

Deep Learning and Machine Vision-Based Inspection of Rail Surface Defects

Hongfei Yang^{ID}, Yanzhang Wang^{ID}, Jiyong Hu^{ID}, Jiatang He^{ID}, Zongwei Yao^{ID}, and Qiushi Bi^{ID}

Abstract—Surface defects are usually the early phenomenon of rail failure, which threatens the safety of railroad transportation critically, and the timely detection of surface defects helps to eliminate the potential risk of rail and reduce the chance of railroad safety accidents. The existing methods of detecting surface defects on rails suffer from a large performance degradation in the application of rails containing pollutions such as rust and oil. Therefore, this article proposes a multilevel, end-to-end fast rail surface defect detection method. First, rail extraction was performed based on the stability of the standard deviation of the edge pixels. Then, differential box-counting (DBC) and GrabCut algorithm are then combined for defect segmentation to boost the speed and accuracy of extracting complex defects. Finally, YOLO v2 is used to precisely locate and detect defects. The experimental results show that the proposed method performs well, with an average accuracy of 97.11%, an average recall of 96.10%, and an average frame rate of 0.0064 s. In addition, the proposed method offers a high robustness in the tests of different use cases.

Index Terms—CNN, defect detection, entropy penalty factor, GrabCut, railroad surface.

I. INTRODUCTION

AFFECTED by the natural environment and long-term service, the rail is subject to a variety of failures during service, which, if not addressed in a timely manner, would lead to train accidents that endanger the personal and property safety of passengers [1]. Hence, it is essential to establish an accurate, effective, and reliable detection method for surface defects to evaluate the status of rails and prevent transportation accidents. Several objective constraints make rapid and accurate detection of track surface quality difficult, which can be summarized in the following three aspects [2], [3]: 1) the complexity of illumination condition can bring unexpected noise and result in image distortion; 2) the reflective properties of the track surface are not homogeneous, i.e., different track surface morphologies and wear status lead to difficulties in

Manuscript received August 28, 2021; revised November 28, 2021; accepted December 13, 2021. Date of publication December 24, 2021; date of current version March 3, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 51875232, Grant 51875233, and Grant 42074216, and in part by the Natural Science Foundation of Jilin Province under Grant 20210101382JC. The Associate Editor coordinating the review process was Shovan Barma. (*Corresponding authors:* Zongwei Yao; Qiushi Bi.)

Hongfei Yang and Yanzhang Wang are with the College of Instrumentation and Electrical Engineering, Jilin University, Changchun, Jilin 130061, China (e-mail: yanghf20@mails.jlu.edu.cn; yanzhang@jlu.edu.cn).

Jiyong Hu and Jiatang He are with FAW-Volkswagen Automobile Company, Ltd., Changchun, Jilin 130011, China (e-mail: jiyong.hu@faw-vw.com; Jiatang.he@faw-vw.com).

Zongwei Yao and Qiushi Bi are with the School of Mechanical and Aerospace Engineering, Jilin University, Changchun, Jilin 130022, China (e-mail: yzw@jlu.edu.cn; bqs@jlu.edu.cn).

Digital Object Identifier 10.1109/TIM.2021.3138498

image segmentation; and 3) the pollutions such as rust and oil on the track surfaces are similar to defects, and the noise generated is usually difficult to eliminate precisely.

The commonly used detection method can be categorized as three main types. First, the manual method greatly relies on the operator's working experience as they had to detect the surface defects and failures by distinguishing the response of knocking on the rails and visual judgments [4], [5]. This type of method is highly subjective and the inspection results are highly dependent on the work experience, skill and expertise, physical/mental state, and working environment of the inspector. Second, the sensor-based method, such as eddy current [6], ultrasonic wave [7], infrared ray [8], and laser [9], which requires high reliability and accuracy of the sensor itself and is poorly facilitated in operation. Third, the machine-vision-based method took the advantage of image processing to accomplish an accurate and rapid detection [10], [11].

He *et al.* [12] introduced a semi-supervised convolutional neural network (CNN) for feature extraction and input the features into a classifier for defect classification. Nataraajan *et al.* [13] proposed a support vector machines (SVM) classifier to classify the surface defects of rolled steel. Zhong *et al.* [14] improved a deep CNN with a new anchoring mechanism to generate suitable candidate frames for objects and then combined multilevel features for cotter pin defect detection. He *et al.* [15] presented a multilevel feature fusion network for steel plate defect detection. Wang *et al.* [16] suggested a segmentation method with CNN as the feature extractor and random forest as the classifier. Masci *et al.* [17] established a method for classifying steel defects based on maximum pooling CNN. Soukup and Huber-Mrk [18] posed a classical CNN trained in a purely supervised manner and explored the impact of regularization methods.

In summary, machine vision and neural networks have been extensively used in industrial detection, but comparatively a few research has been performed on the detection of rail surface defects with complex noise and multiple samples. Therefore, a multilevel, end-to-end rail surface defect detection method is proposed to automate the segmentation of rail surface defects and improve the safety of rail service.

