IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 26, 2020, accepted August 7, 2020, date of publication August 18, 2020, date of current version August 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017691

Change Detection: The Framework of Visual Inspection System for Railway Plug Defects

XINYU DU, YU CHENG, AND ZICHEN GU
Infrastructure Inspection Institute, China Academy of Railway Sciences, Beijing 100081, China

Corresponding author: Xinyu Du (xinyudu@qq.com)

This work was supported in part by the Scientific and Technological Research Projects of the China Railway Corporation under Grant AJJ19203B, and in part by the Scientific Research Projects of China Academy of Railway Sciences under Grant 2019YJ158.

ABSTRACT Railway plug defects impact the safety of a railway system. To detect railway plug defects, we establish the framework of a visual inspection system (VIS), which is the first system that can perform railway plug inspection automatically and intelligently. Using the idea of change detection, the framework includes three algorithm modules, which are named the object location, image alignment and similarity measurement modules. After the image acquisition system captures a rail image as the input, the three algorithm modules process the image in order. First, in the object location module, a deep convolutional neural network is used to perform plug location. Second, in the image alignment module, a simple and fast method is designed to align key images using histogram of oriented gradients features. Third, in the similarity measurement module, the $\chi 2$ distance is used to compute the similarity between the two plug regions in an inspection image and in an aligned ground-truth image. The results of the similarity measurement are sorted when all inspection images are processed. Therefore, the inspection images with smaller similarity values are ranked higher and the plugs in the images have larger probabilities of defects. The framework has passed the practice tests, and the visual inspection system using this framework has already been authorized by the China Railway Corporation and will be equipped in many inspection trains belonging to local railway corporations.

INDEX TERMS Visual inspection system, image processing, railway engineering.

I. INTRODUCTION

In recent years, high speed railway (HSR) transportation has become more important and the length of the HSR in China is increasing greatly. Hence, the demands of HSR maintenance have also rapidly increased and become urgent. For HSR maintenance, inspection is an important and necessary preliminary task to look for and confirm defects in the tracks, catenaries, tunnels, subgrades and various equipment in or by railway lines. However, traditional inspection, depending on contact measurement techniques and human patrolling detection, is slow, subjective, dangerous, and inefficient. Therefore, automatic and noncontact measurement systems are being proposed to overcome the limitations of human inspection.

Since there are many different kinds of inspection targets, many different kinds of methods are used in automatic and noncontact measurement systems. For example, rail internal cracks can detected by ultrasonic techniques [1]–[3], eddy current testing methods have been proposed to confirm railroad damage [4], [5], and acoustic emissions and signal processing techniques have also been used in rail inspection [6]. In addition to all of those, visual inspection systems (VISs) are used more widely, including to detect track surface defects [7], [8], fasteners [9]–[11], bolts [12], slabs [13], squats [14], and catenary geometry parameters and units [15]–[18].

A plug is an important component that is used to transmit control information signals when a train is moving, and its defects may cause very severe incidents, such as rear-end or head-on collisions. Figure 1 shows the appearance of two plugs. Different from the inspections for the track infrastructure and catenaries mentioned above, there are some specificities and challenges for the inspection of railway plugs. First, plug shapes vary because their tail cables are not rigid; therefore, normal plugs (samples) have various appearances

The associate editor coordinating the review of this manuscript and approving it for publication was Min Xia.

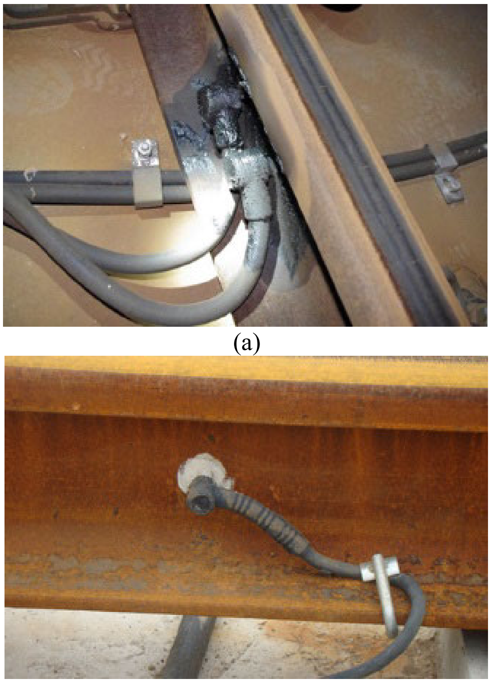


FIGURE 1. Plugs: (a) as the lead wire for the track current, and (b) as the lead wire for the compensation capacitor.

when being assessed using a machine. Second, since the installation sites of plugs are nonuniform and discrete along railways, locating plugs precisely becomes very difficult. According to our rough statistics, approximately 0.5% of inspection images contain plugs. Third, series of images that may contain plugs captured by the same camera look almost the same because the installation sites and angles of those cameras are fixed. At last, there are fewer plugs than other inspection targets, such as tracks and catenaries; therefore, the number of abnormal samples of plugs is lower than that of others and traditional machine learning methods cannot be used directly and practically in an automatic plug inspection system. Therefore, unlike track and catenary inspections, plug inspections always depend on human patrolling detection before our VIS is used. To solve those problems with plug inspection, we design a VIS using a change detection framework. Although the concept of change detection is already used in the fields of remote sensing [19, 20], video surveillance [21], and medical diagnosis and treatment [22], the change detection framework proposed in this paper is designed especially for railways. Because the detected objects vary based on the application, it is difficult to compare algorithms using the change detection framework [23]. It is applied to solve the plug inspection problem for the first time, and the VIS we designed is the first automatic and intelligent plug inspection system. Among our early works, paper [24] presented our earliest work on an objection location module, a small part of the whole VIS, for plug location. The object location methods used in this work are traditional machine learning algorithms. Then, in paper [25], we designed a kind of deep neural network for the plug location module, and it performed better. However, both papers focus only on the object location module. In this paper, we greatly advance the previous work, and detail the whole inspection system for plug defects, including the hardware for image acquisition and the software for the change detection framework that contains three algorithm modules.

The rest of this paper is organized as follows. Section II introduces the overview of the VIS for plug defects. Section III describes the change detection framework. Section IV presents the field experiment results. Finally, section V provides some discussions and conclusions.

II. SYSTEM OVERVIEW

We design a VIS for railway plug defects. The VIS is composed of a hardware and a software part. The hardware part includes two high-speed digital cameras (named the inboard cameras) installed under a test train, as displayed in figure 2. Those cameras are used to capture series of images that may contain plugs when the train runs. This is named the image acquisition subsystem (IAS) in this paper.

Then, the obtained images are analyzed intelligently using a series of image processing and machine learning algorithms in the software part of the system. Those algorithms constitute the change detection framework in our work. Briefly, the core idea of change detection is the following: how to find plugs that contain appearance changes between two inspections. More changes mean greater probabilities of plug defects in one inspection compared to the ground-truth, which consists of nondefective plugs recorded just after the railroad was constructed or that were confirmed via manual inspection. In other words, the ground-truth means that plugs are undamaged in the inspection. In addition, the ground-truth dataset includes serial images captured by the IAS and their locations. The locations of these images are provided by the vehicle positioning system (VPS) that is standard equipment for the test trains used in China and it is not the theme of this paper. It is noted that the image location is presented in the form of kilometers plus meters and all images in this paper are equipped with location information. Using image processing software, the change detection framework is embedded in our VIS for railway plug defects.

