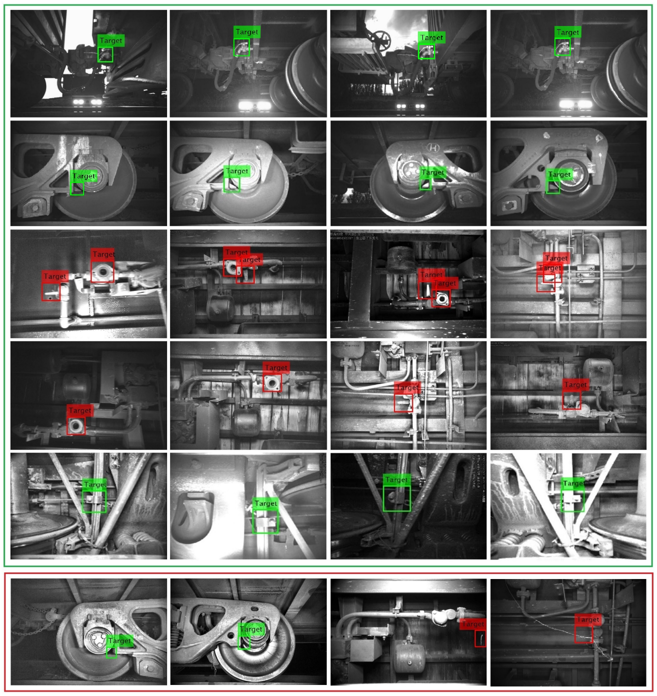


Fig. 5: Qualitative results of our method. The first five rows are the correct detection results. The bottom row is the false results. Our Light FTI-FDet is unsatisfactory for the robustness of noise and the disturbance from other similar structures without faults.

baseline (VGG model).

References

[1] Z. Hui, X. Jin, Q. M. J. Wu, Y. Wang, and Y. Yang, "Automatic visual detection system of railway surface defects with curvature filter and improved gaussian mixture model," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 7, pp. 1–16, 2018.

[2] H. Feng, Z. Jiang, F. Xie, P. Yang, J. Shi, and L. Chen, "Automatic fastener classification and defect detection in vision-based railway inspection systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 4, pp. 877–888, 2014.

[3] L. Liu, F. Zhou, and Y. He, "Automated visual inspection system for bogie block key under complex freight train environment," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 1, pp. 2–14, 2015.

[4] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: the rising trend of vision based measurement," *IEEE Instrumentation and Measurement Magazine*, vol. 17, no. 3, pp. 41–47, 2014.

[5] L. Liu, F. Zhou, and Y. He, "Vision-based fault inspection of small mechanical components for train safety," *IET Intelligent Transport Systems*, vol. 10, no. 2, pp. 130–139, 2016.

[6] F. Zhou, R. Zou, and H. Gao, "Dust collector localization in trouble of moving freight car detection system," *Frontiers of Information Technology and Electronic Engineering*, vol. 14, no. 2, pp. 98–106, 2013.

[7] C. Zheng and Z. Wei, "Automatic online vision-based inspection system of coupler yoke for freight trains," *Journal of Electronic Imaging*, vol. 25, no. 6, p. 061602, 2016.

[8] F. Zhou, R. Zou, Y. Qiu, and H. Gao, "Automated visual inspection of angle cocks during train operation," *Proceedings of the Institution of Mechanical Engineers Part F Journal of Rail and Rapid Transit*, vol. 228, no. 7, pp. 794–806, 2013.

[9] L. Liu, F. Zhou, and Y. He, "Automated status inspection of fastening bolts on freight trains using a machine vision approach," *Proceedings of the Institution of Mechanical Engineers Part F Journal of Rail and Rapid Transit*, vol. 230, no. 7, pp. 1159–1166, 2015.

[10] H. C. Kim and W. Y. Kim, "Automated inspection system for rolling stock brake shoes," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 8, pp. 2835–2847, 2011.

[11] J. Chen, Z. Liu, H. Wang, A. Nunez, and Z. Han, "Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 2, pp. 257–259, 2018.