Specifically, this study is conducted in four parts. The first part is sample acquisition and classification of rail surface defects, the second part is rail accurate extraction, the third part is defect segmentation, and the fourth part is defect pinpointing. In the first part, the design of a rail image acquisition device and the classification of common types of rail image noise are focused. In the second part, a pseudo-edge

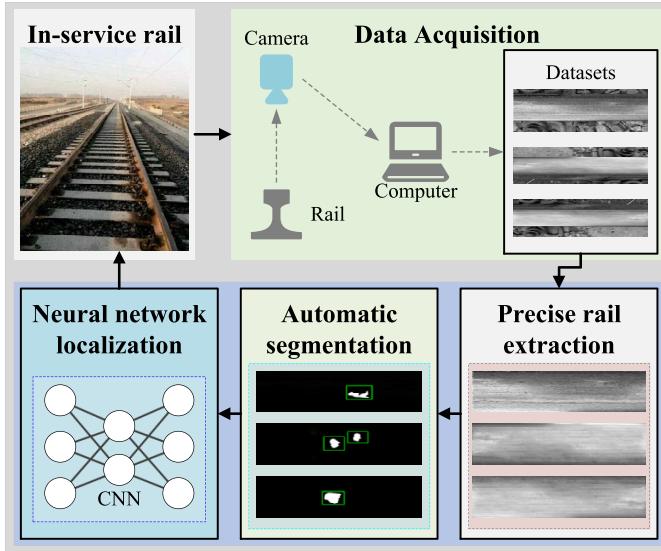


Fig. 1. Process of the proposed method.

preliminary processing method is proposed to optimize the effect of Canny. In the third part, the differential box-counting-GrabCut (DBC-GrabCut) is proposed to automate segmentation and improve segmentation accuracy simultaneously. Finally, the YOLO v2 is employed for the precise location of defects on the rail surface. Furthermore, experiments are carried out to analyze and evaluate the effectiveness, accuracy, and applicability of the proposed method in each part.

The main contributions of this article are as follows.

- 1) An efficient rail surface defect detection method is proposed to automate rail extraction, defect segmentation, and defect pinpointing.
- 2) An accurate rail extraction algorithm is proposed for complex and diverse noise.
- 3) Integrating DBC approach with GrabCut and optimizing the segmentation energy function by introducing an entropy penalty factor to automate the segmentation of rail surface defects.
- 4) Precision localization of track surface defects in collaboration with DBC-GrabCut and YOLO v2.

II. METHODOLOGY

The process of the proposed method is shown in Fig. 1. Following the acquisition of images of the rail surface, precise rail extraction, automatic segmentation, and neural network localization operations are performed sequentially to accurately segment and locate rail surface defects. The details are described below.

A. Data Acquisition

The approach for acquiring images of the rail surface is shown in Fig. 2. To enhance the contrast between the rail surface and the background, light sources are added around the camera. The different service time of the rails can result in different types of noise in the corresponding image samples. In this article, according to the type of noise contained, the image samples are classified into five different datasets,

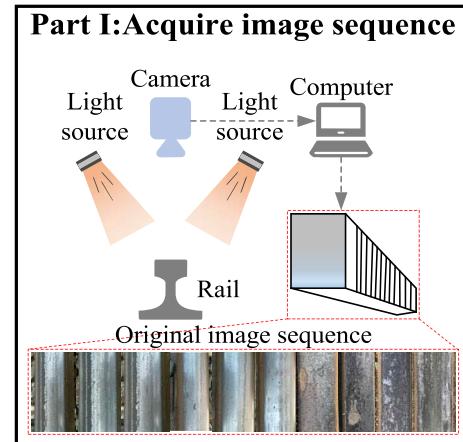


Fig. 2. Diagram of data acquisition method.

TABLE I
RAIL SURFACE IMAGE DATASET

Datasets	Scene	Characteristic
C1	Clean rail surface	Smooth, uneven reflected light
C2	C1 + Slight external noise	Smooth, noisy
C3	Rust, dirt, and complicated rail surface	Rough
C4	C3 + Gaussian noise	Rough, noisy
C5	C3 + Salt pepper noise	Rough, noisy

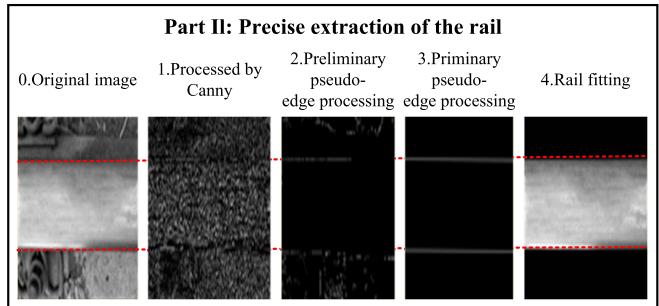


Fig. 3. Process of the proposed method.

as shown in Table I. Each consists of 800 samples, totaling 4000 rail images.

B. Precise Rail Extraction

Precise extraction of rail is a prerequisite for accurate detection of surface defects. In this article, rail extraction is presented in the following four steps: Canny processing, preliminary edge processing, secondary pseudo-edge processing, and rail fitting, as shown in Fig. 3.