A. HARDWARE: IAS

The IAS is composed of some commercially available components, such as cameras, light sources, etc. As shown in figure 2, two Dalsa Spyder 2 line-scan cameras installed under the test train are used to capture the plug images of the left and right tracks. Using those cameras, the maximum line rate is 65 000 lines/s and the images have a resolution of 1024 pixels. The protocol of those cameras is the Cameralink protocol and those images are captured by a IPC(Industrial Personal Computer)-CamLink frame grabber. An illumination setup equipped with a line array of LED light sources is installed under the test train near the line-scan cameras to reduce the effect of natural light. In addition,

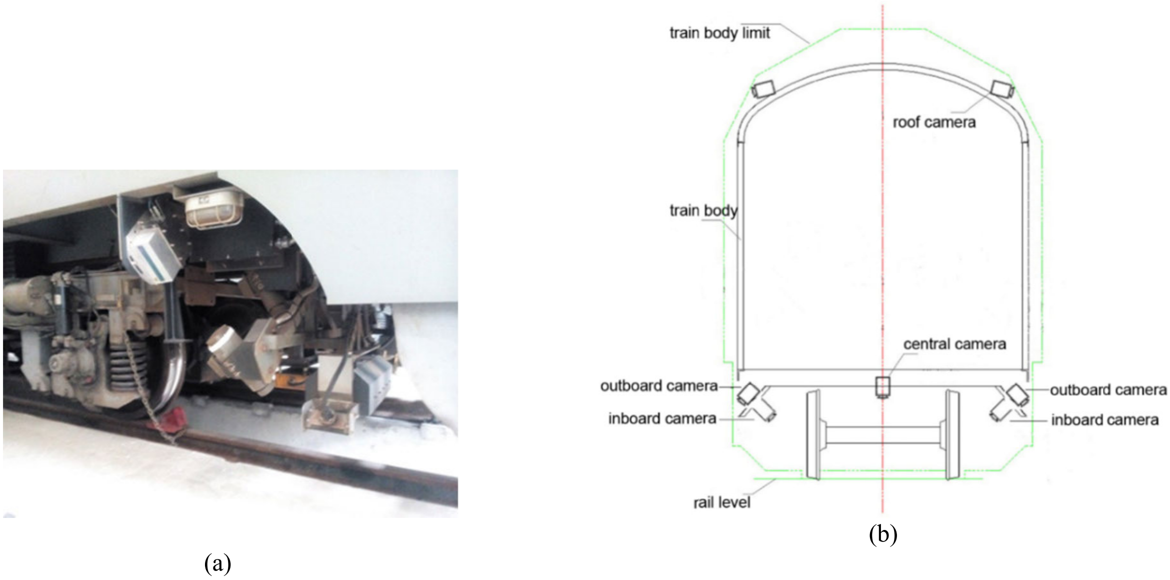


FIGURE 2. The high-speed digital cameras are shown in the following: (a) the real installation picture, and (b) the design drawing.

a wheel encoder is used to trigger the inboard, outboard, and central cameras to synchronize their data acquisition every two meters. The images captured by these cameras were taken synchronously at the same location of the rail, and can be stitched into a complete picture of the tracks and accessories. Equipped with those components, the IAS can capture high-resolution images for the VIS.

B. SOFTWARE: CHANGE DETECTION FRAMEWORK

The VIS in this study includes the IAS and change detection modules for plug inspection. The IAS is described in the previous section. In this section, the change detection modules are introduced.

Concretely speaking, the change detection framework is composed of three algorithm modules, which are named the object location, image alignment and similarity computation modules, respectively. It is noted that the input images are all preprocessed using histogram equalization and median filtering for enhancement and denoising, respectively. Since the two preprocessing algorithms are classical operations and commonly used in image processing systems, we do not describe them below. The diagram of the framework is shown in figure 3.

1) OBJECT LOCATION

After an image is captured by the IAS, it is processed by the object location module that is used to locate plugs precisely by providing their rectangular coordinates in the image. The algorithm in this module is "object detection" algorithm from the field of computer vision [26]. Here, we use it to meet the second challenge of plug inspection mentioned in section I. As displayed in the first module of figure 3, the located plug (shown in figure 4(a)) is surrounded by a white rectangle denoted with coordinates in the image plane. In the object location module, an object detection algorithm using a Convolutional Neural Network (CNN) is proposed to locate plugs. The algorithm is presented in section III.A.

2) IMAGE ALIGNMENT

Then, the image alignment module is used to align the images containing plugs in one inspection with the images containing the same plugs in the ground-truth dataset. It is necessary to perform image alignment considered the VPS's precision. The images with the same location denoted by the VPS during two inspections may contain different objects. Similarly, images containing the same objects may be denoted with different locations by the VPS during two inspections. As shown in figure 4, the left image (figure 4(a)) was captured during one inspection and it was denoted by the VPS as 130 kilometers plus 219 meters on the railway from Jinshan North to Haining West. The middle image (figure 4(b)) with the red border was the image denoted by the VPS as from the same location in the ground-truth dataset. Obviously, the two images denoted by the VPS as the same location are different images that contain different objects. The right image (figure 4(c)) with the green border was the aligned image that contains the same objects in the ground-truth dataset, but its location was 130 kilometers plus 226 meters. Hence, to ensure the applicability of the next module, an image alignment algorithm must be executed. The module can align images with pixel-level precision because the plugs belonging to some railways must be periodically detected according to the maintenance plans and the IAS captures images with almost the same shooting angles and distances. Therefore, although the inspections for the plugs of the same railway are scheduled at different times, similar images are obtained. Although there are many image alignment algorithms in the field of machine learning [27], the algorithm used in this module is designed according to the characteristics of plug images. The algorithm of this module can match the image

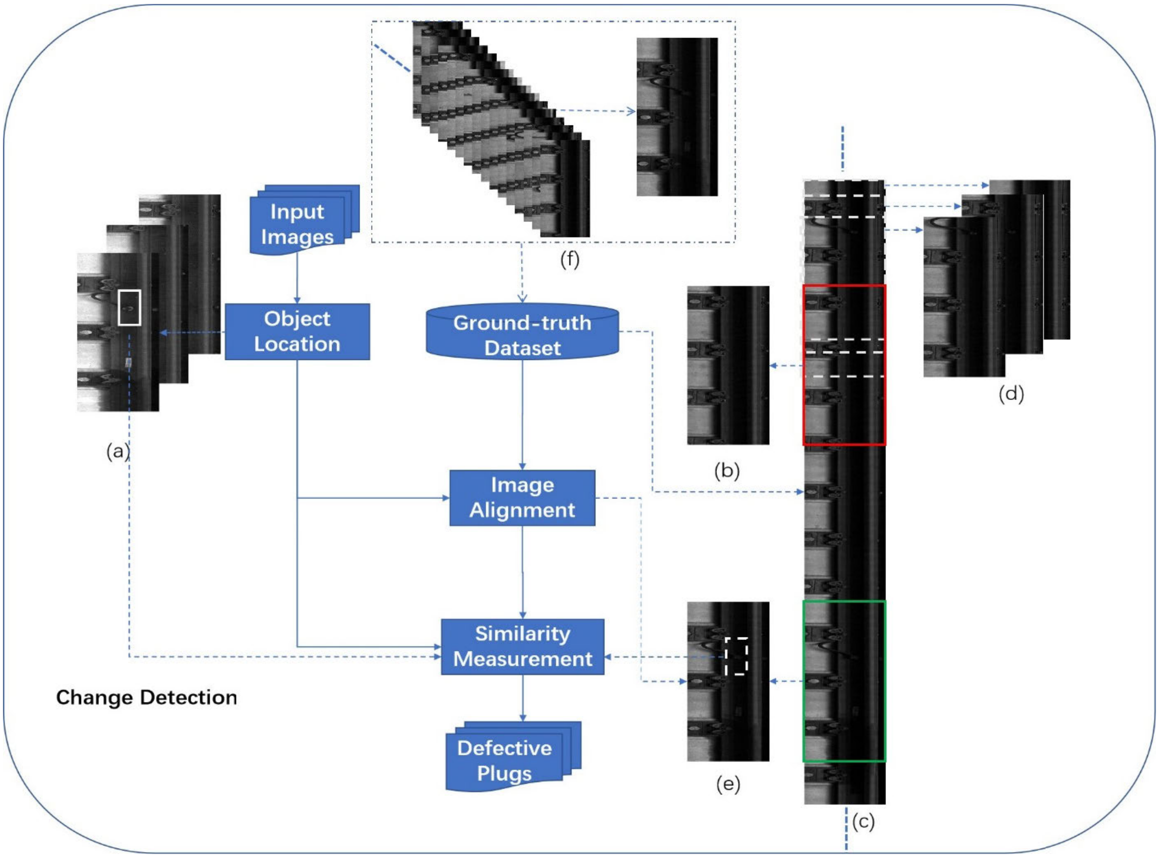


FIGURE 3. The change detection framework. The input image Is containing a located plug (denoted as 240 kilometers plus 610 meters) is shown in (a) and the image with the same location in the ground-truth dataset is shown in (b). Obviously, those images do not present the same scene. The fragment of joined image Ij is shown in (c). Some images in the series of In are shown in (d). The resulting image Ia aligned to the input image Is is shown in (e). A part of the database is shown in (f). The location of Ia can be considered to be 240 kilometers plus 614 meters. This is detailed in section III.B.

captured during an inspection to the image of the ground-truth dataset with pixel-level precision. Figure 4 shows that the plug image is precisely matched to the image in the plug ground-truth dataset. The algorithm will be presented in section III.B.

