A Looseness Detection Method for Railway Catenary Fasteners based on Reinforcement Learning Refined Localization

Zhong, Junping; Liu, Zhigang; Wang, H.; Liu, Wenqiang; Yang, Cheng; Han, Zhiwei; Nunez, Alfredo

DOI
10.1109/TIM.2021.3086913

Publication date
2021

Document Version
Final published version

Published in
IEEE Transactions on Instrumentation and Measurement

A Looseness Detection Method for Railway Catenary Fasteners Based on Reinforcement Learning Refined Localization

Junping Zhong, *Student Member, IEEE*, Zhigang Liu, *Senior Member, IEEE*, Hongrui Wang, *Member, IEEE*, Wenqiang Liu, *Graduate Student Member, IEEE*, Cheng Yang, Zhiwei Han, *Member, IEEE*, and Alfredo Núñez, *Senior Member, IEEE*

***Abstract*— Brace sleeve (BS) fasteners, i.e., nut and bolt, are small components but play essential roles in fixing BS and cantilever in railway catenary system. They are commonly inspected by onboard cameras using computer vision to ensure the safety of railway operation. However, most BS fasteners cannot be directly localized because they are too small in the inspection images. Instead, the BS is first localized for detecting the BS fastener. This leads to a new problem that the localized BS boxes may not contain the complete BS fasteners due to low localization accuracy, making it infeasible to further diagnose the fastener conditions. To tackle this problem, this article proposes a novel pipeline for BS fastener looseness diagnosis. First, the competitive deep learning model Faster RCNN ResNet101 is used to coarsely localize BSs. Second, an action-driven reinforcement learning agent is adopted to refine the coarse-localized boxes through a dynamic position searching process. Then, BS fasteners are extracted from the refined localized BS image by the deep segmentation model YOLACT++, which is fast and interpretable. Finally, a looseness diagnosis criterion based on segmented information are proposed. We evaluate the performance of submodels independently and the overall performance of the whole model on a real-life catenary image dataset collected from a high-speed line in China. The test results show that the proposed method is effective for BS looseness detection in railway catenary.**

***Index Terms*— Component segmentation, looseness detection, railway catenary fasteners, reinforcement learning (RL).**

I. Introduction

CATENARY systems are important infrastructures that support the electric power transmission in railway power supply system. As a key component in catenary, fasteners installed on the brace sleeve (BS) are small but plays an important role in binding together the BS and a cantilever. Due to the physical/mechanical impact triggered by the high-speed trains and complex environmental influences along the railway line, the fastener may become loose after a long-term operation, as shown in Fig. 1.

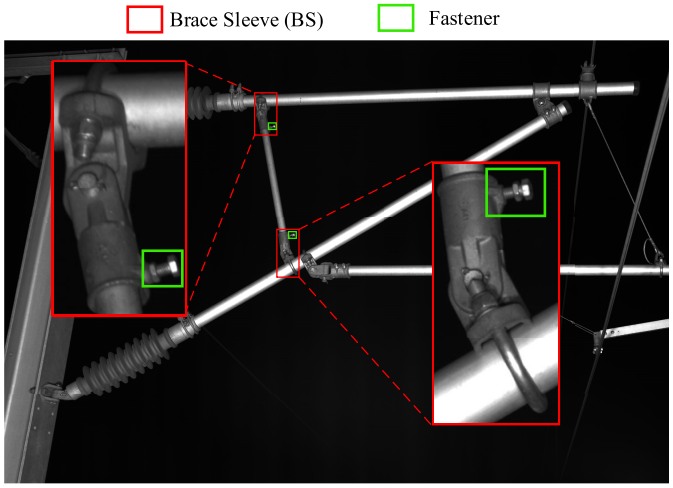


Fig. 1. Normal (left) and loose (right) fasteners in the global catenary image.

The loose fasteners increase the risk of disrupting train services and compromising operation safety. Therefore, it is essential to monitor the fastener conditions. To automatically monitor the catenary components, image acquisition and processing methods have been developed to replace manual inspection [1]. The first step of this technique is localizing components from a global catenary image captured from train inspections. Then, defect diagnosis is implemented based on the specific features associated with the localized components. For BS fasteners, localization is very difficult because fasteners are very small among all components. But, diagnosing fastener conditions must rely on an accurate localization result.

In the past decade, class-agnostic object localization methods are mainly based on traditional handcrafted features [2], [3] and deep learning techniques [4]–[6]. They have been widely used for railway components localization. In terms of traditional methods, Han *et al.* [7] utilized histogram of oriented gradient (HOG) features to represent a series of sliding window images, which are sent to support vec-

Manuscript received January 19, 2021; revised May 17, 2021; accepted May 26, 2021. Date of publication June 7, 2021; date of current version June 18, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant U51977182 and in part by the Science and Technology Innovation Talents of Sichuan Science and Technology Plan under Grant 2021JDRC0008. The Associate Editor coordinating the review process was Qiang Miao. *(Corresponding author: Zhiwei Han.)*

Junping Zhong, Zhigang Liu, Wenqiang Liu, Cheng Yang, and Zhiwei Han are with the School of Electrical Engineering, Southwest Jiaotong University, Chengdu 611756, China (e-mail: zhongjunping@my.swjtu.edu.cn; liuzg_cd@126.com; liuwq_2009@126.com; yangc@my.swjtu.edu.cn; zw.han@my.swjtu.edu.cn).

Hongrui Wang and Alfredo Núñez are with the Section of Railway Engineering, Delft University of Technology, 2628 Delft, The Netherlands (e-mail: h.wang-8@tudelft.nl; a.a.nunezvicencio@tudelft.nl).

Digital Object Identifier 10.1109/TIM.2021.3086913

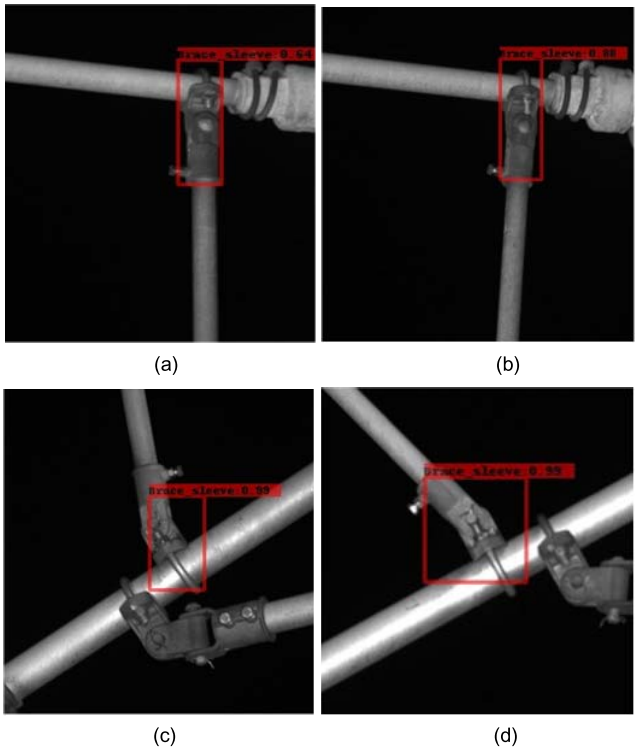


Fig. 2. (a)–(d) Examples of fasteners that are not included in the localized BS boxes produced by Faster RCNN ResNet101.

tor machine to localize catenary clevis. Zhong *et al.* [8] applied template matching on a standard catenary sleeve image and an original image to search the target position based on scale invariant feature transform (SIFT). Fan *et al.* [9] proposed a line local binary patterns (LBP) encoding method to localize fasteners on the railway track. Recently, deep learning models have shown great power in railway component localization. Kang *et al.* [10] applied Faster RCNN VGG16 to localize multiple-class components, such as isoelectric line, steady arm base, and insulator. Chang *et al.* [11] proposed refine-inception net (RIN) which integrates three novel feature enhancers on RefineDet [12] to maximize the ability of expressing deep features. The RIN greatly improves localization accuracy for small lassoes on the train. In [13], an improved YOLOv3 [14] which adopts a deblur block to enhance image quality is proposed to localize catenary split pins. Liu *et al.* [15] proposed an improved Faster RCNN ResNet101 [16] to localize BS fastener directly and increased its localization accuracy from 0.49 to 0.58, which is still too low compared with larger components and for application requirements.

Overall, traditional handcrafted feature-based methods are simpler, but the performance of deep learning-based methods is by far superior for detecting catenary components. As BS fastener is too small, localizing complete BSs first is considered the optimal choice. However, even the state-of-the-art deep learning methods [15], [16] may provide incorrect localizations by failing to include BS fasteners, which makes it infeasible to further diagnose the fastener condition, as shown in Fig. 2.