The target image is first primed using Canny, in which the pseudo-edge priming is performed according to Algorithm 1. The greyscale values of all pixels in row i of the image to be detected are summed to get S_i and compared with the threshold S_T : if $S_i > S_T$, the row is kept as an edge, and if $S_i \leq S_T$, the current step is skipped and the process of summing and comparison of pixels in row $i + 1$ is carried out. The loop of comparison runs until it finds a pixel row that satisfies the condition, and only the region that satisfies the threshold is retained in the processed image. The key of this algorithm lies

Algorithm 1: Preliminary Processing of Pseudo-Edges

Data: I_s
Input: The original rail image \mathcal{A}
Output: Images that contain only the edges that satisfy the condition \mathcal{A}'

- 1 Graying out \mathcal{A} gives \mathcal{G} ;
- 2 $I_{\max}x \leftarrow$ maximum gray value of all pixels in \mathcal{G} ;
- 3 $(m, n) \leftarrow$ numbers of columns and rows of \mathcal{G} ;
- 4 Calculate S_T with Equation (1);
- 5 **for each** $i \in [1, 2, \dots, n]$ **do**
- 6 $S_i \leftarrow$ the sum of the grayscale values of all pixels in the i^{th} row of \mathcal{G} ;
- 7 **if** $S_i > S_T$ **then**
- 8 Kept the i^{th} row of \mathcal{G} as an edge;
- 9 **else**
- 10 Set the grayscale value of all pixels in the i^{th} row of \mathcal{G} to 0;
- 11 **end**
- 12 **end**
- 13 $\mathcal{A}' = \mathcal{G}$.

in the selection of S_T , which will directly affect the efficiency and accuracy [19].

The favorable S_T for extraction can be described as

$$S_T = \frac{n \times I_{\max}}{I_s} \quad (1)$$

where n is the total number of image columns, I_{\max} is the maximum gray value of pixels, and I_s is the reduction multiplier, which is the first dynamic factor of rail extraction.

After the preliminary pseudo-edge processing, a secondary pseudo-edge process is performed, where each edge is judged to be a pseudo-edge based on the dispersion of its pixel value. Considering that the completeness and accuracy of the edges should be ensured, the standard deviation of the pixels in three adjacent rows is applied as the criterion in this article

$$\left(c - \frac{nc}{m}\right) < \text{STD}_i < \left(c + \frac{nc}{m}\right) \quad (2)$$

where STD_i is the standard deviation of pixel values in rows $i-1, i$ and $i+1$ ($i = (3, 4, \dots, m)$) of the grayed image, n and m represent the number of rows and columns of the image, and c is an adjustable threshold, which is the second dynamic factor of rail extraction.

The edge points retained after the pseudo-edge clean-up are linearly fitted, and the actual edges can be retrieved once the positions of the fitted lines have been determined.

C. Automatic Segmentation

GrabCut [20] integrates texture and boundary information, which performs the best among all the commonly used GraphCut-based interactive image segmentation algorithms. Although the interaction of GrabCut has been comparatively simple, it still cannot achieve automatic segmentation. Therefore, this article applies fractal theory to GrabCut to perform automatic segmentation of targets in complex natural contexts. First, a Gaussian mixture model (GMM) of the track surface

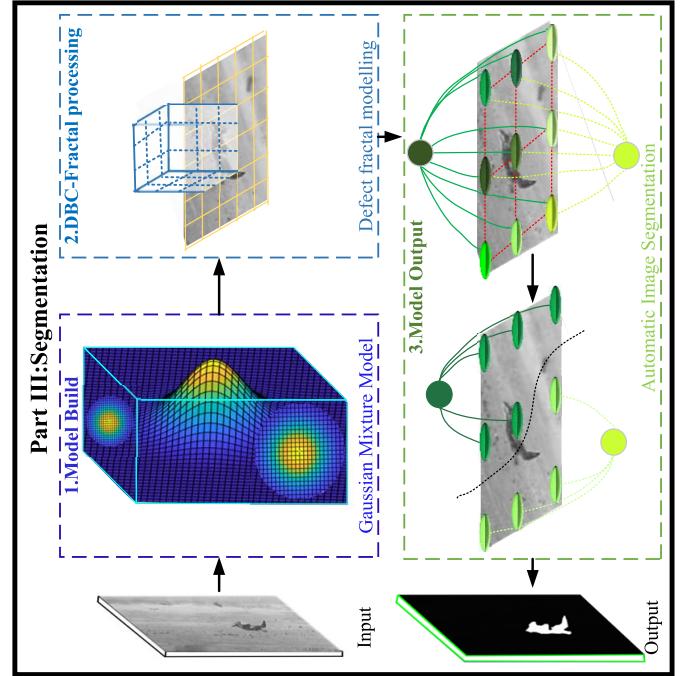


Fig. 4. Process of defect segmentation.

image is established, based on which the prior targets of foreground and background are determined with DBC [21], followed by smoothing and denoising, and finally an entropy penalty factor is introduced to optimize the segmentation energy function to segment defects, as shown in Fig. 4.

To characterize the complexity of an image, the fractal theory is introduced, where the fractal dimension represents the smoothing characteristics of the image, with larger dimensions corresponding to a rougher image.