3) SIMILARITY MEASUREMENT

The similarity computation module is used to compute the similarity between the two plug regions in an inspection image and an aligned ground-truth image. The similarity measurement is conducted after all inspection images are processed directly. The inspection image with a smaller similarity value is ranked higher and the plugs in the image have a larger probability of having a defect. The details of the measurement are in section III.C.

4) GROUND-TRUTH DATASET

The plugs in the images contained in the ground-truth dataset are all perfect. In other words, those images are standard images. How can we build the dataset? There are two ways: first, when a new railway is just finished and its suitability for operation is tested, serial images could be taken and stored in the dataset; and second, when all the plugs in a railway are found to be perfect after manual inspection, the serial images containing those plugs in the railway could be stored in the dataset. Obviously, those serial images are indispensable in the image alignment and similarity measurement computation modules. As displayed in figure 3(c), some serial images

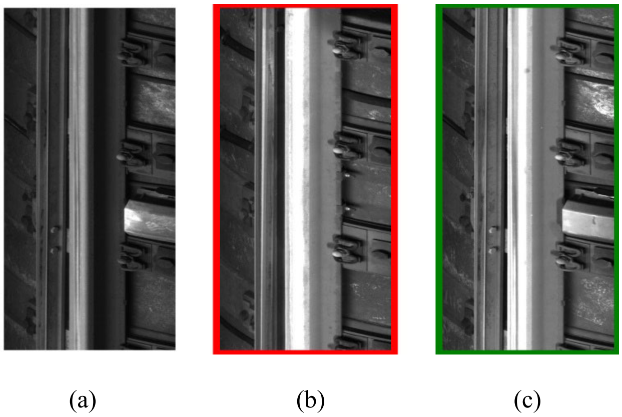


FIGURE 4. The necessity and feasibility of image alignment: (a) the image that was captured during one inspection and it was denoted by the VPS as 130 kilometers plus 219 meters; (b) the image with the red rectangle was the image denoted by the VPS as the same location in the ground-truth dataset; and (c) the image with the green rectangle was the aligned image that contains the same objects in the ground-truth dataset, but its location was 130 kilometers plus 226 meters.

in the ground-truth dataset are joined in succession and the aligned segment is denoted with a green rectangle.

III. CHANGE DETECTION

A. OBJECT LOCATION

The object detection algorithm utilized in this module to locate plugs has three steps: (1) ‘Region Proposal’ (RP) [28] provides some possible rectangle regions that may contain a plug, (2) a CNN is used to extract plug region features, and (3) a support vector machine (SVM) is used as a classifier to judge whether the region features represent a plug.

1) REGION PROPOSAL (RP)

The source image captured by the IAS is shown in figure 5(a). The region of the rail waist structures, as the input image for the object detection algorithm, is shown in figure 5(b). Obviously, the plug looks small and salient in figure 5(b), but the rest of the region looks smooth and monotonous. In other words, if an input image contains a plug, the input image should present a different texture. Because different textures can be distinguished in the frequency domain, the magnitude spectrum of the input image containing a plug (figure 5(b)) can have the magnitude spectrum of the average image subtracted from it, and the result is called the spectrum residual. Figure 5(c) is the average image and its pixel values are the mean pixel values of the 384956 rail waist structure regions that do not contain plugs in this work. Then, the spectrum residual is processed by the phase-holding IFFT and the resulting image will distinctly show the salient regions. Those salient regions are proposal regions where plugs may be contained, as displayed in figure 5(d). In the figure, the real region is denoted by an arrow. From the above, the algorithm proposed to conduct RP is named the spectrum residual region proposal (SRP) algorithm, and figure 6 shows the flowchart. The outputs of the SRP are some proposal regions. Those proposal regions, which are encircled by rectangular boxes, are then used as the input images to the following CNN.

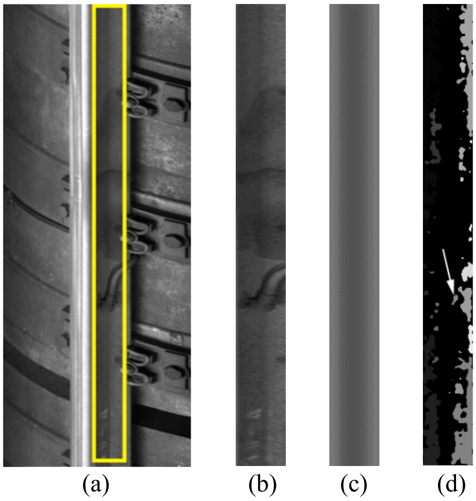
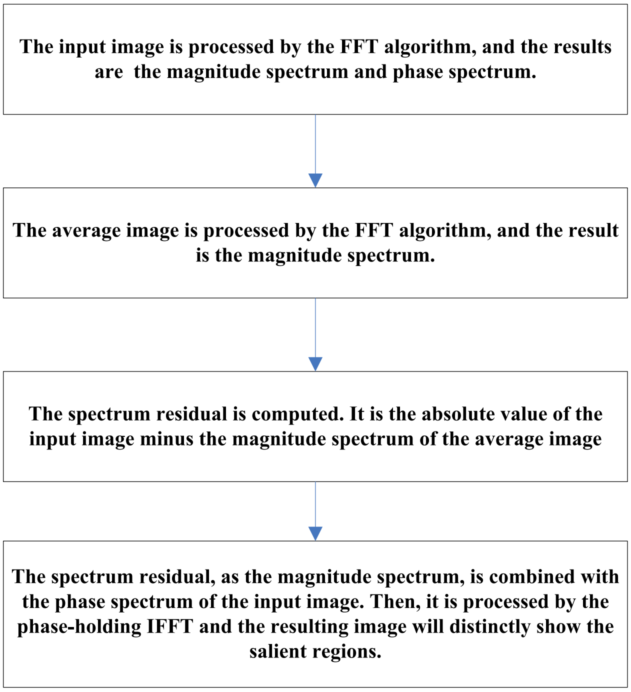


FIGURE 5. An example of an SRP: (a) source image, (b) the region of the rail waist structures (the region in the yellow rectangle in (a)), (c) the average image, and (d) the proposed regions.



The input image is processed by the FFT algorithm, and the results are the magnitude spectrum and phase spectrum.

The average image is processed by the FFT algorithm, and the result is the magnitude spectrum.

The spectrum residual is computed. It is the absolute value of the input image minus the magnitude spectrum of the average image

The spectrum residual, as the magnitude spectrum, is combined with the phase spectrum of the input image. Then, it is processed by the phase-holding IFFT and the resulting image will distinctly show the salient regions.

FIGURE 6. The flowchart of spectrum residual region proposal (SRP).

2) PLUG CNN

The CNN is designed for plug detection in this paper, and we name it the plug CNN (pCNN). The function of the pCNN is to extract the features of the input images, and its structure is shown in figure 7. Above all, the terminology of the pCNN in this paper is the same as those in the famous classical paper about the CNN [29]. The pCNN includes four convolution modules, and each module consists of a convolution layer, a nonlinear activation layer, a normalization layer and a pooling layer.