To address the localization problem shown in Fig. 2 and make the localized box more accurate, a reasonable solution is finding a box searching model that can change the position and the shape of an incorrect box, and make it get close to ground-truth. This article adopts a reinforcement learning (RL)-based solution. The RL refers to a broad group of learning techniques. RL agent emulates the way living beings learn by trying actions and learning from successes and failures. As shown in Fig. 3, an RL agent is trained to make good decisions in a given environment by receiving rewards when the decisions are considered positive. The agent observes the state of a given environment, and takes actions that transform the environment into a new state according to its state-action policy, which is learned during training. A Markov decision process (MDP) is a formal mathematical representation of how the agent interacts with the environment to learn its policy. As the dynamic box searching process follows MDP, thus training a box searching model can be realized by RL. Actually, the box searching model is a kind of agents, and training the box searching model is a kind of RL optimization problems. By properly defining the basic RL elements (Actions, Rewards, Environment, State, and Agent) according to the requirement of localization refinement, the obtained RL agent can be used as a box searching model. Recent works [17]–[19] in the RL field have proposed to combine deep neural networks with RL algorithms such as value function or policy function. For computer vision problems, various methods have been proposed in the literature. In [22], RL was adopted to learn a policy of selecting a region from five fixed subregions, and realize object localization by only a few steps. So far, we are not aware of available literature applying RL to solve catenary component localization problems. Caicedo and Lazebnik [20] proposed an active class-specific localization approach. Yun *et al.* [21] proposed an action decision method for object tracking by RL. Both works [20], [21] adopt action-driven RL that defines Actions as box transformations that can directly and intuitively present the box changing process. In this article, we select action-driven RL for localization refinement because it can be intuitively explained. As far as we know, this article is the first work that introduces RL for railway catenary localization refinement.

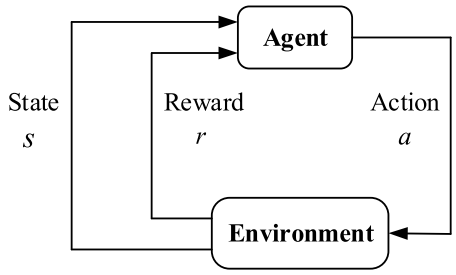


Fig. 3. Schematic flow of RL system.

For BS fastener defect detection, there is scarcely any related work that quantifies the defect state. In this article, we propose a diagnosis method which uses image segmentation masks to characterize fastener state for defect detection.

Image segmentation can be regarded as a binary classification task of pixels. In recent years, tremendous progress has been made on image segmentation with the development

of deep learning. Ronneberger *et al.* [23] designed U-net which adopts a contracting path and a symmetric expanding path to segment objects precisely. He *et al.* [24] proposed Mask R-CNN which uses feature pyramid network (FPN) to extract object features and predicts mask within the detected object box. Bolya *et al.* [26] proposed a real-time instance segmentation YOACLT, which uses one-stage architecture and remove the feature repooling step to enhance the segmentation speed. Based on YOLACT, Bolya *et al.* [27] proposed YOLACT++ which applied deformable convolutions and mask rescoring modules to improve the segmentation accuracy. For image segmentation in railway, the recent works [28], [13] have utilized deep segmentation models for rail surface defect detection and catenary split pin defect diagnosis.

In summary, the following issues from existing methods need to be addressed for BS fastener monitoring.

1) Localization accuracy for BS is not sufficiently high. When a part of or an entire fastener is missing in the localized image, it is impossible to perform fastener diagnosis.
2) Lack of effective fastener diagnosis method. Due to the inaccurate localization of fasteners and the lacking of defective samples, there are very few methods developed for BS fastener looseness detection.

In a previous study, Zhong *et al.* [29] investigated the use of RL for BS localization. This work is an extension that aims to address the aforementioned problems. We summarize the contributions of this article as follows.

1) A complete two-stage defect-detection pipeline is proposed for BS fasteners. The performance comparisons among different pipelines on a real-life catenary image dataset show that our method is effective.
2) For BS localization, an action-driven RL method is adopted to refine the coarse localized box automatically. We demonstrated that RL can effectively refine both Gauss initialized box and deep learning model produced box.
3) For BS fastener diagnosis, the deep learning segmentation model YOLACT++ is first utilized to extract the masks of fasteners. Then, a diagnosis criterion is proposed based on the characteristics of segmented masks for detecting the looseness of BS fasteners. Comparisons with existing methods verified the effectiveness of YOLACT++ and the proposed diagnosis criterion.

The rest of this article is organized as follows. Section II gives an overview of the proposed method. Section III introduces the action-driven RL model and describes how it localize the BSs from coarse to fine. Section IV presents the deep segmentation model for fastener looseness detection. Section V presents the experimental results and evaluates the performance of our method. Finally, conclusion and further works are summarized in Section VI.

II. Overview of the Proposed Method

The flowchart of the proposed defect detection method for BS fasteners is shown in Fig. 4. The input is a global catenary image which has a size of 6600 × 4400. It will go through two stages successively. In the first stage, the current state-of-the-art method Faster RCNN ResNet101 [15] is adopted to localize BSs coarsely. The localized BS may not include a fastener, as in the white boxes shown in Fig. 4. Then, a RL trained agent called action decision network (ADNET) is applied to refine the coarse boxes automatically. The RL agent takes a sequence of actions to adjust the coarse BS boxes close to their ground-truth positions. These actions are defined as a series of moving and shape changing transformations that will be introduced in Section III. The refinement process follows the box changes from white to purple in Fig. 4. In the second stage, the pixel-level segmentation model YOLACT++ [13] is utilized to extract the sleeve, nut, and bolt accurately. As the mask information of the extracted components can well characterize fastener conditions, an effective defect diagnosis criterion is proposed accordingly. Finally, BS fastener defect can be detected based on the proposed criterion.

III. Localizing BS Components Form Coarse to Fine

In this article, we consider the existing object localization method as a coarse prior step, and adopt an action-driven RL agent act as a generic postprocess step to improve the localization accuracy. In the following, we first briefly introduce the existing Faster RCNN ResNet101. Then, the adopted action-driven RL is elaborated in detail.

A. Coarse Localization by Faster RCNN ResNet101

The Faster RCNN [5] is a classic deep learning framework for object localization. It consists of three parts, namely, convolutional neural network (CNN) backbone, region proposal head, and object localization head. The CNN backbone extracts image features which greatly affects the performance of the whole framework. Many networks such as ZF, VGG, and ResNet can be used as the CNN backbone. Particularly, the ResNet101 [16] has a very deep structure with residual blocks, which makes ResNet101 learn discriminative features for localization. In [15], the Faster RCNN with ResNet101 has shown better localization capability than other structures. However, it still can produce inaccurate localization results when the object is too small, as shown in Fig. 2. These coarse BS boxes do not include the target fasteners and must be further improved.

B. Localization Refined by Action-Driven RL

Motivated by the reward-action in RL and [21], we consider the localization refinement problem as a control problem where a sequence of steps to refine the geometry of the localization box is obtained. Then, the refinement becomes a MDP that can be trained with RL. We define the actions as position-moving, scale-changing, and shape-changing. The reward is feedback about how well the current localization is compared with the ground truth. Therefore, the action-decision policy can be learned according to the obtained rewards. The agent is a deep CNN called ADNET, as shown in Fig. 5.

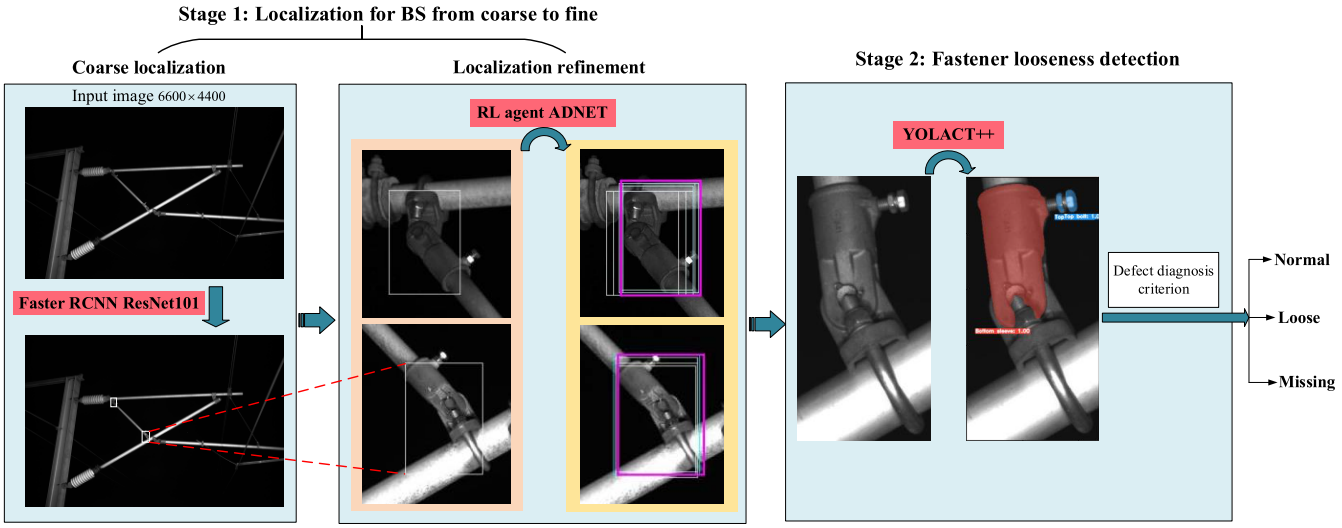


Fig. 4. Flowchart of the proposed method. Stage one: BS components are coarsely localized by generic localization methods (here, state-of-the-art method Faster RCNN ResNet101 [15] is selected). Then, RL trained agent is applied to refine the boxes. Stage two: Fastener defect is detected based on deep segmentation model YOLACT++ and a proposed criterion.