Let F be any nonempty bounded subset in R^n and mark $N(F, \delta)$ to denote the smallest number in the set whose maximum diameter is δ and can cover F . If the upper and lower dimensions are equal, the box dimension of F can be defined as

$$\dim_B F = \lim_{\delta \rightarrow 0} \frac{\ln N(F, \delta)}{\ln(1/\delta)}. \quad (3)$$

The determination of the box dimension can be based on the following two theorems [25].

Theorem 1: Let $A \in H(x)$, where (X, d) is the metric space, $\varepsilon_n = cr^n$, where $0 < r < 1$, $c > 0$, and $n = 1, 2, 3, \dots$. If $D = \lim_{\delta \rightarrow 0} (\ln N(A, \varepsilon_n)/\ln(1/\varepsilon_n))$, then A has subdimension D .

When $X = R^m$, $r = 1/2$, and $c = 1$, Theorem 1 is the well-known box-counting theorem.

Theorem 2: Let $A \in H(R^m)$ and use the Euclid metric. R^m is covered by square boxes with edge length $1/2^n$, denoting $N_n(A)$ as the number of square boxes intersecting the attractor with edge length $1/2^n$. If $D = \lim_{\delta \rightarrow 0} (\ln(N_n(A))/\ln 2^n)$, then A has subdimension D .

The DBC scales down a grayscale image of $M \times M$ pixels and a grayscale level of G to an image consisting of $s \times s$ subblocks, corresponding to h gray-scale levels, with $G/h = M/s$, s being an integer and $1 < s \leq M/2$, where the scale

factor $r = s/M$. The minimum and maximum gray levels of the sub-block at (i, j) fall within the boxes labeled k and l , respectively, then

$$n_r(i, j) = l - k + 1. \quad (4)$$

The total number of boxes for all sub-blocks is

$$N_r = \sum_{i,j} n_r(i, j). \quad (5)$$

From (4) and (5), the desired r and N_r are obtained by adjusting s . Furthermore, the fractal dimension is obtained by least-squares fitting

$$D = \frac{\log N_r}{\log(1/r)}. \quad (6)$$

The optimization of energy function is performed accordingly. The input image $x = (x_1, \dots, x_n, \dots, x_N)$, x_i is of dimension d , and the GMM is introduced to describe the distribution of foreground and background pixels. Each pixel is represented by K mixed features $m = (k_1, \dots, k_n, \dots, k_N)$ of GMM, $k_n \in 1, 2, \dots, K$, α_n denotes the transparency of the pixel, $\alpha_n = 0$ indicates the background, and $\alpha_n = 1$ refers to the foreground

$$p_k(x_i, \theta) = \frac{\exp\left[-\frac{1}{2}(x_i - u_k)^T \Sigma_k^{-1}(x_i - u_k)\right]}{(2\pi)^{\frac{d}{2}} |\Sigma_k|^{\frac{1}{2}}}. \quad (7)$$

Equation (7) denotes that x_i is a function following the k^{th} class of Gaussian distribution

$$p(x_i/\alpha_n, \theta) = \sum_{k=1}^K \pi_k p_k(x_i, \theta) \quad (8)$$

where $\theta = \{\pi_k, u_k, \Sigma_k\}$ is the model parameter, π_k is the mixing scale factor, u_k is the mean of each Gaussian probability distribution, and Σ_k is the covariance. The maximum mixed negative log-likelihood estimate can be yielded as

$$L(x) = -\ln \prod_{i=1}^N \sum_{k=1}^K \pi_k p_k(x_i, \theta) = -\sum_{i=1}^N \ln \left[\sum_{k=1}^K \pi_k p_k(x_i, \theta) \right]. \quad (9)$$

The Gibbs energy function in GrabCut is given as follows:

$$\begin{aligned} E(\alpha, m, \theta, x) &= U(\alpha, m, \theta, x) + V(\alpha, x) \\ &= \sum_n D(\alpha_n, k_n, \theta, x_n) + V(\alpha, x) \end{aligned} \quad (10)$$

where U stands for the data term and V represents the smoothing term.

Equation (11) is obtained from (9)

$$\begin{aligned} D(\alpha_n, k_n, \theta, x_n) &= -\ln[\pi_k p_k(x_n, \theta)] \\ &= -\ln \pi_k - \ln p_k(x_n, \theta) \\ &= -\ln \pi_k + \frac{1}{2} \ln \det \sum (\alpha_n, k_n) \\ &\quad + \frac{1}{2} [x_n - u(\alpha_n, k_n)]^T \sum (\alpha_n, k_n)^{-1} \\ &\quad \times [x_n - u(\alpha_n, k_n)]. \end{aligned} \quad (11)$$

Finally, the GMM parameters are updated to obtain as follows:

$$\theta = \arg \min U(\alpha, m, \theta, x) \quad (12)$$

$$V(\alpha, x) = \gamma \sum_{(m,n) \in c} [\alpha_m \neq \alpha_n] \exp(-\beta \|x_m - x_n\|^2) \quad (13)$$

where $\gamma = 50$, c is the set of adjacent pixel pairs, $[\cdot]$ is a function that takes the value 0 or 1, $[\alpha_m \neq \alpha_n] = 1$ when and only when $\alpha_m \neq \alpha_n$, $[\alpha_m \neq \alpha_n] = 0$ when and only when $\alpha_m = \alpha_n$, $\beta = (2(\|x_m - x_n\|^2))^{-1}$, and $\langle \cdot \rangle$ denotes the mathematical expectation of the sample.