In detail, the input image is resized to 32*32 pixels, and 2 pixels are padded to surround the resized image. This is convenient for the convolution operation. Next, in the first convolution layer named ‘conv1’, the size of the receptive field, defined as the size of the region in the input that produces the feature [26], is 5*5, its dimension is 32 and the step size is 1. As the nonlinear activated layer, to accelerate the convergence of the pCNN, the ‘relu1’ function is used to get the maximum value of the convolution results. Then, local response normalization is applied to the results of relu1 to gain better generalization. The operation is named ‘norm1’ in the paper. Then, a pooling layer is used to get the maximum values of the previous step’s results in a 3*3 receptive field and a step size of 2. We name the layer ‘pool1’. In short, conv1, relu1, norm1 and pool1 build up the first convolution module of the pCNN. It is noted that the measurement unit is pixels in the description of the

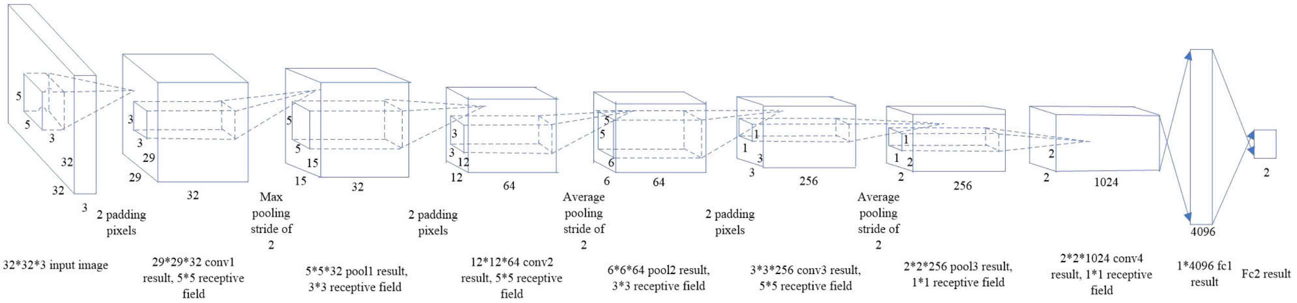

5
5
3
32
32
3
2 padding pixels
3
3
29
29
32
Max pooling stride of 2
5
5
15
15
32
2 padding pixels
3
3
12
12
64
Average pooling stride of 2
5
5
6
6
64
2 padding pixels
1
1
3
3
256
Average pooling stride of 2
1
1
2
2
256
2
2
1024
4096
2
32*32*3 input image
29*29*32 conv1 result, 5*5 receptive field
5*5*32 pool1 result, 3*3 receptive field
12*12*64 conv2 result, 5*5 receptive field
6*6*64 pool2 result, 3*3 receptive field
3*3*256 conv3 result, 5*5 receptive field
2*2*256 pool3 result, 1*1 receptive field
2*2*1024 conv4 result, 1*1 receptive field
1*4096 fc1 result
Fc2 result

FIGURE 7. The structure of the pCNN.

pCNN. Similarly, the following second convolution module of the pCNN can be described as follows: one convolution layer named 'conv2', the size of the receptive field is 5*5, the dimension is 64, there are 2 padding pixels and the step size is 1. Then, the 'relu2' function, which is the same as 'relu1', is used; local response normalization is applied to the results of relu2 and the operation is named 'norm2'. Then, the pooling layer is the same as pool1 and is named 'pool2'. Next, the following third convolution module of the pCNN is described as follows: one convolution layer named 'conv3', the size of the receptive field is 5*5, the dimension is 256, 2 padding pixels are used and the step size is 1. Then, the function 'relu3', which is the as same as 'relu1' and 'relu2', is used; local response normalization is applied to the results of relu3 and the operation is named 'norm3'. Then, a pooling layer is used to get the maximum values of the previous results in the 1*1 receptive field and the step size is 2. We name the layer 'pool3'. Next, the following forth convolution module of the pCNN is described as follows: one convolution layer named 'conv4', the size of the receptive field is 1*1, the dimension is 1024 and the step size is 1. Then, the 'relu4' function, which is as same as 'relu1', 'relu2' and 'relu3', is used; local response normalization is applied to the results of relu4 and the operation is named 'norm4'. Next, a fully connected layer named 'fc1' is used to realize the inner product of the previous results. Then, a dropout layer named 'drop1' is used to set the results of fc1 as 0 or 1 randomly. It can also gain better generalization for the CNN. The output of the pCNN in this work is the feature of the input image, and the feature dimension is 4096. Finally, a fully connected layer named 'fc2' is used to determine whether the input image contains a plug. This layer is only used in the pCNN training.

In figure 8(a), the input image containing a plug is used to illustrate the features extracted by the pCNN. The intermediate results are shown in figures 8 (b)∼(e). Those figures show that as the result of SPR, the input image is processed by the first to the fourth convolution modules in order and the results of the 4096-dimension features are sparse, as displayed in figure 8(f).

To train the pCNN, 6000 typical plug images are collected and resized to 32*32 as positive samples. In the same way, 6000 typical nonplus images are collected and resized to 32*32 as negative samples. Those images constitute the training dataset, and some samples are displayed in figures 9 (a)∼(b). Then, another 3000 positive samples and 3000 negative samples are collected similarly, and those images constitute the validation dataset. The training parameters are configured as follows: base learning rate of 0.01, momentum of 0.9, weight decay of 0.004, gamma of 0.1, step size of 20000 and a maximum number of iterations of 150000.

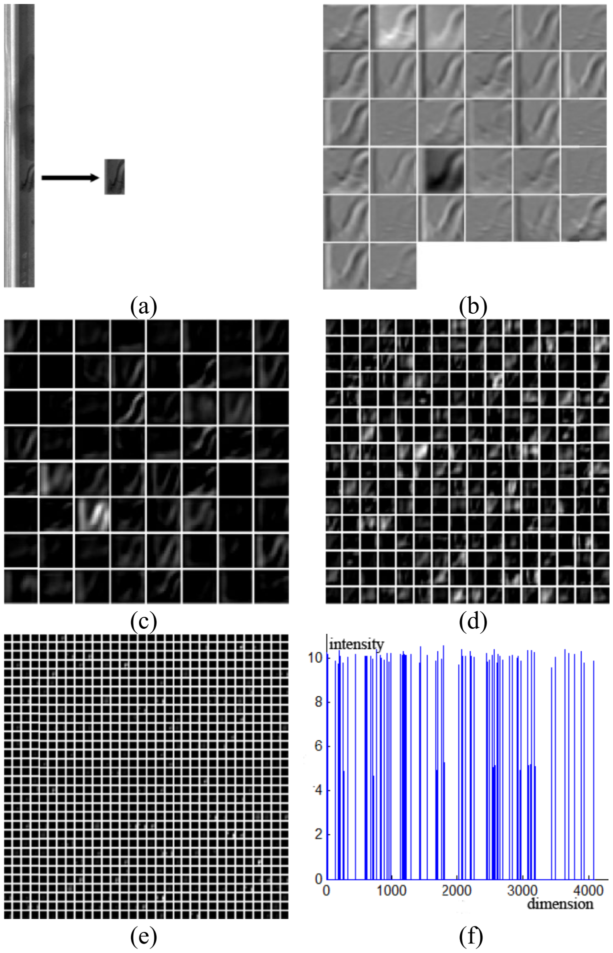


FIGURE 8. An example of feature extraction by the pCNN: (a) the input image containing a plug, (b∼e) intermediate results from the first convolution module to the fourth convolution module, and (f) the output of fc1 for feature extraction by the pCNN.

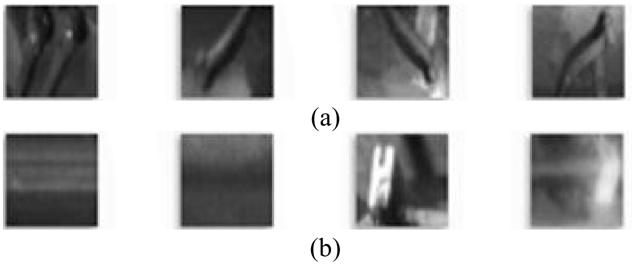


FIGURE 9. Some samples for training the pCNN: (a) positive samples, and (b) negative samples.