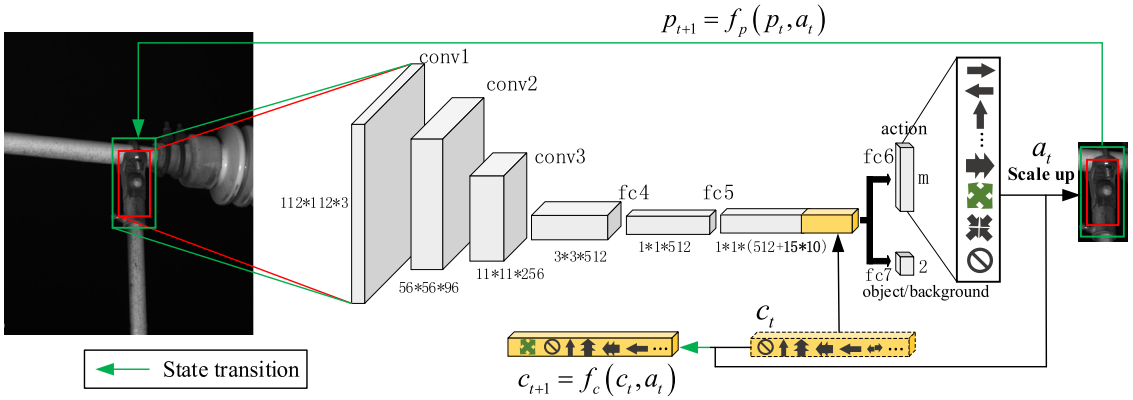


Fig. 5. Architecture of ADNET.

1) MDP Formulation: The MDP is defined by states $s \in S$, actions $a \in A$, state transition function $s' = f(s, a)$, and the reward $r(s, a)$. Here, we take the ADNET as an agent to find accurate box regions of BSs by taking sequential actions. By formulating the localization refinement as the MDP, the action policy of ADNET can be optimized by RL. The action, state, state transition function, and reward are formulated as follows.

a) Action: To make the initial box fit the position and shape of BS, transformations of moving (left, right, up, down), scale changing (scale up, scale down), and shape-changing (fatter, taller) are defined as possible actions. Especially, when the agent finds the optimum location or the current localized box is the same as the previous box, a stop action is needed to finalize the box searching during training. We define the action space A as shown in Fig. 6. Space A includes 15 actions and provides sufficient transform options for box changing.

b) State: As the localization refinement is a process of changing the geometry of box, the information of what actions the ADNET has taken before can help predict better boxes [20], [21]. The image patch within a box and the history

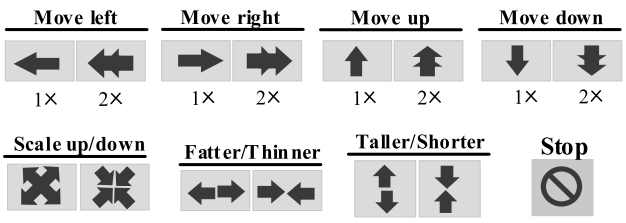


Fig. 6. Defined actions in our method.

actions are used to form the state s. For the localization refinement in image I at step t, the state s_t is defined as a tuple (p_t, c_t), where $p_t \in R^{112*112*3}$ is the image within the current box and $c_t \in R^{150}$denotes the encoded vector of action history. As such, p_t can be formulated as

$$p_t = \phi([x_t, y_t, w_t, h_t], I) \tag{1}$$

where (x_t, y_t) is the coordinate of center point of p_t in image I, w_t and h_t are the width and height of p_t, respectively. The function ϕ crops p_t from image I and resizes it to the input size of ADNET. c_t is a 150-D vector, because we choose

previous ten actions as the history actions, and each action is encoded by 15-D.

c) State Transition Function: When the ADNET selects an action a_t, the current s_t will transit to s_{t+1}. The state transition is performed by two functions $f_{\mathrm{p}}(p_t, a_t)$ and $f_{\mathrm{c}}(c_t, a_t)$, which are implemented on current image patch p_t and current action history c_t, respectively. As for the function f_{p}, the discrete amount of action transformations should be given. The discrete amounts of moving actions are given in (2). Formula (3) defines the discrete amounts of scale-changing and shape-changing actions. As the initial box is not far away from the BS, we set factors α_1, α_2, α_3, and α_4 to 0.05 in our experiments

$$x_t = \alpha_1 * w_t, \quad y_t = \alpha_2 * h_t \tag{2}$$

$$w_t = \alpha_3 * w_t, \quad h_t = \alpha_4 * h_t. \tag{3}$$

As for the state transition f_{c}, it adds the current action a_t into action history c_t as the latest action, and removes the earliest action.

d) Reward: The reward can be regarded as a feedback after taking an action. During the RL training, if the selected action can move the state transition to a better state, the agent will get a positive reward; otherwise, a zero reward or negative reward will be returned. In this article, the reward function $R_t(p_t, p_{t+1})$ is defined as follows:

$$R_t(p_t, p_{t+1}) = \begin{cases} +1, & \text{if } \mathrm{IoU}(p_{t+1}, G) > \mathrm{IoU}(p_t, G) \\ 0, & \text{if } \mathrm{IoU}(p_{t+1}, G) = \mathrm{IoU}(p_t, G) \\ -1, & \text{if } \mathrm{IoU}(p_{t+1}, G) = \mathrm{IoU}(p_t, G) \end{cases} \tag{4}$$

where G is the ground-truth box of target BS, the $\mathrm{IoU}(p_t, G)$ denotes the overlap ratio of the current patch p_t and the ground truth G of the target BS with intersection-over-union (IoU) criterion. In (4), only when the next patch gets closer to the ground-truth compared with the current patch, can the agent obtain a positive reward.

2) Training Objective: The agent ADNET is trained by RL to learn a state-action policy that makes a sequence of action decisions. Before applying RL, we initialize the ADNET by utilizing the weight parameters trained by supervised learning (SL), which has been proven useful for policy learning [17], [21].

In the SL stage, training samples p_i $\{i = 1, 2, \ldots, N\}$ are generated by imposing Gaussian noise on the BS ground-truth G_i $[x_i, y_i, w_i, h_i]$. The action label $o_i^{(\mathrm{act})}$ and class label $o_i^{(\mathrm{cls})}$ are defined by the following equations, respectively:

$$o_i^{(\mathrm{act})} = \arg\max_a \mathrm{IoU}\big(f_{\mathrm{p}}(p_i, a), G_i\big), \quad a \in A \tag{5}$$

$$o_i^{(\mathrm{cls})} = \begin{cases} 1, & \text{if } \mathrm{IoU}(p_i, G) > 0.6 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where A is the action space described in Fig. 6 and includes 15 actions. Then, the initial weight W_{SL}, $\{w_1, w_2, \ldots, w_7\}$, is learned by minimizing the loss L_{RL}

$$L_{\mathrm{SL}} = \frac{1}{N} \sum_{i=1}^{N} \Big[L\Big(o_i^{(\mathrm{act})}, \hat{o}_i^{(\mathrm{act})}\Big) + L\Big(o_i^{(\mathrm{cls})}, \hat{o}_i^{(\mathrm{cls})}\Big) \Big]. \tag{7}$$

The loss L_{SL} includes action prediction loss and classification (object/background) loss. N is the batch size of training samples and L denotes the cross-entropy loss. The variables $\hat{o}_i^{(\mathrm{act})}$ and $\hat{o}_i^{(\mathrm{cls})}$ are the predicted action and predicted class for sample i, respectively. Note that the history action vector is not used in this stage.

In the RL training stage, the weight W_{RL}, $\{w_1, w_2, \ldots, w_6\}$, is initialized by W_{SL}. The layer fc7 is ignored because only actions are concerned in RL. For an image frame l, an initial box will take T actions that are successively predicted by ADNET. In each step t, new features are drawn from the image, allowing the RL algorithm to adapt to new information. The action $a_{t,l}$ is selected by

$$a_{t,l} = \arg\max p\big(a | s_{t,l}; W_{\mathrm{RL}}\big) \tag{8}$$

where $t = 1, 2, \ldots, T-1, T$.