The higher the entropy, the richer the information contained in the image. Leveraging the properties of entropy, this article employs the mixed negentropic Gibbs energy as the penalty function. According to informatics, the entropy is defined as $H_\pi = -\sum_k^K \pi_k \ln \pi_k$, which reveals the amount of information contained in π_k .

The optimized energy function is shown in the following:

$$\begin{aligned} E(\alpha, m, \theta, x, H) &= U(\alpha, m, \theta, x) + V(\alpha, x) + \rho[-H(\pi)] \\ &= \sum_n D(\alpha_n, k_n, \theta, x_n) + V(\alpha, x) \\ &\quad + \rho \sum_{k=1}^K \pi_k \ln \pi_k \end{aligned} \quad (14)$$

where ρ is the penalty factor and is taken to be five in this article.

The optimized segmentation energy function not only eliminates the redundant Gaussian components, but also preserves the detailed information of the image as much as possible which improves the segmentation accuracy. The Gibbs energy function converges through the mentioned iterative optimization, and the high-precision objective is generally obtained by two to three iterations. As a result, automatic target segmentation in complex contexts can be accomplished by following the above process.

D. Positioning of Rail Surface Defects

YOLO v2 is an excellent tool for locating surface defects in rail images, as shown in Fig. 5. The loss function of YOLO v2 is shown as follows:

$$\begin{aligned} \text{loss}_t &= \sum_{i=0}^W \sum_{j=0}^H \sum_{k=0}^A \mathbb{1}_{\text{MaxIOU} < \text{Tresh}} \lambda_{\text{noobj}} \times (-b_{ijk}^o)^2 \\ &\quad + 1_{t < 12800} \lambda_{\text{prior}} \times \sum_{r \in (x, y, w, h)} (\text{prior}_k^r - b_{ijk}^r)^2 \\ &\quad + 1_k^{\text{truth}} \left[\lambda_{\text{coord}} \times \sum_{r \in (x, y, w, h)} (\text{truth}^r - b_{ijk}^r)^2 + \lambda_{\text{obj}} \right. \\ &\quad \times (\text{IOU}_{\text{truth}}^k - b_{ijk}^o)^2 + \lambda_{\text{class}} \\ &\quad \left. \times \sum_{c=1}^C (\text{truth}^c - b_{ijk}^c)^2 \right] \end{aligned} \quad (15)$$

where W and H represent the width and height of the feature map, respectively, A denotes the number of prior frames, and each λ represents the weight coefficient of each loss. The first

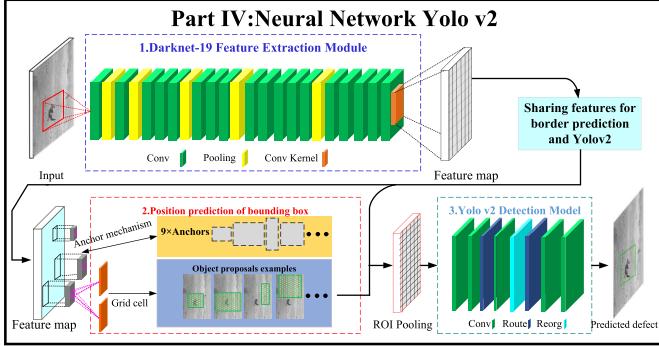


Fig. 5. Model for localization.

TABLE II
RAIL EXTRACTION PERFORMANCE AT DIFFERENT I_s

I_s	Proportion of total samples that can be successfully extracted (%)				
	C1	C2	C3	C4	C5
3	28.25	21.125	12.25	13.45	10.125
4	93.125	83.25	74	63.625	66.25
5	100	99.875	99.5	99.25	99.375
6	88.25	84.375	81	71.5	71.75
7	39.625	26.45	34.125	30.245	24.25

TABLE III
RAIL EXTRACTION PERFORMANCE AT DIFFERENT c

c	Proportion of total samples that can be successfully extracted (%)				
	C1	C2	C3	C4	C5
10	6.325	11.5	12.875	8.375	6.875
15	10.25	32.5	31.25	22.125	18.875
20	99.875	99.625	98.5	96.375	96
25	90.125	89.5	80.875	66.25	68.75
30	73.25	72.125	50.125	45.5	48.875

part calculates the confidence error of background. The second part evaluates the coordinate error between the prior frame and the prediction width. The third part computes the loss of the prediction frame matched with each groundtruth.

III. EXPERIMENT AND ANALYSIS

Four sets of experiments are designed in this article, first to validate the presented rail extraction algorithm and to investigate the effect of dynamic factors I_s and c on the result of rail and background partitioning, then to demonstrate the capability of DBC-GrabCut and the performance of automatic defect segmentation with different noise, next to determine the parameters of the optimal model for defect localization neural network, and finally to analyze the robustness and adaptability of the proposed method for surface defects and to evaluate it in comparison with other existing methods.