[12] J. Sun, Z. Xiao, and Y. Xie, "Automatic multi-fault recognition in tfds based on convolutional neural network," *Neurocomputing*, vol. 222, pp. 127–136, 2017.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[14] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: object detection via region-based fully convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 379–387.

[15] Y. Zhang, K. Lin, H. Zhang, Y. Guo, and G. Sun, "A unified framework for fault detection of freight train images under complex environment," in *Proceedings of the IEEE International Conference on Image Processing*, 2018, pp. 1348–1352.

[16] G. Nan and Y. Gao, "Automated visual inspection of multipattern train components using gradient information and feature fusion under the illumination-variant condition," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, vol. 232, no. 5, pp. 1500–1513, 2018.

[17] Y. Cao, B. Zhang, J. Liu, and J. Ma, "Weighted margin sparse embedded classifier for brake cylinder detection," *Neurocomputing*, vol. 120, no. 10, pp. 560–568, 2013.

[18] L. Nan, Z. Wei, and Z. Cao, "Automatic fault recognition for brake-shoe-key losing of freight train," *Optik - International Journal for Light and Electron Optics*, vol. 126, no. 23, pp. 4735–4742, 2015.

[19] G. Sun, Y. Zhang, H. Tang, H. Zhang, M. Liu, and D. Zhao, "Railway equipment detection using exact height function shape descriptor based on fast adaptive markov random field," *Optical Engineering*, vol. 57, no. 5, p. 053114, 2018.

[20] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[22] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4203–4212.

[23] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5244–5252.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, Oct 2017, pp. 2980–2988.

[25] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750.

[26] S. Liu, D. Huang, and Y. Wang, "Receptive field block net for accurate and fast object detection," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 385–400.

[27] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "DSOD: learning deeply supervised object detectors from scratch," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1919–1927.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[30] R. B. Girshick, "Fast R-CNN," in *Proceedings of the International Conference on Computer Vision*, 2015, pp. 1440–1448.

[31] A. Shrivastava, A. Gupta, and R. B. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[32] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: towards accurate region proposal generation and joint object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 845–853.

[33] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 354–370.

[34] T. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 936–944.

[35] H. Wang, Q. Wang, M. Gao, P. Li, and W. Zuo, "Multi-scale location-aware kernel representation for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1248–1257.

[36] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6154–6162.

[37] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu, "CoupleNet: coupling global structure with local parts for object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4126–4134.

[38] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-Head R-CNN: In defense of two-stage object detector," *arXiv preprint arXiv: 1711.07264*, 2017.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, 2015.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[41] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: towards accurate region proposal generation and joint object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 845–853.

[42] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proceedings of the International Conference on Learning Representations*, 2014.

[43] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, 2015, pp. 448–456.

[44] N. Bodla, B. Singh, C. Rama, and L. Davis, "Soft-NMS – Improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5562–5570.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[47] Y. Zhang, "Hierarchical feature matching of fault images in tfds based on improved markov random field and exact height function," Master's thesis, Hubei University of Technology, 2017.

[48] M. Everingham, S. Eslami, L. Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[49] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[50] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 675–678.

Yang Zhang received the B.S. degree and the M.S. degree from Hubei University of Technology, Wuhan, China, in 2014 and 2017. He is currently pursuing the Ph.D. degree in the National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing, China.

His current research interests are machine learning and computer vision.

Moyun Liu received the B.S. degree from Hubei University of Technology, Wuhan, China, in 2019. He is currently pursuing the M.S. degree in the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China.

His current research interests are computer vision and image processing.

Yunian Chen received the B.S. degree from Jilin University, Changchun, China, in 2013 and 2017. She is currently pursuing the M.S. degree in the National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing, China.

Her current research interests are machine learning and computer vision.

Hongjie Zhang is working toward the PhD degree in the National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing, China.

His research interests include computer vision and graphics, digital image processing, and pattern recognition.

Yanwen Guo received the PhD degree in applied mathematics from the State Key Lab of CAD & CG, Zhejiang University, China, in 2006. He was a visiting scholar in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, from 2013 to 2015.

He is currently a full professor in the National Key Lab for Novel Software Technology and the Department of Computer Science and Technology at Nanjing University. His research interests include image and video processing, vision, and Computer Graphics.