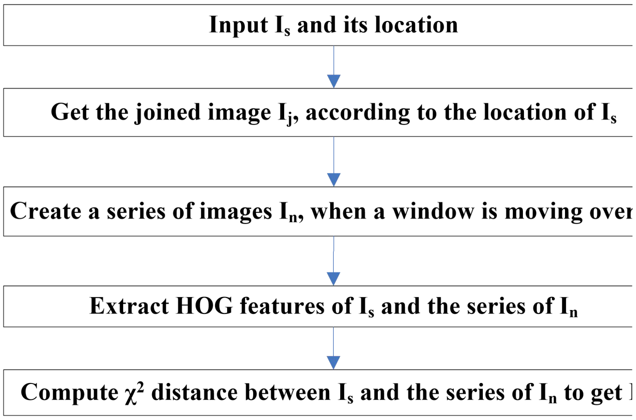

Input $\mathbf{I_s}$ and its location
Get the joined image $\mathbf{I_j}$, according to the location of $\mathbf{I_s}$
Create a series of images $\mathbf{I_n}$, when a window is moving over
Extract HOG features of $\mathbf{I_s}$ and the series of $\mathbf{I_n}$
Compute χ^2 distance between $\mathbf{I_s}$ and the series of $\mathbf{I_n}$ to get

FIGURE 10. The flow of the image alignment algorithm.

3) SVM

The SVM classification follows the pCNN. The input is the features such as figure 8(f) and the result is 0 or 1, which represents whether the input image contains a plug. In addition, the SVM is a linear classifier [30] realized using the libsvm toolbox [31]. To train the linear SVM, the features of fc1 are input after the pCNN is trained. In SVM training, 10-fold cross-validation is adopted and the regularization factor C is set as 2.

The experiments about the object location module are shown in section IV.A.

B. IMAGE ALIGNMENT

Figure 10 shows the flow of the image alignment process. At first, the location (denoted as kilometers plus meters) of an image is provided by the VPS. The image is processed by the previous module (object location module) to ascertain whether a plug is contained in the image. The image is named $\mathbf{I_s}$ (shown in figure 3(a)). Second, series of images in the ground-truth dataset are retrieved and joined to an image named $\mathbf{I_j}$ (shown in figure 3(c)). Those series of images' locations (also denoted as kilometers plus meters) are all within a 20 meters range of the location of $\mathbf{I_s}$. The range of 20 meters can be deemed to be the error of the VPS. In our IAS, ten images can cover twenty meters of continuous track without overlap. Therefore, $\mathbf{I_j}$, the joined image, is 21 times the height and the same width as $\mathbf{I_s}$. Third, the rectangular window that is same size as $\mathbf{I_s}$ moves pixel by pixel in the joined image $\mathbf{I_j}$ from head to end (shown as the red rectangle in figure 3(c)). During the moving, a new image named $\mathbf{I_n}$ is created when the window moves a pixel. The new image is the same as the region of $\mathbf{I_j}$ under the window. Obviously, we can get a series of $\mathbf{I_n}$ (shown in figure 3(d)), when the window is moving. Fourth, the histogram of oriented gradients (HOG) features are extracted from $\mathbf{I_n}$ since $\mathbf{I_n}$ has more lines and right angles [32]. In addition, the HOG features of $\mathbf{I_s}$ are also computed. HOG is a classical and stable feature descriptor that is used for object detection in the field of computer vision. It presents a normalized histogram that is obtained by computing the histogram of oriented gradients in a local image region. Fifth, the $\chi 2$ distances between $\mathbf{I_s}$ and those created images (the series of $\mathbf{I_n}$) are computed [33] in the space of the HOG histograms. The image in the series of $\mathbf{I_n}$, named $\mathbf{I_a}$, with the minimum distance to $\mathbf{I_s}$ is the aligned image in the ground-truth dataset.

The χ^2 distance $\mathbf{d(x, y)}$ is the distance between two histograms $x = [x_1, \cdots, x_n]$ and $y = [y_1, \cdots, y_n]$ with both having $\mathbf{n}$ bins. Moreover, both histograms are normalized, i.e., their entries sum up to one. The HOG features used in this work are just normalized histograms, and so we can denote the features of the two images as $\boldsymbol{x}$ and $\boldsymbol{y}$. The distance $\mathbf{d}$ is usually defined as

$$d(x, y) = \frac{1}{2} \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}$$

It is often used in computer vision to compute the distances between two images.

Figure 3 illustrates the plug image alignment algorithm. The input image $\mathbf{I_s}$ containing a located plug (denoted as 240 kilometers plus 610 meters) is shown in figure 3(a) and the image with the same location in the ground-truth dataset is shown in figure 3(b). Obviously, those images do not present the same scene. The fragment of joined image $\mathbf{I_j}$ is shown in figure 3(c). Some images in the series of $\mathbf{I_n}$ are shown in figure 3(d). The resulting image $\mathbf{I_a}$ aligned to the input image $\mathbf{I_s}$ is shown in figure 3(e). The location of $\mathbf{I_a}$ can be considered to be 240 kilometers plus 614 meters. The more experiments about the image alignment module are shown in section IV.B.

C. SIMILARITY MEASUREMENT

From the previous steps, the aligned images in the ground-truth dataset are created and we get pairs of images—inspection images containing plugs and their aligned images in the ground-truth dataset. In this module, the similarity measurements for same plug regions in those pairs of images, as displayed in figure 3, are computed with their local binary pattern (LBP) features [34]. The LBP is a classical and stable operator to extract an image's statistical and structural features. It describes the statistical properties and texture structure in the form of a normalized histogram. That is, in figure 3, the white solid rectangular region in the inspection image (shown in figure 3(a)) and the white dashed rectangular region in the aligned image (shown in figure 3(e)) are the corresponding area to be measured. In addition, the similarity measurement is also the $\chi 2$ distance in the space of the LBP histograms. An example is shown in Figure 11, where the LBP histograms of the images in the ground-truth dataset and those captured during inspection were calculated. The $\chi 2$ distance between (a) and (b) was 0.0248 and that between (c) and (d) was 0.0349. The image with a defective plug has larger distance value. Obviously, the similarity decreases as the distance becomes larger, and we may replace the similarity with respect to the distance in the following. Then, the values of the measurement metric, the $\chi 2$ distances, are sorted from the largest to the smallest. Obviously, a larger value means that it is more likely that the inspection image contains defective plugs. Finally, the inspection images containing defective plugs are sorted according to the possibility that they contain a defect. Those images are the results we wanted. The experiments of this module are shown in section IV.C.

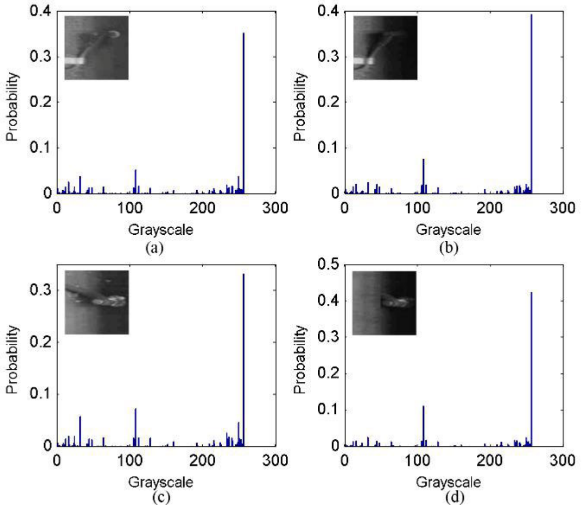


FIGURE 11. LBP histograms of plugs. (a) A plug in the ground-truth dataset, (b) the plug in (a) with no defect captured during inspection, (c) a plug in the ground-truth dataset, and (d) the plug in (c) with an artificial defect.