A. Precise Positioning of the Rails

1) *Dynamic Factor I_s :* According to the description in Table I, multiple experiments were conducted on each dataset with I_s as 3–7, respectively, and the results are listed in Table II, and a comparison of the positioning effects is shown in Appendix A. The results showed that the highest proportion of total samples that can be successfully extracted was experienced for each sample when $I_s = 5$, thereby setting $I_s = 5$ in the subsequent experiments.

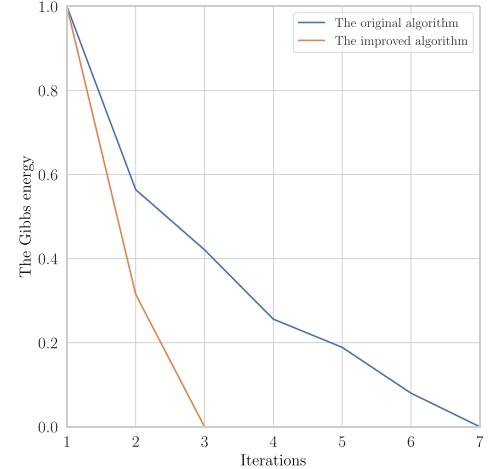


Fig. 6. Comparison of Gibbs energy convergence speed of different algorithms.

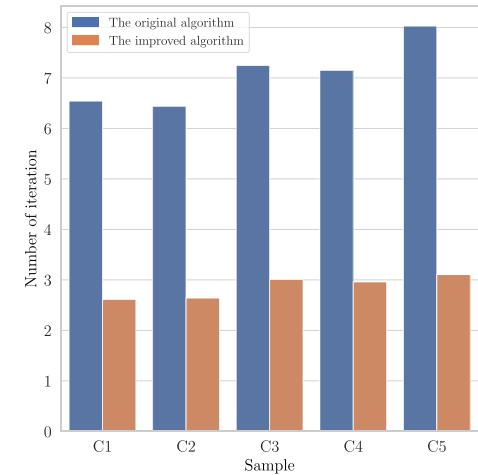


Fig. 7. Comparison of the iterations of different algorithms in processing different datasets.

2) *Dynamic Threshold c :* The results of the five datasets processed by the preliminary pseudo-edge removal algorithm @ $I_s = 5$ were fed into the secondary pseudo-edge processing under different c . c was taken as 10, 15, 20, 25, and 30, and the results are listed in Table III, and a comparison of the extraction effects is shown in Appendix B. The results show that the highest proportion of total samples that can be successfully extracted for each sample is achieved when $c = 20$, which means $I_s = 5$ and $c = 20$ are the optimal combination for rail accurate extraction.

B. Segmentation of Surface Defects

1) *Convergence of the DBC-GrabCut:* The normalized energy resulting from the segmentation of rail images using GrabCut and t DBC-GrabCut is shown in Fig. 6. It can be seen that the convergence direction of the Gibbs energy function of DBC-GrabCut is the same as that of GrabCut. The optimized energy function decreases faster and converges in three iterations, while the original algorithm has a slow energy change and requires more than seven iterations to converge. That is, DBC-GrabCut delivers better segmentation results with fewer iterations.