After action $a_{t,l}$ has been taken, the ADNET gets a reward $R_{t,l}$ according to (4). Meanwhile, a history of actions that ADNET has taken before is also recorded. Then, the history action vector c_t is updated by adding $a_{t,l}$ and removing the earliest action.

Finally, W_{RL} is updated using stochastic gradient ascent [29] to maximize the accumulated rewards of the training samples as follows:

$$\Delta W_{\mathrm{RL}} \infty \sum_{l}^{L} \sum_{t}^{T_l} \frac{\partial \log p\big(a | s_{t,l}; W_{\mathrm{RL}}\big)}{\partial W_{\mathrm{RL}}} R_{t,l} \tag{9}$$

where L is the size of image patches used for one iteration.

IV. Fastener Looseness Diagnosis

As the relationships between bolt, nut, and sleeve in the localized BS image can well characterize the fastener conditions, a fastener diagnosis criterion can be formed accordingly. To achieve effective detection, we utilize a deep segmentation model YOLACT++ to accurately extract the key components for diagnosis.

A. YOLACT++

The process of key components segmentation implemented by YOLACT++ is shown in Fig. 7. The input is the localized BS image which has been resized to 550 × 550 pixels. In inference, the input goes through the process as follows. First, a deformable convolution backbone which combined ResNet101 [16] with FPN [25] is applied to extract convolutional features (P3, P4, P5, P6, P7) from BS image. Then, the extracted features will be used in two branches to find out object regions for each key component. For the bottom branch, a Protonet is applied on feature map P3 to produce 32 prototypes, which will be the mask bases for segmentation. Only P3 is selected because P3 is the deepest features and has a large size, which helps segmentations of small components. For the upper branch, the prediction head is applied on all feature maps to generate initial boxes for components. The information of each initial box contains class, box coordinates, and mask coefficients. After the nonmaximum suppression (NMS) operation, only the boxes that were correctly classified are preserved. In Fig. 7, three boxes are preserved because

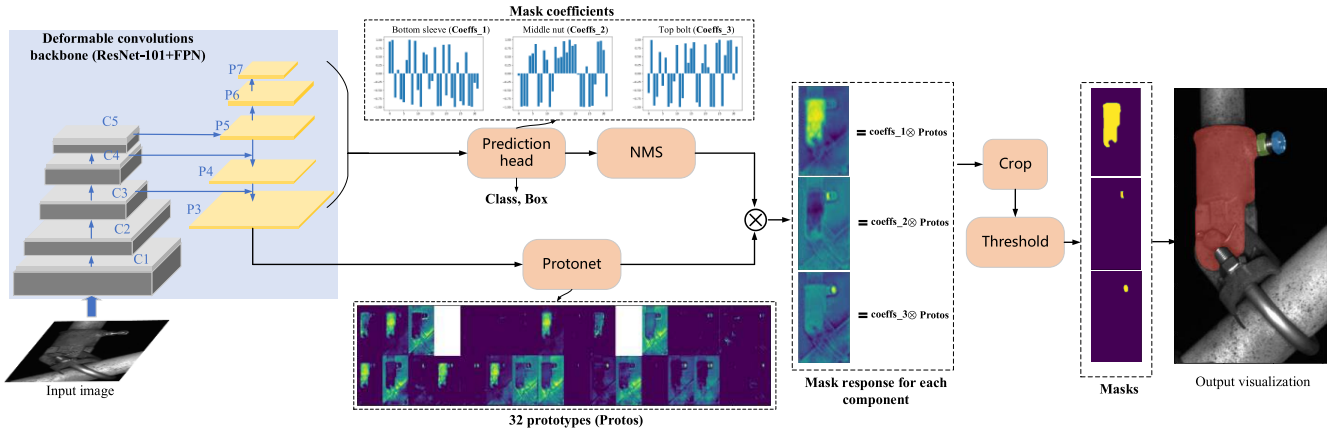


Fig. 7. Key components segmentation by YOLACT++.

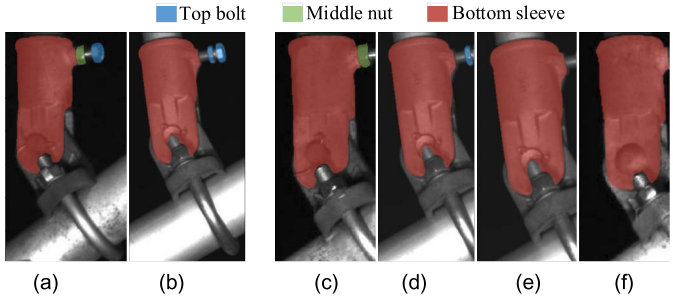


Fig. 8. Extracted masks in different localized BS images. (a) and (b) Segmentation results on complete BSs. (c)–(f) Segmentation results on incomplete BSs.

they are correctly classified as bottom sleeve, middle nut, and top bolt, respectively. For each preserved case, its mask coefficients will assemble with 32 prototypes (mask bases) linearly and form the mask response. As shown in Fig. 7, mask responses for each key component (sleeve, nut, and bolt) are very obvious, which verifies the effectiveness of the prototypes-based combination. Finally, final masks are obtained by cropping the mask response map and filtering the response values (activation results of sigmoid function) lower than threshold T. The output visualization shows the final segmentation results produced by YOLACT++.

B. Defect Diagnosis Criterion

Although the adopted RL agent has greatly improved the localization accuracy and YOLACT++ can extract the key components accurately, we still need to consider inaccurate localizations when forming a diagnosis criterion.

Some segmentation results in different localized BS images are shown in Fig. 8. Fig. 8(a) and (b) can obtain all key component masks, while Fig. 8(c)–(f) miss some masks because of inaccurate localizations. Based on the segmentation results and the position relationship between key components, we propose a defect diagnosis criterion for BS fastener as given in Table I. In our experiment, the bottom sleeve can always be extracted because it is relatively obvious. As shown in Fig. 8(a) and (c), if the middle nut is also extracted, then the fastener is diagnosed as normal and its defect degree value is 0. As shown in Fig. 8(b), if two top bolts are detected (note that when the nut is loose, it will be classified as top bolt in our algorithm because the spatial coherence of objects is preserved [28], [13]), then the fastener is diagnosed as loose and its defect degree value is in range (0, 1], which can be obtained by the process visualized in Fig. 9. As shown in Fig. 8(d), if only one top bolt is extracted, a loose state is also concluded, but the defect degree cannot be obtained due to the lack of further information. Finally, if only the bottom sleeve is extracted, a missing state label will be assigned and the defect degree value is 1, as shown in Fig. 8(e) and (f). The proposed criterion covers all fastener conditions and can provide defect degree values as maintenance decision support for railway asset managers.

TABLE I
PROPOSED DEFECT DIAGNOSIS CRITERION FOR BS FASTENER

State	Extracted key components			Defect degree
	Bottom sleeve	Middle nut	Top bolt	
Normal	√	√	√ (×)	0
Loose	√	×	√√	$\frac{\lvert BC\rvert}{\lvert BC\rvert+\lvert DE\rvert}\in(0,1]$
	√	×	√	—
Missing	√	×	×	1

V. EXPERIMENT AND ANALYSIS

A. Dataset and Experiment Setup

1) Dataset and Platform: The dataset built in our system is real catenary images collected by the JX-300 inspection vehicle shown in Fig. 10. It consists of a training set and a test set. We collect 4503 global images from the Changsha–Zhuzhou high-speed rail line in China, among which 3377 images form the training set and the rest 1126 images form the test set. To verify the method's generalization ability for other rail lines, we add 100 defect images from another railway line into the test set. Therefore, there are in total 1226 test images. Specifically, there are 3209 BS components in the training set,

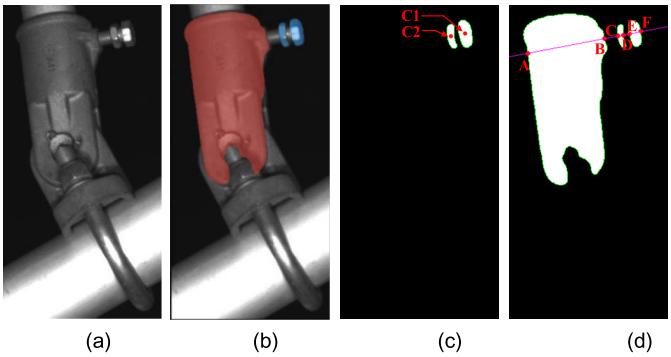


Fig. 9. Process of obtaining the fastener defect degree. (a) Original BS image. (b) Key components segmentation. (c) Centroids $C1$ and $C2$ are extracted from two top bolts and form a line. (d) Lengths of |BC| and |DE| are obtained according to intersections of the masks and the line.

and all of them are normal because constructing the proposed models do not require defect samples. Note that the training samples should include image cases with different illumination and angles, which are consistent with the real environment of a railway line. There are 1189 BSs in the test set, among which 1074 are normal and 115 are defective.