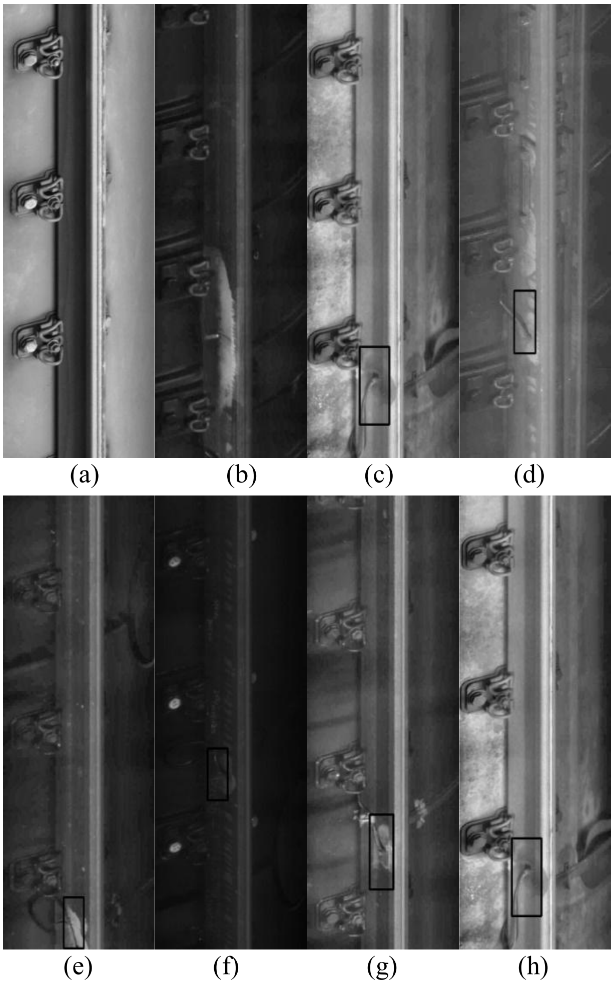


FIGURE 12. Some typical results of our object location module: (a) the image containing no plug was captured in a normal environment; (b) the image containing no plug looks smudged due to corrosion and scratches; (c) the image containing a plug was captured in a normal environment; (d) the image containing a plug was captured in a complex environment, such as a rail switch; (e) the image contains a half plug; (f) the image containing a plug was captured in insufficient illumination and it looks smudged due to corrosion and scratches; (g) the image contains a plug that was also manually marked by trackwalkers; and (h) the image containing a plug that looks a little motion blurred.

IV. EXPERIMENTS

In this section, we illustrate some experiment results about the change detection framework. The operating environment is a computing workstation that is equipped with dual Intel Xeon E5-2680V4 (14 kernels, 2.4 GHz) CPUs and 256G DDR4 RAM.

A. OBJECTION LOCATION

We illustrate some typical experimental results for the objection location module in Figures 12∼13. The parameter settings are the same as those described in section III.A.

Figure 12 shows some typical results. In figure 12(a), the image containing no plug was captured in a normal environment and the location result shows that no object was located. In figure 12(b), the image containing no plug looks smudged due to corrosion and scratches and the location result shows that no object was located. In figure 12(c), the image containing a plug was captured in a normal environment and the location result shows that the object enclosed by a rectangle was correctly located. In figure 12(d), the image containing a plug was captured in a complex environment, such as a rail switch, and the location result shows that the object enclosed by a rectangle was correctly located. In figure 12(e), the image contains a half plug and the location result shows that the half plug enclosed by a rectangle was correctly located. In figure 12(f), the image containing a plug is captured in insufficient illumination and it looks to be smudged due to corrosion and scratches. The location result shows that the object enclosed by a rectangle was correctly located. In figure 12(g), the image contains a

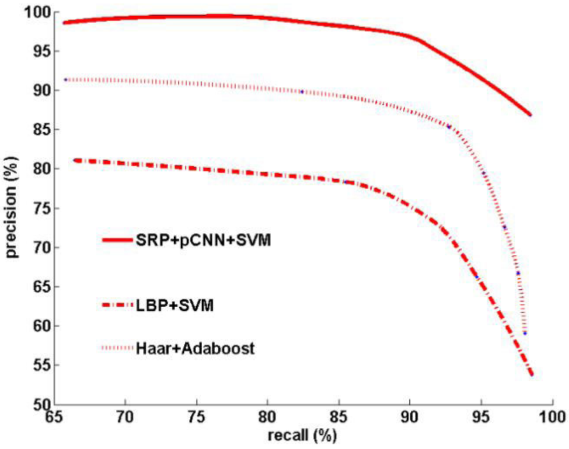


FIGURE 13. Precision-recall curves.

plug that was also manually marked by trackwalkers and the location result shows that the located enclosed by a rectangle was correctly located. In figure 12(h), the image containing a plug looks a little motion blurred and the location result shows that the object enclosed by a rectangle was correctly located. Moreover, the precision-recall curve also shows that the object location module in this paper (SRP+pCNN+SVM) gets better results than those in our previous work [24], as represented by the curves denoted as LBP+SVM and Haar+Adaboost, as displayed in figure 13. Our previous article [25] presents more details about the object location module in this paper (SRP+pCNN+SVM). Obviously, deep learning results perform better than traditional machine learning results.

B. IMAGE ALIGNMENT

In this section, we illustrate the experimental results for the image alignment module. As shown in figure 14, the upper-left image (figure 14(a)) was captured during one inspection and it was denoted by the VPS as 239 kilometers plus 545 meters on the railway from Changzhou to Zhenjiang. The upper-middle image (figure 14(b)) was the image denoted by the VPS as the same location in the ground-truth dataset. Obviously, the two images denoted by the VPS as the same location are different images that contain different objects. The upper-right image (figure 14(c)) was the aligned image that contains the same objects in the ground-truth dataset, but its location was 239 kilometers plus 541 meters. Similarly, the bottom-left image (figure 14(d)) was captured during one inspection and it was denoted by the VPS as 127 kilometers plus 710 meters on the railway from Yuanping to Xinzhou. The bottom-middle image (figure 14(e)) was the image denoted by the VPS as the same location in the ground-truth dataset. Obviously, the two images denoted by the VPS as the same location are different images that contain different objects. The bottom-right image (figure 14(f)) was the aligned image that contains the same objects in the ground-truth dataset, but its location was 127 kilometers plus 712 meters. Using this method, we can get 100% alignment accuracy.

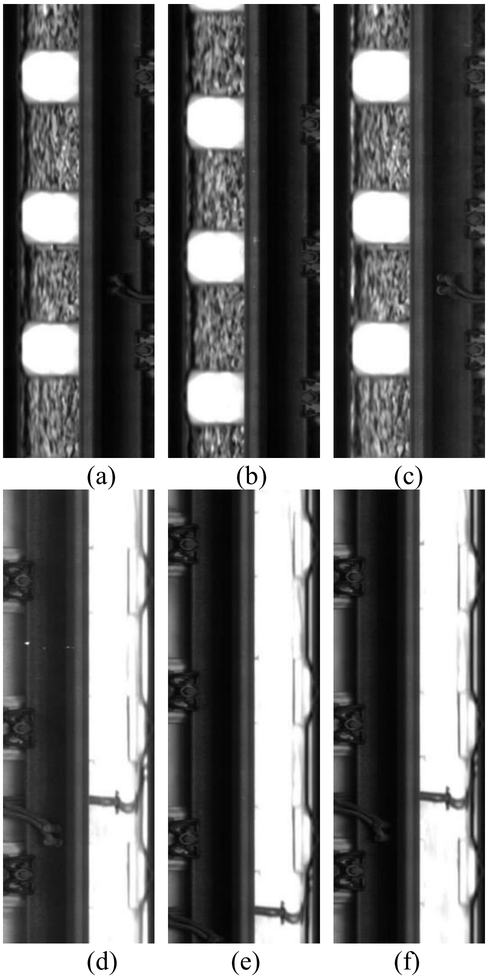


FIGURE 14. Two experiments on the image alignment module: (a) the image was captured during one inspection and it was denoted by the VPS as 239 kilometers plus 545 meters; (b) the image denoted by the VPS as the same location in the ground-truth dataset; (c) the image was the aligned image that contains the same objects in the ground-truth dataset, but its location was 239 kilometers plus 541 meters; (d) the image was captured during one inspection and it was denoted by the VPS as 127 kilometers plus 710 meters; (e) the image denoted by the VPS as the same location in the ground-truth dataset; and (f) the image was the aligned image that contains the same objects in the ground-truth dataset, but its location was 127 kilometers plus 712 meters.