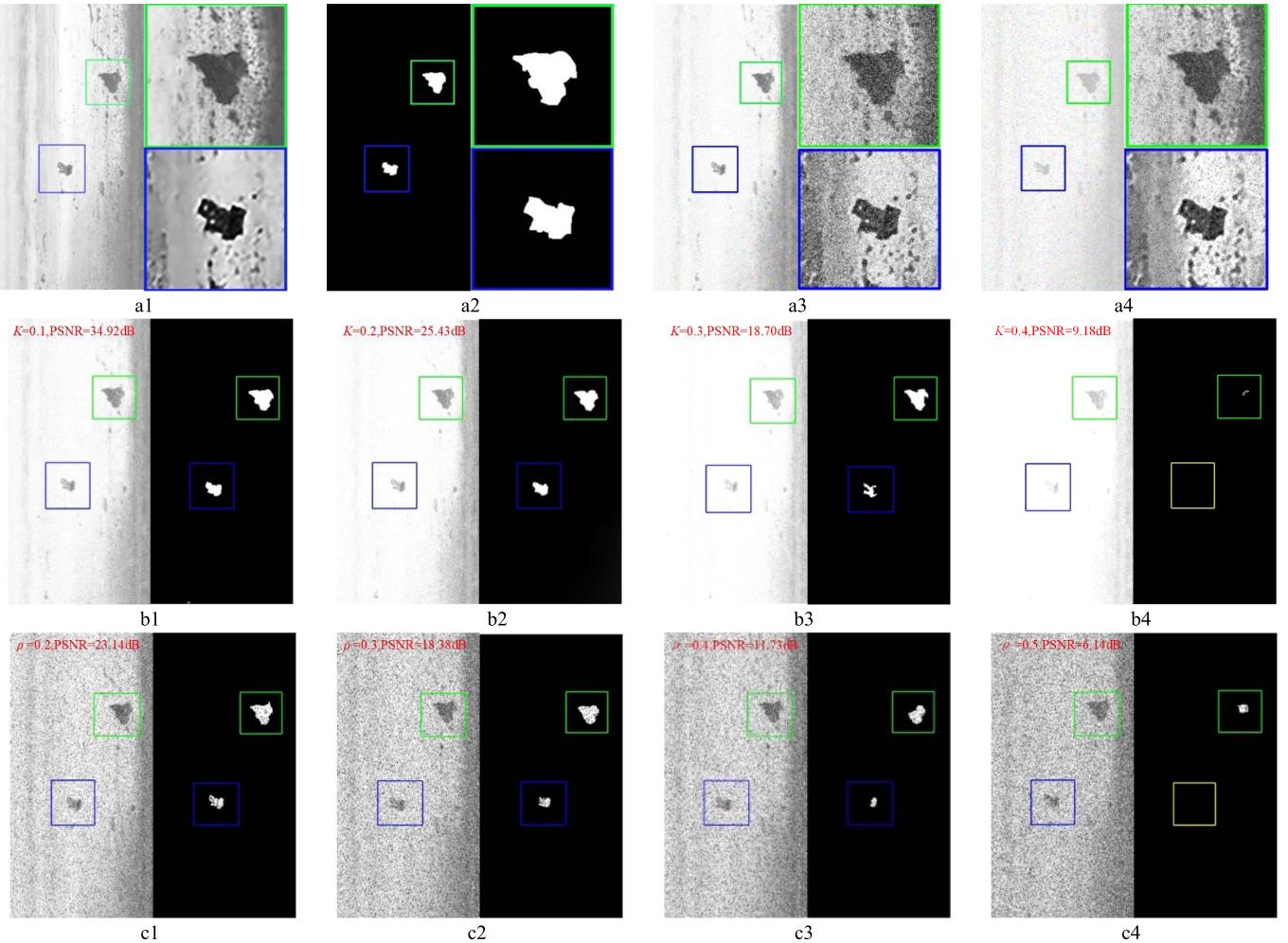


Fig. 8. Performance of the algorithm under different types of noise and different intensities. (a1) Original, (a2) Ground truth, (a3) Corrupted with Gaussian noise, and (a4) Corrupted with pepper noise. (b1)–(b4) Process results with different strength level of Gaussian noise. Correspondingly, the labels placed on the top left, K and PSNR represent the variance and signal to noise ratio of the added Gaussian noise. (c1)–(c4) Process results with different density of salt and pepper noise. Correspondingly, the labels placed on the top left, ρ and PSNR represent the density and signal to noise ratio of the added salt and pepper noise.

Moreover, five datasets are experimented for segmentation, and the average number of iterations is shown in Fig. 7. In C1 and C2, GrabCut iterates more than six times on average, while DBC-GrabCut comes in under three iterations on average; in C3-C5, GrabCut iterates more than seven times on average, while DBC-GrabCut comes in around three iterations on average. It is clear that the performance of the DBC-GrabCut is promising and consistent under different conditions.

2) Serviceability of the DBC-GrabCut: To evaluate the performance of the improved algorithm, different types and intensities of noise are added to the original rail samples in C1–C3 and then performed segmentation. Fig. 8(a) shows the samples of original, groundtruth, Gaussian noise, and pretzel noise from left to right.

Fig. 8(b) shows the results for samples contaminated by Gaussian noise with mean of 0, varying variance k , and signal to noise ratio (PSNR). The results with mild noise, as shown in Fig. 8(b1) and (b2), and the defects segmented with DBC-GrabCut are clearly bounded. As the noise intensity increases, the morphology and size of the defects change, as shown in Fig. 8(b3), where the probability of defects boundary blurring is even as high as 83.75% for C3, for

example, but the DBC-GrabCut works remarkably well and no missed segmentation occurred. The percentage of missed segmentation of the DBC-GrabCut is up to 14.5% for samples with extremely strong Gaussian noise, as shown in Fig. 8(b4).

Pepper noise with different densities ρ and PNSR is applied to the images and processed with the DBC-GrabCut, as shown in Fig. 8(c). The defect is clearly segmented and the boundary details are well-defined for the samples with mild noise, as shown in Fig. 8(c1). The result with medium intensity noise is shown in Fig. 8(c2), with a limited amount of noise near the defects; nevertheless, the defects are still properly segmented. A lot of the edge details are lost due to strong noise, as shown in Fig. 8(c3), which affected the segmentation effect of DBC-GrabCut. The result with extremely strong noise is shown in Fig. 8(c4), in which case the algorithm only segmented large defect with significant changes in morphology and size with blurring inside. In addition, the small defect was missed and the highest percentage of missed segmentation is 9.75%.

The results above show that the DBC-GrabCut features high noise immunity and only fails in the case of extremely strong noise.