The training parameters of deep learning models used in this article are as follows. For Faster RCNN ResNet101, the iteration number is 30 000, learning rate is 0.001, and weight decay is 0.0001. For RL model, we set the epoch numbers of training iteration to be 200 for SL and 300 for RL, respectively. The learning rate is 0.0001 for conv1–conv3 and 0.001 for fully connected layers (fc4–fc7), and the weight decay is 0.0005. For YOLACT++, the iteration number is 32 000, learning rate is 0.001, and weight decay is 0.0005. Other methods for performance comparisons have similar settings. The experimental environment is as follows: Linux Ubuntu 18.04, MATLAB 2017a, Pytorch, and NVIDIA GTX1080 GPU. To obtain a more generalized representation of the test results, a fourfold cross validation (CV) is applied on all adopted deep learning models and the proposed pipeline.

2) Evaluation Metrics: For quantitative accuracy evaluation, the widely used metrics average recall (AR), mean average precision (mAP), and F1-score [14], [16], [28], [13] are adopted to evaluate the localization, segmentation, and classification tasks, respectively. The formulas of these metrics are as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$\text{AR} = \int_0^1 R(\text{IoU}) d_{\text{IoU}} \tag{12}$$

$$\text{mAP} = \frac{1}{n} \int P(R) d R \tag{13}$$

$$\text{F1-score} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{14}$$

where TP is the number of true positive tested samples, FP is the number of false positive tested samples, FN is the number of false negative tested samples, TN is the number of false positive (FP) tested samples, and n is the class number. In localization performance comparisons, we also use the improved proportion to show the proportion of improved samples in the test dataset. For efficiency evaluation, the frame per second (FPS) is the selected metric.

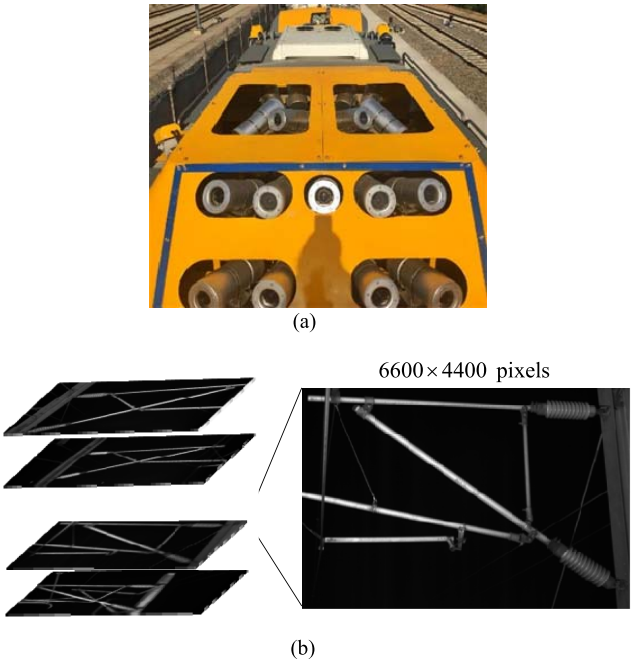


Fig. 10. Image collection system and samples. (a) JX-300 inspection vehicle. (b) Collected global catenary images.

B. Comparisons and Analysis

We conduct detailed experimental comparisons to verify the effectiveness of the proposed method. First, the performance of RL for improving localization is evaluated. Then, the effectiveness of the segmentation method YOLACT++ is verified. Finally, the performance of the overall framework is evaluated. For each stage, both quantitative and qualitative results are presented.

1) Verifying the Effectiveness of RL for Improving Localization: In our test dataset, there are 1074 labeled BS components. For a comprehensive comparison, we first apply generic localization methods and the existing state-of-the-art method Faster RCNN ResNet101 on the test images, and then use the RL agent to process their resulting boxes to see if the boxes are improved. Here, we use a Gauss initialization to represent generic localization methods. The Gauss initial method applies a Gauss function on the true BS box and generates boxes with random distribution around the true box. It means that the Gauss initial method can produce all possible box conditions. Therefore, it can represent generic localization methods and be used for performance comparison. The quantitative comparison of localization performance is given in Table II and Fig. 11. Qualitative results of localization refinement are displayed in Fig. 13. For convenience, the Faster RCNN ResNet101 is simplified as FRCNN ResNet101 in Figs. 11 and 12 and Tables II, III, and VI.

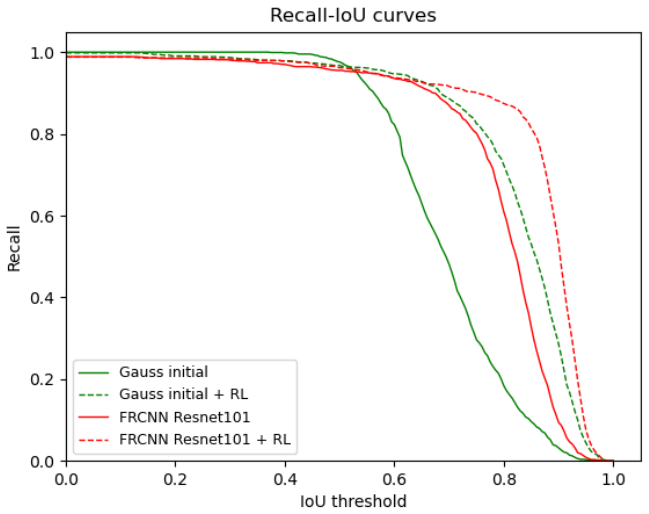


Fig. 11. Recall versus IoU curves of compared methods.

TABLE II
QUANTITATIVE COMPARISON OF LOCALIZATION PERFORMANCE

Method	AR	Improved proportion	Speed(f/s)
Gauss initial	0.696	--	--
Gauss initial + RL	**0.824**	**82.5% (866/1074)**	**6.13**
FRCNN ResNet101	0.790	--	--
FRCNN ResNet101 + RL	**0.862**	**86.4% (928/1074)**	**6.13**

In Table II, the AR of Gauss initial method is 0.696, which corresponds to the area under the solid green line in Fig. 11. The solid green line shows the recall is nearly linearly decreased in the IoU range [0.5, 1], which indicates the produced boxes are indeed distributed randomly. When the trained RL agent is applied on these Gauss boxes, the overall localization accuracy AR increased from 0.696 to 0.824 by a margin of 0.128. As shown in Fig. 11, the Recall–IoU curve changing from green solid to green dash shows details about AR improvement by using our RL refinement model. Specifically, among 1074 Gauss initialized BS boxes, 866 boxes get close to BS true positions. Thus, the improved proportion is 82.5%. For Faster RCNN ResNet101, the original localization accuracy AR is 0.790, which is much higher than the average precision (AP) of Gauss initial method but is a little bit lower than the AR of "Gauss initial + RL" method. It indicates that applying the RL agent on the random initialized boxes can obtain better localization performance than Faster RCNN ResNet101. When the RL agent is used on the boxes produced by Faster RCNN ResNet101, the accuracy AR can be increased to 0.862. The AR improvement can be observed from the Recall–IoU curve changing from red solid to red dash, as shown in Fig. 11. Specifically, 928 of 1074 BS boxes produced by Faster RCNN ResNet101 have been refined closer to BS true positions with an improved proportion of 86.4%. As for the efficiency of RL, the processing speed is 6.13 frames/s, as given in Table II. It means the RL agent takes only 0.16 s for each BS image. The quantitative comparisons in Table II and Fig. 11 verified the effectiveness of the proposed RL model for improving localization.

TABLE III
QUANTITATIVE COMPARISON OF FOURFOLD CV ON LOCALIZATION MODELS

Method	AR	Improved AR	Speed(f/s)
FRCNN ResNet101 (CV1)	0.790	--	--
FRCNN ResNet101 + RL (CV 1)	**0.862**	**0.072**	**6.13**
FRCNN ResNet101 (CV 2)	0.780	--	--
FRCNN ResNet101 + RL (CV 2)	**0.845**	**0.065**	**6.13**
FRCNN ResNet101 (CV 3)	0.790	--	--
FRCNN ResNet101 + RL (CV 3)	**0.868**	**0.078**	**6.13**
FRCNN ResNet101 (CV 4)	0.787	--	--
FRCNN ResNet101 + RL (CV 4)	**0.854**	**0.067**	**6.13**

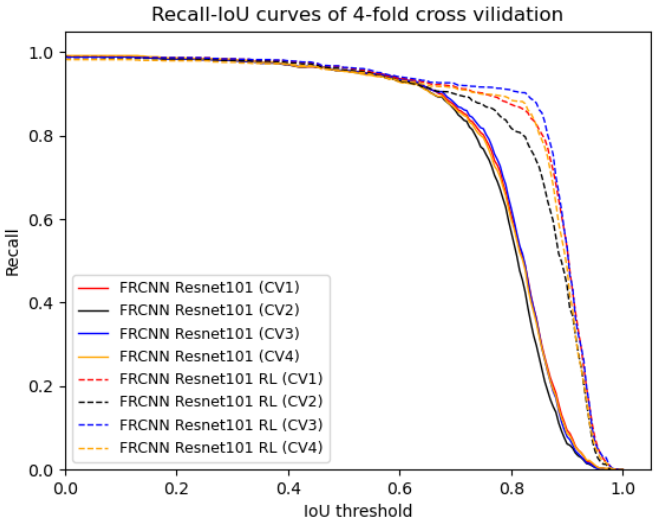


Fig. 12. Fourfold CV results of Faster RCNN ResNet101 and RL.