C. SIMILARITY MEASUREMENT

In this section, we illustrate some experimental results about the similarity measurement module. Those results are also the final results for the plug inspections and the images were captured along the Shanghai–Hangzhou high-speed railway line. It is noteworthy that those experiments are processed by the VIS installed on the inspection train owned by the China Academy of Railway Sciences. We present the first four results measured by the $\chi 2$ distances and sorted the distances from the largest to smallest. Figure 15 shows the plug inspection results. For example, figure 15(a) shows that the plug cable was moved compared with the aligned image in the ground-truth dataset, and this means that the plug may be

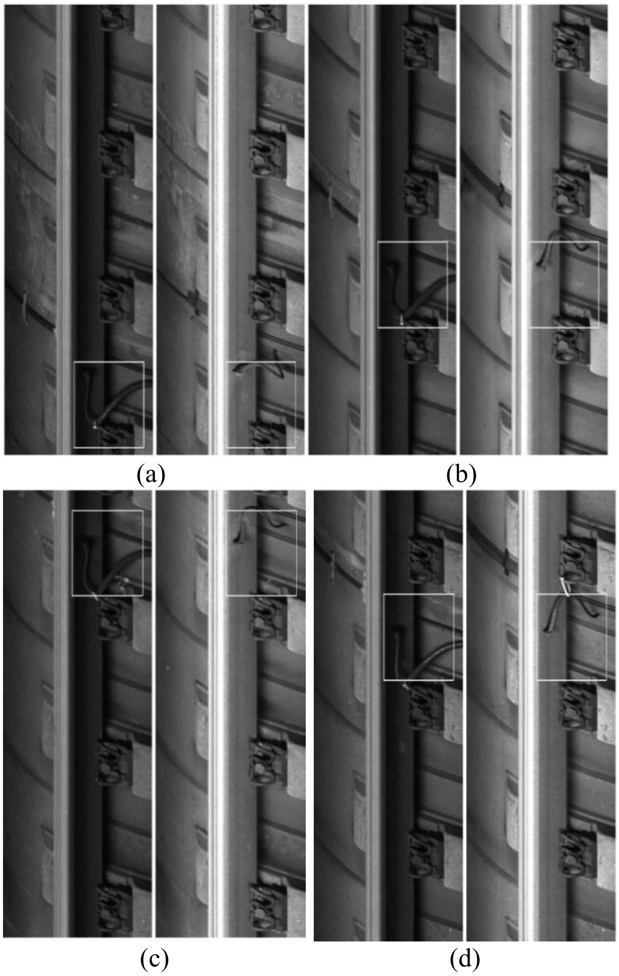


FIGURE 15. The plug inspection results: (a)~(d) show that the plug cable was moved compared with the aligned image in the ground-truth dataset.

defective with the maximum possibility. It is also noted that the aligned image in the ground-truth dataset is the left image and the inspection image is the right one in figure 15(a). Comparing the rectangles in above figures, the rectangles surrounding the plugs in this figure are enlarged to avoid missing some details of those plugs. This is the same with figures 15(b)~(d). Obviously, those plugs were all touched by something or somebody, and this would very likely cause some defects since plug tail cables could be broken off easily. With those results, professional maintenance engineers can judge whether the plugs need to be maintained. Using this means, we also provide some possible defective plug positions when railway maintenance managers make a predictive maintenance plan for the next maintenance period.

To evaluate the performance of the whole system, we chose a ground-truth dataset with 191,530 images in which 9,276 included a plug. Since it is impossible to obtain defective plugs via destructive tests on real railways, every time five images were randomly picked from the 9,276 plug images, defective plugs were imitated. The precision-recall curve of the 10 training and calculation sessions of the Top-1 accuracy and the Top-20 accuracy are shown in Figure 16. In the field of machine learning, Top-N accuracy means that the correct result gets to be in the Top-N probabilities for it to count as "correct". The processing time for one image using these algorithm modules was approximately 0.1 second.

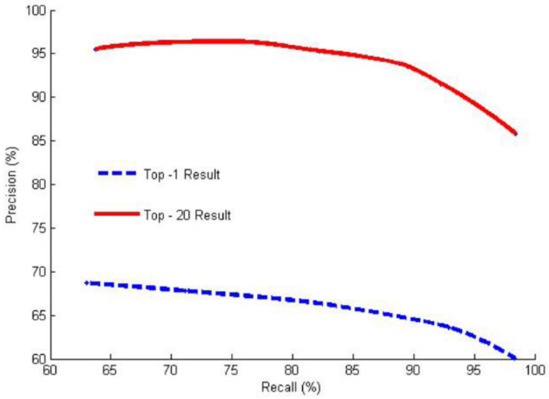


FIGURE 16. Precision-recall curve of the change detection framework.

In most cases, if a defective plug was found, it must be replaced by a new one instantly. So, there will be no defective plug in this railway section for a very long time. Therefore, the experiment (Fig. 16) with imitated and augmented defective plugs was designed to provide some statistical results in normative rule of academic standard. As an industry application in practical use, the top-20 results (those are the first twenty plug images presented by our system), sorted by defect probability from large to small, must include the defective ones, if defective plugs exist in this railway section. The top-1 result is not used in practice. In fact, the framework has passed the practice tests of 1,1905.5 km and 150 hours.

V. DISCUSSION AND CONCLUSION

The VIS with the change detection framework is the first visual system that can perform railway plug inspection automatically and intelligently. Regarding the algorithm modules, the precision-recall curve of the object location module looks better than those of the other methods that we used before [24], and the image alignment module can also achieve 100% alignment accuracy in our experiments; however, the series of inspection images look almost the same except for those images containing plugs. In the last module experiments, we find that plugs that are very likely to be defective can be selected from a large number of plug images automatically and intelligently. Then, that small number of selected plug images will be reviewed manually, and this is practically easy work. It is also the reason why we do not provide the final system accuracy because that small number of recommended images should be reviewed by professional maintenance engineers to finally confirm whether those plugs are defective or not. In fact, during practical inspection work, the speed of a test train may be 120 km/h (kilometers/hour), and the run time is about three hours for every inspection. Therefore, the total number of images is approximately 180,000 since one image can cover about two meters of track. If the object location module is the only one adopted to select plug images, and the states of those plug images are then manually judged by maintenance engineers, approximately 9,000 plug images will be reviewed manually after every inspection. Using our system with the framework, after a large number of experiments, we have determined that the first twenty plug images, sorted by defect probability from large to small, are enough to be reviewed manually. Thus, the VIS with the change detection framework can tremendously alleviate maintenance engineers' work compared with traditional manual detection.

The change detection framework can be implemented in various ways according to significantly different applications, which sometimes makes it difficult to compare algorithms directly [23]. Compared to the change detection in other fields, our framework has the following characteristics. (1) The specific image capture approach. In our system, the image sensors were fixed on a moving train, and so the captured images were spatially continuous and temporally discrete. In addition, there was no deformation of these images because of the fixed relative position of the camera and the tracks. For video surveillance, the image sensors were always fixed, and the acquired images were temporally continuous for a static scene. In the remote sensing field, the images are large scale and have greatly varying resolutions compared to our system. Furthermore, medical images, such as MRI images, show some image deformation due to some individual differences and body movements. (2) The specific alignment algorithm. In our framework, the alignment is a kind of image location method from image series. However, for video surveillance, the Gaussian mixture model was used for static scene recovery [21]. Meanwhile, for remote sensing and medical images, to perform image alignment, some elastic registration algorithms should be used to deal with the deformed images [20], [22]

Our study is the first practical system for plug inspection, and so there are hardly any images containing defective plugs collected by us. Therefore, just at the present stage, the number of defective plugs is too scarce to train a classifier that can classify defective plugs directly from large numbers of inspection images. However, since the massive construction of HSR urgently needs more efficient maintenance work, the VIS with the change detection framework can be further developed to satisfy the practical work. Considering the safety of the system, we prefer some reliable and stable algorithms, such as the HOG and LBP, which were used widely before. In a similar way, the deep neural network we used is concise because the computation speed and stability must be considered in industrial application. Although some of the latest networks are tested, such as Senet [35], the concise network, the pCNN, is found to be the optimal structure with respect to the system efficiency, maintainability and stability.