To verify the generalization ability of FRCNN ResNet101 and RL. First, we apply a fourfold CV on FRCNN ResNet101. The performances of four trained models are shown in Fig. 12 and Table III. It can be seen that the FRCNN ResNet101 CV1–CV4 have close Recall–IoU curves and close accuracies (ARs), which are 0.790, 0.780, 0.790, and 0.787, respectively. The results indicate that the Faster RCNN ResNet101 is stable and has good generalization ability.

Then, the CV is applied to the localization refinement model RL. Four RL models RL CV1–CV4 are trained by train sets CV1–CV4. As the function of the RL model is processing the results produced by Faster RCNN ResNet101.

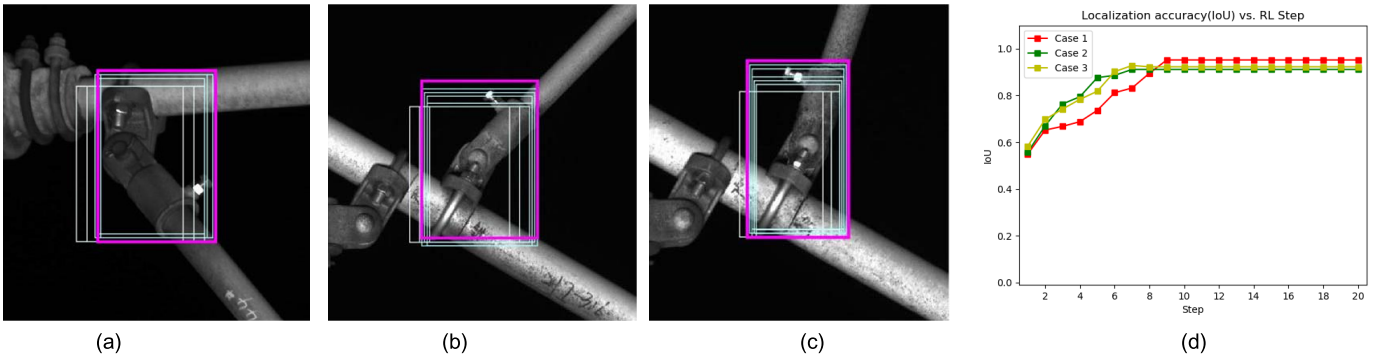


Fig. 13. Visualization of localization refinement cases performed by RL agent. (a) Case 1. (b) Case 2. (c) Case 3. (d) Accuracy changes of the cases during their refinement processes.

TABLE IV
QUANTITATIVE COMPARISON OF SEGMENTATION PERFORMANCE

Method	AP (Average Precision for mask)			mAP	Speed(f/s)
	Top bolt	Middle nut	Bottom sleeve		
Mask R-CNN	0.937	0.896	0.966	0.933	2.86
YOLACT	0.970	0.962	0.993	0.975	17.01
YOLACT++	0.963	**0.965**	**0.993**	0.974	**18.10**

TABLE V
QUANTITATIVE COMPARISON OF FOURFOLD CV ON YOLACT ++

Method	AP (Average Precision for mask)			mAP	Speed(f/s)
	Top bolt	Middle nut	Bottom sleeve		
YOLACT++ (CV1)	0.963	0.965	0.993	0.974	18.10
YOLACT++ (CV2)	0.974	0.952	0.990	0.972	
YOLACT++ (CV3)	0.963	0.965	0.990	0.973	
YOLACT++ (CV4)	0.972	0.954	0.990	0.972	

Therefore, the test sets for RL CV1–CV4 are the results produced by Faster RCNN ResNet101 CV1–CV4, respectively. The performances of four trained RL models are also shown in Fig. 12 and Table III. It can be seen that the FRCNN ResNet101 RL CV1–CV4 have relatively similar Recall–IoU curves and close accuracies (ARs), which are 0.862, 0.845, 0.868, and 0.854, respectively. The improved ARs are 0.072, 0.065, 0.078, and 0.067. The results indicate that the proposed RL models are relatively stable with a good generalization ability.

For qualitative results, we present some cases of the localization refinement processes in Fig. 13. Fig. 13(a)–(c) shows how a coarse box is changed during the refinement. The white box is the initial box produced by Faster RCNN ResNet101 or Gauss initial. Then, RL agent takes a series of actions to change the box from white to purple. During the refinement, the later box has a darker color than the previous box. Fig. 13(d) shows the IoU changes for each case during the refinement. For each case, the accuracy gets higher with the increase in the number of steps. The qualitative results in Fig. 13 show the adopted RL agent can improve BS localization adaptively.

2) Verifying the Effectiveness of YOLACT++ for Key Components Segmentation: As described in Section IV-B, the proposed defect diagnosis criterion is based on the segmentation results. If the key components, namely, top bolt, middle nut, and bottom sleeve in the localized BS image can be correctly extracted, the defect diagnosis criterion can be performed. Therefore, the effectiveness of YOLACT++ should be verified. To this end, we compare YOLACT++ with two other competitive deep learning segmentation methods, i.e., Mask R-CNN [24] and YOLACT [28]. The accuracy and efficiency evaluation metrics are mAP and speed, respectively, which are introduced in Section V-A. The test images are 1074 ground-truth BS images labeled with masks.

The IoU threshold in the experiment is set to 0.6. We obtain the quantitative results of all selected segmentation methods, as given in Table IV. For Mask R-CNN, the APs of top bolt, middle nut, and bottom sleeve are 0.937, 0.896, and 0.966, respectively. Combining the APs of all classes, the mAP of Mask R-CNN is 0.933, which is a high segmentation accuracy. However, the speed of Mask R-CNN is 2.86 frames/s, meaning each test image takes 0.35 s. The YOLACT++ and YOLACT have close mAPs, which are 0.974 and 0.975, respectively. Compared with Mask R-CNN, the YOLACT++ and YOLACT exceed about 0.042 in terms of mAP, and all three classes of components have superior accuracy. As for the efficiency, the speed of YOLACT++ and YOLACT are 18.10 and 17.01, respectively, which are about six times faster than Mask R-CNN. Both YOLACT and YOLACT++ are promising for real-time application. Considering the balance between accuracy and speed, the YOLACT++ is used to extract key components in this article.

For the CV on YOLACT++, the results of FRCNN ResNet101 RL CV1–CV4 are relatively close but with

TABLE VI
QUANTITATIVE COMPARISON OF OVERALL PERFORMANCE

	Method	TN	FN	FP	TP	F1-score	Speed(f/s)
Pipeline 1	Gauss initial + YOLACT++	931	17	143	98	0.551	--
Pipeline 2	Gauss initial + RL + YOLACT++	1032	5	42	110	0.824	--
Pipeline 3	FRCNN ResNet101 + YOLACT++	1019	6	55	109	0.781	7.33
Pipeline 4	FRCNN ResNet101 + RL + YOLACT++	**1039**	**3**	**35**	**112**	**0.855**	3.34

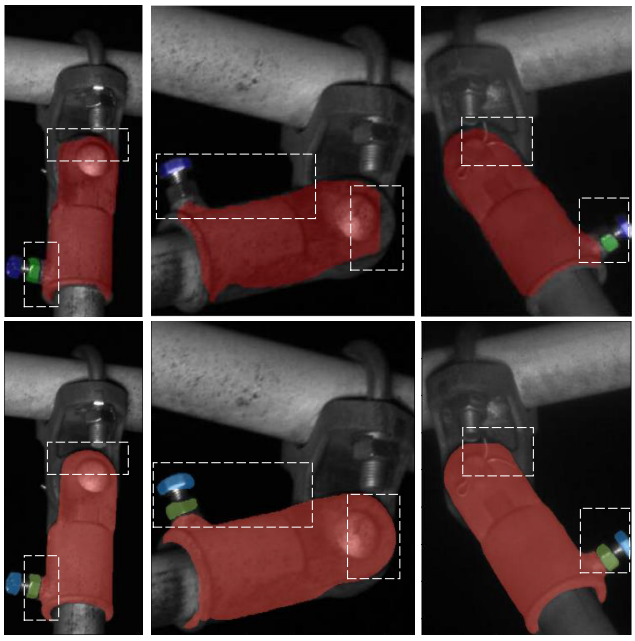


Fig. 14. Qualitative comparison of segmentation results. Masks produced by Mask R-CNN (top). Masks produced by YOLACT++ (bottom).

small differences. To perform fair comparisons between YOLACT++ CV1, YOLACT++ CV2, YOLACT++ CV3 and YOLACT++ CV4, the ground-truth BS components in global image test set CV1–CV4 are used as the test set for YOLACT++ CV1–CV4, respectively. The CV results for YOLACT++ are shown in Table V. It can be seen that the YOLACY++ CV1–CV4 have close accuracies (mAPs), which are 0.974, 0.972, 0.973, and 0.972, respectively. The results indicate that the YOLACT++ is stable and has good generalization ability.

Fig. 14 gives several cases for qualitative comparison of segmentation performance. The masks in top images are produced by Mask R-CNN and masks in bottom images are produced by YOLACT ++. In these cases, Mask R-CNN cannot predict good masks in dashed-line boxes. However, YOLACT++ can predict better masks in the same boxes. The results in Table IV and Fig. 14 show that YOLACT++ is superior to the other two competitive deep learning methods for catenary component segmentation.

3) Overall Diagnosis Performance Comparison: So far, we verified the effectiveness of our localization method and segmentation method independently. As the defect diagnosis system is a cascade pipeline, the final results rely on the cascaded performance. Here, we evaluate the overall performance of the method combining the localization stage, the segmentation stage, and the defect diagnosis criterion. The test set includes 1074 normal BS fasteners and 115 defective BS fasteners. We implement four different cascaded methods on the test set and compare their performances by using the F1-score metric described in Section V-A. The test results are shown in Table VI. To simplify the name of cascaded methods, we rename them as Pipeline 1–Pipeline 4.

In Table VI, the results of true positives (TPs), false negatives (FNs), FPs, and true negatives (TNs) of each method are presented. Specifically, TP is the number of defective fasteners that are correctly diagnosed, FN is the number of defective fasteners that are mistakenly diagnosed as normal fasteners, FP is the number of normal fasteners that are mistakenly diagnosed as defective fasteners, and TN is the number of normal fasteners that are correctly diagnosed. By observing the above four numbers, we can know the diagnosis results of all BS fasteners in the dataset. Finally, the F1-score reflects the overall performance of the cascaded method.

As given in Table VI, Pipeline 4 has the highest F1-score 0.855. Comparing Pipeline 4 with Pipeline 3, the application of RL can improve the F1-score of Pipeline 3 by a margin of 0.031. Comparing Pipeline 2 with Pipeline 1, the application of RL increases the F1-score of Pipeline 1 by a margin of 0.23. These two comparisons indicate that: 1) the proposed RL agent is effective for improving the performance of the overall defect diagnosis system and 2) the performance of localization can directly affect the final diagnosis.

For the TP, FP, FN, and TN results, Pipeline 4 performs better than all other pipelines. Specifically, the TN and FP of Pipeline 4 are 1039 and 35, respectively, which indicates 96.7% [TN/(TN + FP)] of normal BS fasteners in the test set are correctly classified. The rest 3.3% normal samples are mistakenly classified as defects. For the diagnosing results of defective samples, Pipeline 4 can correctly detect 97.4% [TP/(TP + FN)] of defect samples. Among 115 defect samples in the test set, 112 (TP) defect samples are correctly

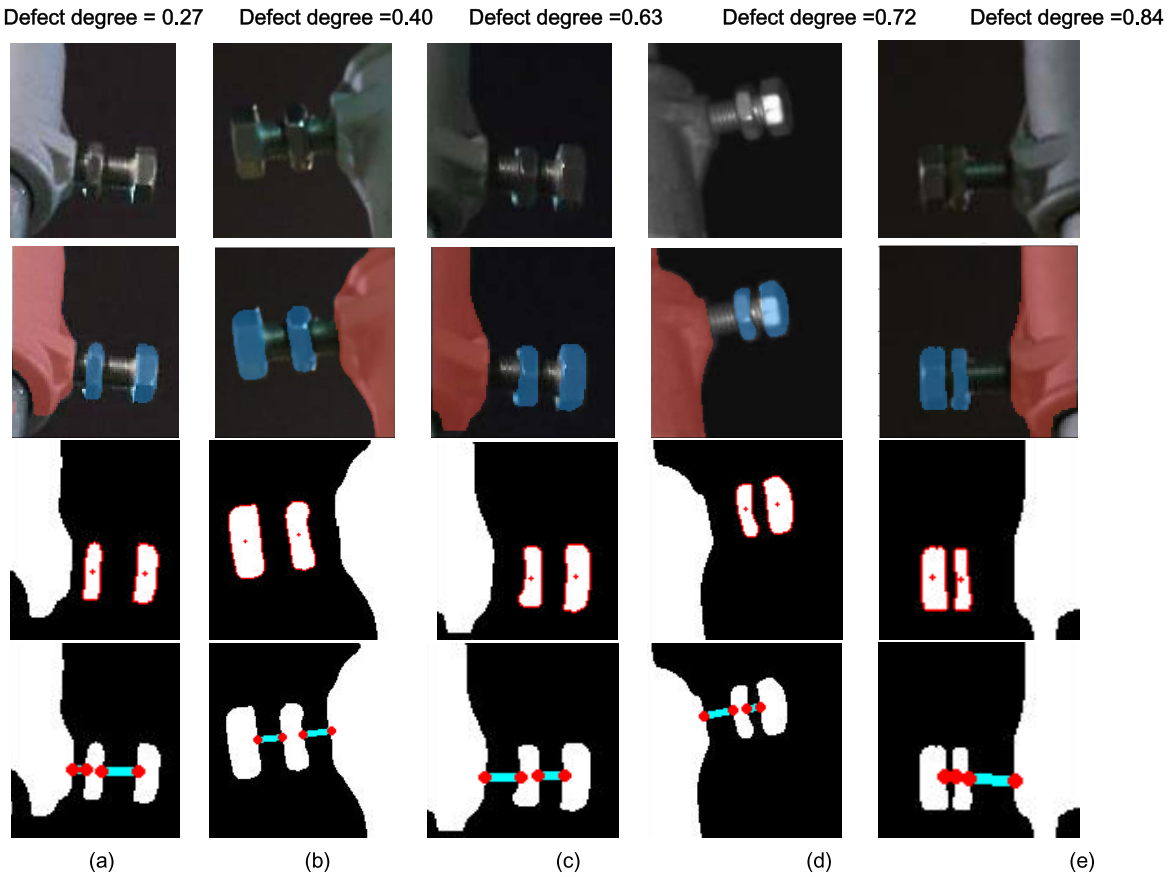


Fig. 15. Looseness degree results produced by the proposed diagnosis criterion. The looseness degree increases from (a)–(e), which are 0.27, 0.40, 0.63, 0.72, and 0.84, respectively. Raw images (first row). Segmented masks (second row). Obtained centroids of two top bolts (third row). Looseness is characterized and defect degrees are obtained (fourth row).

TABLE VII
QUANTITATIVE COMPARISON OF FOURFOLD CV ON THE OVERALL PIPELINE

Overall method	TN	FN	FP	TP	F1-score	Speed(f/s)
Pipeline 4 (CV1)	1039	3	35	112	0.855	3.34
Pipeline 4 (CV2)	1065	3	37	112	0.849	
Pipeline 4 (CV3)	1030	3	30	112	0.872	
Pipeline 4 (CV4)	1108	3	34	112	0.858	

diagnosed and only three defect samples are mistakenly classified as normal samples. If a loose state is diagnosed by Pipeline 4, the looseness degree can be further obtained according to the criterion given in Table I. As shown in Fig. 15, the samples with different looseness degrees are assigned with the corresponding defect scores. The defect scores from Fig. 15(a) to (e) are increasing with the looseness getting severer. As for the efficiency, the speed of Pipeline 4 is 3.34 frames/s, which is larger than the minimum method speed 3.33 frames/s to achieve real-time detection when the train speed is 300 km/h. Normally, the maximum speed of inspection trains for China high-speed lines is 300 km/h. It indicates that our method can achieve real-time detection when the train runs at a speed slower than 300 km/h. Comparing results of Pipeline 3 and Pipeline 4 in Table VI, although the method speed is decreased, the accuracy increased from 0.781 to 0.855. As a low accuracy can cause overlooked defects, a high detection accuracy should have a higher priority over real-time detections in practice. Therefore, the time sacrifice is acceptable and worthwhile.

The fourfold CV results for the overall pipeline (FRCNN ResNet101 + RL + YOLACT++ with defect diagnosis

criterion) are given in Table VII. It can be seen that the Pipeline CV1–CV4 have relatively close accuracies (F1-scores), which are 0.855, 0.849, 0.872, and 0.858, respectively. The CV results indicate the proposed pipeline has a good generalization ability.