In addition, compared with other railway infrastructures, such as fasteners [9], [11], the number of plugs is much less, and thus the number of defective plugs is less. In practice, there is often up to one or two defective plugs in one inspection. This means that the ratio (the number of abnormal plugs: total plugs) is very imbalanced at over 1: 4500, and the data are extremely skewed. Therefore, even if detective plugs were collected for many years, the number of detective plugs might not be enough to train the classifier mentioned above. It is the open challenge named 'Learning from Imbalanced Data' in the machine learning field [36]. Moreover, because even a few defective plugs may cause a potential safety hazard, a stable and high-performance system is required. Thus, we propose the change detection framework to bypass the problem and it does work well in practice. In summary, to solve the problem of imbalanced samples in practice, we establish the framework of a visual inspection system for railway plug defects. Using the idea of change detection, the framework includes three algorithm modules, which are named the object location, image alignment and similarity measurement modules. In the view of application studies, it could be considered as the most important "novelty" of the paper and it has been used to solve the actual problems in railway inspections reliably. The framework has passed field tests, and it is also found in this paper that it conforms to the special section subject of this journal. In addition, the change detection framework also presents another approach for the inspection of many railway infrastructures, such as balises, rail surface defects, and catenary support devices. The limitations of this technique are as follows: 1) building a ground-truth database needs to consume lots of resources, including time, labor costs and memory storage; and 2) if the IAS camera shooting angle is changed in one inspection, although it may hardly happen, the final results could be worse because the image alignment module cannot work well in this case.

As far as we know, there are still no other study teams that have published a paper on automatic inspection systems for railway plug defects in influential journals or conference proceedings. In this paper, we propose the change detection framework used in our VIS for railway plug inspection for the first time. The framework includes the ground-truth dataset and three algorithm modules, which are the object detection, image alignment and similarity measurement modules. The results of the experiments showed that the system can detect the defective plugs with high accuracy and that can improve the efficiency of railway inspection. The VIS embedded with the framework has already been authorized by the China Railway Corporation and it will be equipped in many inspection trains belonging to many local railway corporations in China.

REFERENCES

[1] D. A. Pagano, "Method and apparatus for detecting internal rail defects," U.S. Patent 9 010 186, Apr. 21, 2015.

[2] Z. Yuhua, X. Guiyang, L. Pei, S. Yongsheng, and H. Xiaoyan, "Key technology to autonomous ultrasonic detection system of rail flaw detection car," *China Railway Sci.*, vol. 36, no. 5, pp. 131–136, 2015.

[3] S. Mariani, T. Nguyen, X. Zhu, and F. L. di Scalea, "Field test performance of noncontact ultrasonic rail inspection system," *J. Transp. Eng., A, Syst.*, vol. 143, no. 5, May 2017, Art. no. 04017007.

[4] P. Chandran, M. Rantatalo, J. Odelius, H. Lind, and S. M. Famurewa, "Train-based differential eddy current sensor system for rail fastener detection," *Meas. Sci. Technol.*, vol. 30, no. 12, Dec. 2019, Art. no. 125105.

[5] X. Li, B. Gao, W. L. Woo, G. Y. Tian, X. Qiu, and L. Gu, "Quantitative surface crack evaluation based on eddy current pulsed thermography," *IEEE Sensors J.*, vol. 17, no. 2, pp. 412–421, Jan. 2017.

[6] X. Zhang, Y. Cui, Y. Wang, M. Sun, and H. Hu, "An improved AE detection method of rail defect based on multi-level ANC with VSS-LMS," *Mech. Syst. Signal Process.*, vol. 99, pp. 420–433, Jan. 2018.

[7] Y. Wu, Y. Qin, Z. Wang, and L. Jia, "A UAV-based visual inspection method for rail surface defects," *Appl. Sci.*, vol. 8, no. 7, p. 1028, Jun. 2018.

[8] H. Zhang, X. Jin, Q. M. J. Wu, Y. Wang, Z. He, and Y. Yang, "Automatic visual detection system of railway surface defects with curvature filter and improved Gaussian mixture model," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 7, pp. 1593–1608, Jul. 2018.

[9] Q. Mao, H. Cui, Q. Hu, and X. Ren, "A rigorous fastener inspection approach for high-speed railway from structured light sensors," *ISPRS J. Photogramm. Remote Sens.*, vol. 143, pp. 249–267, Sep. 2018.

[10] Q. Han, S. Wang, Y. Fang, L. Wang, X. Du, H. Li, Q. He, and Q. Feng, "A rail fastener tightness detection approach using multi-source visual sensor," *Sensors*, vol. 20, no. 5, p. 1367, Mar. 2020.

[11] P. Dai, X. Du, S. Wang, Z. Gu, and Y. Ma, "Rail fastener automatic recognition method in complex background," *Proc. SPIE*, vol. 10806, Aug. 2018, Art. no. 1080625.

[12] L. Ramana, W. Choi, and Y.-J. Cha, "Fully automated vision-based loosened bolt detection using the Viola–Jones algorithm," *Struct. Health Monitor.*, vol. 18, no. 2, pp. 422–434, 2019.

[13] Z.-W. Li, X.-Z. Liu, and Y.-L. He, "Identification of temperature-induced deformation for HSR slab track using track geometry measurement data," *Sensors*, vol. 19, no. 24, p. 5446, Dec. 2019.

[14] K. P. Rishitha, J. R. Goud, T. V. Kumar, and S. Suryanarayana, "Automatic detection of squats in railway track," *J. Emerg. Technol. Innov. Res.*, vol. 5, no. 4, pp. 623–625, Apr. 2018.

[15] D. Zhan, D. Jing, M. Wu, D. Zhang, L. Yu, and T. Chen, "An accurate and efficient vision measurement approach for railway catenary geometry parameters," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 12, pp. 2841–2853, Dec. 2018.

[16] J. Zhong, Z. Liu, Z. Han, Y. Han, and W. Zhang, "A CNN-based defect inspection method for catenary split pins in high-speed railway," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2849–2860, Aug. 2019.

[17] 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 Trans. Instrum. Meas.*, vol. 67, no. 2, pp. 257–269, Feb. 2018.

[18] 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.

[19] Z. Zhu, "Change detection using landsat time series: A review of frequencies, preprocessing, algorithms, and applications," *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 370–384, Aug. 2017.

[20] M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, "Graph-based registration, change detection, and classification in very high resolution multitemporal remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 2940–2951, Jul. 2016.

[21] K. Goyal and J. Singhai, "Review of background subtraction methods using Gaussian mixture model for video surveillance systems," *Artif. Intell. Rev.*, vol. 50, no. 2, pp. 241–259, Aug. 2018.

[22] Ž. Lesjak, F. Pernuš, B. Likar, and Ž. Špiclin, "Validation of white-matter lesion change detection methods on a novel publicly available MRI image database," *Neuroinformatics*, vol. 14, no. 4, pp. 403–420, Oct. 2016.

[23] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 339–367, May 2017.

[24] X. Du, "Automatic location algorithms for railway plugs based on compute vision technology," *Railway Signaling Commun.*, vol. 52, no. 9, pp. 68–72, 2016.

[25] X. Du, P. Dai, Y. Li, Y. Cheng, S. Wang, Q. Han, and H. Wang, "Automatic detection algorithm for rail plugs based on deep learning," *China Railway Sci.*, vol. 38, no. 3, pp. 89–96, 2017.

[26] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[27] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019.

[28] 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.

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

[30] C. Lee and S. Woo, "Linear classifier design in the weight space," *Pattern Recognit.*, vol. 88, pp. 210–222, Apr. 2019.

[31] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.

[32] İ. Meşecan, İ. Ö. Bucak, and B. Çiço, "Comparison of histograms of oriented gradients (HOG) and n-row average subtraction (nRAS) using GprMax," *Microprocessors Microsyst.*, vol. 63, pp. 140–146, Nov. 2018.

[33] D. Yadolah, "Chi-square distance," in *The Concise Encyclopedia of Statistics*. New York, NY, USA: Springer, 2008, pp. 68–70.

[34] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikainen, "Median robust extended local binary pattern for texture classification," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1368–1381, Mar. 2016.

[35] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[36] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 1–12, 2016.

XINYU DU received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012. He is currently a Researcher with the Infrastructure Inspection Institute, China Academy of Railway Sciences, Beijing, China. His research interests include image processing and measurement.

YU CHENG received the M.S. degree from Beijing Jiaotong University, Beijing, China, in 2014. He is currently a Research Associate with the Infrastructure Inspection Research Institute, China Academy of Railway Sciences Corporation Ltd., Beijing. His current research interests include track inspection technology, embedded data acquisition and processing technology, and system safety verification technology in railway systems.

ZICHEN GU received the M.S. degree from the University of Bristol, U.K., in 2012. He is currently an Engineer with the Infrastructure Inspection Research Institute, China Academy of Railway Sciences Corporation Ltd., Beijing, China. His research interests include image recognition and object detection.