The quantitative comparison of overall performance in Tables VI and VII shows the proposed method Faster RCNN ResNet101 + RL + YOLACT++ is effective for diagnosing catenary BS fasteners. The diagnosis results can be directly used as maintenance decision support for asset managers.

VI. Conclusion

This article proposes an effective pipeline for the defect detection of BS fasteners in railway catenary systems based on computer vision. The proposed method can address several existing issues for diagnosing BS fasteners. First, we adopt an action-driven RL method to refine the coarsely localized BS boxes, which improves the localization accuracy. Second, the deep learning segmentation model YOLACT++ is utilized to extract the masks of key fastener components accurately and in a real-time. Third, a diagnosis criterion is proposed for defect detection. The effectiveness of each module in the proposed pipeline is verified by experiments and comparisons with existing methods. Nevertheless, there are still some further improvements can be conducted.

1) Although the proposed pipeline has achieved a good diagnosis performance, there is still room for further improvement in detection accuracy. Reducing the false detections (FN and FP in Table VI) should be further considered.
2) For real-time detection, although our method can achieve real-time detection when the inspection train runs slower than 300 km/h, there are still two aspects that can be further investigated to realize real-time detection when the train speed is higher. First, more advanced computation devices such as NVIDIA Quadro RTX8000, Titan RTX could be employed. Second, lighter detection models can be developed, which is very challenging because the measures taken to reduce time cost usually reduces the detection accuracy. A high accuracy of defect detection should be guaranteed for catenary monitoring, because low accuracy results are meaningless in practice.
3) From the literature on RL, we have worked with one selected method. Meanwhile, other methods can be applied in future research [17]–[19], [22], [31], [32]. Elements from RL applications in other fields, such as robotics and control [33], [34], can also be considered for the dynamic monitoring of catenary systems.

References

[1] S. Gao, "Automatic detection and monitoring system of pantograph–catenary in China's high-speed railways," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021, doi: 10.1109/TIM.2020.3022487.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[4] 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)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[5] 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.

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

[7] Y. Han, Z. Liu, X. Geng, and J. P. Zhong, "Fracture detection of ear pieces in catenary support devices of high-speed railway based on HOG features and two-dimensional Gabor transform," *J. China Railway Soc.*, vol. 39, no. 2, pp. 52–57, 2017.

[8] J. Zhong, Z. Liu, G. Zhang, and Z. Han, "Condition detection of swivel clevis pins in overhead contact system of high-speed railway," *J. China Railway Soc.*, vol. 39, no. 6, pp. 65–71, Jun. 2017.

[9] H. Fan, P. C. Cosman, Y. Hou, and B. Li, "High-speed railway fastener detection based on a line local binary pattern," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 788–792, Jun. 2018.

[10] G. Kang, S. Gao, L. Yu, and D. Zhang, "Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2679–2690, Aug. 2019.

[11] L. Chang, Z. Liu, Y. Shen, and G. Zhang, "Novel multistate fault diagnosis and location method for key components of high-speed trains," *IEEE Trans. Ind. Electron.*, vol. 68, no. 4, pp. 3537–3547, Apr. 2021.

[12] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4203–4212.

[13] J. Wang, L. Luo, W. Ye, and S. Zhu, "A defect-detection method of split pins in the catenary fastening devices of high-speed railway based on deep learning," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9517–9525, Dec. 2020.

[14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[15] Z. Liu, Y. Lyu, L. Wang, and Z. Han, "Detection approach based on an improved faster RCNN for brace sleeve screws in high-speed railways," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4395–4403, Jul. 2020.

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

[17] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[18] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," 2015, *arXiv:1509.06461v2*. [Online]. Available: https://arxiv.org/abs/1509.06461v2

[19] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Beijing, China, 2014, pp. 387–395.

[20] J. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2488–2496.

[21] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2711–2720.

[22] S. Liu, D. Huang, and Y. Wang, "Pay attention to them: Deep reinforcement learning-based cascade object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2544–2556, Jul. 2020.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput. Assist. Interv. (MICCAI)*, 2015, pp. 234–241.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016, *arXiv:1612.03144*. [Online]. Available: http://arxiv.org/abs/1612.03144

[26] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 9156–9165.

[27] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT++: Better real-time instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, Dec. 3, 2020, doi: 10.1109/TPAMI.2020.3014297.

[28] J. Cao, G. Yang, and X. Yang, "A pixel-level segmentation convolutional neural network based on deep feature fusion for surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021, doi: 10.1109/TIM.2020.3033726.
[29] J. Zhong, Z. Liu, H. Wang, W. Liu, C. Yang, and A. Nunez, "Action-driven reinforcement learning for improving localization of brace sleeve in railway catenary," in *Proc. Int. Conf. Sens., Meas. Data Anal. era Artif. Intell. (ICSMD)*, Xi'an, China, Oct. 2020, pp. 100–105.
[30] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, May 1992.
[31] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.
[32] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annu. Rev. Control*, vol. 46, pp. 8–28, Jan. 2018.
[33] Y. P. Pane, S. P. Nageshrao, J. Kober, and R. Babuška, "Reinforcement learning based compensation methods for robot manipulators," *Eng. Appl. Artif. Intell.*, vol. 78, pp. 236–247, Feb. 2019.
[34] D. L. Leottau, J. Ruiz-del-Solar, and R. Babuška, "Decentralized reinforcement learning of robot behaviors," *Artif. Intell.*, vol. 256, pp. 130–159, Mar. 2018.

Junping Zhong (Student Member, IEEE) received the B.S. degree in electronic and information engineering from Southwest Jiaotong University, Chengdu, China, in 2014, where he is currently pursuing the Ph.D. degree in electrical engineering.

His current research interests include image processing, computer vision, and their applications in railway fault detection.

Zhigang Liu (Senior Member, IEEE) received the Ph.D. degree in power systems and their automation from Southwest Jiaotong University, Chengdu, China, in 2003.

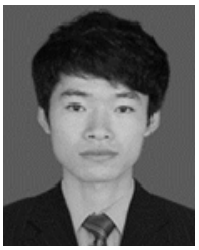
He is currently a Full-time Professor with the School of Electrical Engineering, Southwest Jiaotong University. He has written three books and authored more than 150 peer-reviewed journal articles and conference papers. His research interests are the electrical relationship of EMUs and traction, detection, and assessment of pantograph-catenary in high-speed railway.

Dr. Liu was elected as a fellow of The Institution of Engineering and Technology (IET) in 2017. He is an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, and IEEE ACCESS. He received the IEEE TIM's Outstanding Associate Editors for 2019 and 2020 as well as the Outstanding Reviewer for IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT in 2018.

Hongrui Wang (Member, IEEE) received the Ph.D. degree from the Section of Railway Engineering, Delft University of Technology, Delft, The Netherlands, in 2019.

He was a Post-Doctoral Researcher with the Delft University of Technology, until November 2020, where he is currently an Assistant Professor with the Department of Engineering Structures. His research interests include signal processing, artificial intelligence, and their applications in the structural health monitoring and digital modeling and the design of railway infrastructures.

Dr. Wang is an Associate Editor of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

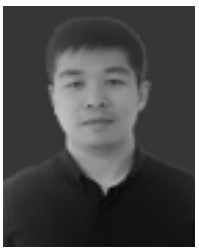


Wenqiang Liu (Graduate Student Member, IEEE) received the B.S. degree in electronic information engineering from Southwest Jiaotong University, Chengdu, China, in 2013, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering.

His current research interests include image processing, computer vision, deep learning, 3-D modeling, and virtual reality and their applications in fault detection and diagnosis in the electrified railway industry.

Cheng Yang received the B.S. degree in power system and its automation from Southwest Jiaotong University, Leshan, China, in 2018. He is currently pursuing the master's degree in electrical engineering with Southwest Jiaotong University, Chengdu, China.

His current research interests include computer vision and their applications in railway fault detection.



Zhiwei Han (Member, IEEE) received the Ph.D. degree in power system and its automation from Southwest Jiaotong University, Chengdu, China, in 2013.

He is currently a Lecturer with the School of Electrical Engineering, Southwest Jiaotong University. His current research interests include modern signal processing, computer vision, and their application in railway and electric power system.

Alfredo Núñez (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Universidad de Chile, Santiago, Chile, in 2010.

He was a Post-Doctoral Researcher with the Delft Center for Systems and Control, Delft, The Netherlands. Since 2013, he has been with the Section of Railway Engineering, Department of Engineering Structures, Delft University of Technology, Delft, where he is currently an Assistant Professor (tenured) of Data-Based Maintenance for Railway Infrastructure. His current research interests include the maintenance of railway infrastructures, intelligent conditioning monitoring in railway systems, big data, risk analysis, and optimization.

Dr. Núñez is in the Editorial Board of the *Applied Soft Computing* and an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.