TABLE IV
PERFORMANCE OF DIFFERENT MODELS WHEN DEALING WITH DIFFERENT SAMPLES

Samples	Algorithm	MCR (%)	RI (%)	PSNR (db)	MSE	VI (pixel)	Time (s)	GCE (%)
C1	K-means	68.35	23.71	4.21	21000	3.65	0.36	4.05
	GMM	58.67	28.34	10.32	5270	4.98	0.28	8.65
	FCM	28.76	58.24	17.65	2137	4.14	4.26	3.75
	GraphCut	22.54	78.58	22.61	652	1.86	18.34	2.98
	GrabCut	12.54	90.52	25.67	54	2.34	6.47	2.65
	DBC-GrabCut	3.56	97.11	34.61	42	0.48	0.31	2.05
C2	K-means	67.61	21.14	5.32	20600	3.56	0.45	15.64
	GMM	59.43	28.01	8.34	10500	4.93	0.57	15.57
	FCM	42.45	47.47	10.56	1020	3.11	9.84	11.53
	GraphCut	15.64	70.24	25.79	706	1.95	15.37	4.84
	GrabCut	13.88	91.47	24.51	59	3.21	7.32	3.68
	DBC-GrabCut	5.46	93.49	26.34	403	1.05	0.24	2.57
C3	K-means	73.27	26.34	2.37	21600	3.98	0.24	16.32
	GMM	75.31	26.47	5.64	24000	5.54	0.54	18.03
	FCM	67.67	31.51	4.24	22700	4.03	3.35	15.98
	GraphCut	34.35	65.20	3.76	1700	4.04	20.2	6.57
	GrabCut	18.48	70.25	6.54	9753	3.25	10.24	5.65
	DBC-GrabCut	9.35	76.34	8.99	7920	1.31	0.34	5.45
C4	K-means	72.61	25.36	4.42	25200	3.68	0.65	12.57
	GMM	68.67	32.14	4.35	19600	4.97	0.45	12.35
	FCM	40.57	47.54	9.52	5690	3.44	8.65	9.23
	GraphCut	48.34	55.48	10.54	6820	2.75	132.41	8.75
	GrabCut	20.17	68.34	28.34	5570	5.68	12.65	9.87
	DBC-GrabCut	13.42	80.24	11.29	3240	1.07	0.71	8.04
C5	K-means	73.52	27.36	5.34	23500	4.37	0.54	13.14
	GMM	72.45	31.27	5.41	15000	5.04	0.27	13.01
	FCM	60.29	29.45	9.47	12800	4.21	4.35	14.54
	GraphCut	50.47	51.27	9.87	16000	3.87	142.4	10.57
	GrabCut	19.67	76.37	23.72	2100	4.97	7.67	11.57
	DBC-GrabCut	10.57	86.34	13.49	2400	0.97	1.07	9.98

3) *Performance Comparison of Segmentation Algorithms:* We compare the segmentation effects of C1–C5 under the operation of hybrid models, including K-means [22], FCM [23], standard GMM [24], GraphCut [25], and GrabCut [20]. The hardware applied for the experiment is an AMD Ryzen 5 3600 6-Core processor at 3.60 GHz coupled with a GPU with 6 GB of graphical memory. All algorithms are run until convergence.

The statistical results are shown in Table IV, where miss classification ratio (MCR), PSNR, and mean-squared error (MSE) are assessed for images with noise, and rand index (RI), variation of information index (VI), and global consistency error (GCE) are adopted to evaluate segmentation results against groundtruth. The corresponding segmentation results for each algorithm are shown in Appendix C.

As shown in the table, the following factors are used:

- 1) Although the samples in C1 have smooth surface with slight noise, they are not uniform to illumination reflection. K-means, GMM, and FCM have excellent operational efficiencies but high MCR, indicating that these algorithms are sensitive to uneven illumination reflection. The efficiency of GraphCut and GrabCut is not as good as the first three, but the MCR is lower, i.e., it can tolerate the negative effects caused by uneven illumination reflection. In addition, the DBC-GrabCut processes fast with low MCR, indicating its strong comprehensive performance.
- 2) For samples in C2 with mild noise and uneven illumination reflection and rough rail samples in C3 with containing rust and oil, the MCRs of algorithms such as K-means, GMM, and FCM are all high, i.e., they

cannot effectively suppress noise. GraphCut and GrabCut perform a certain suppression on noise, but the computational efficiency is low.

- 3) For samples in C4 with rust, oil, and Gaussian noise, K-means and GMM are quick in processing but showed serious mis-segmentation. The MCRs of GraphCut and GrabCut are not so high, which suggests a certain suppression of Gaussian noise, but the operational processes are slow. For sample in C5 with rust, oil, and pretzel noise, K-means, GMM, and FCM run fast yet with high MCR, indicating that they fail to perfectly suppress the pretzel noise. GraphCut and GrabCut run slowly although they have low MCRs. The proposed algorithm succeeded in eliminating the pretzel noise and segmenting the defects accurately. However, due to the combined effect of oil, rust, and strong noise, the algorithm also suffers from over-segmentation.

In summary, although there is over-segmentation in C4 and C5, indicating that the performance of DBC-GrabCut is not effective enough under the influence of intensive Gaussian and pretzel noise, however, compared with the other algorithms, the algorithm ranked third only in terms of efficiency in C4 and C5, with the rest of the tests in the top two for each metric. Hence, DBC-GrabCut outperforms other models in terms of overall performance, allowing fast, and accurate acquisition of defect edges with high robustness.

C. Neural Network Model Training

From the above analysis, it is clear that DBC-GrabCut does not segment the samples in C4 and C5 satisfactorily. For addressing the issues, a parallel deep multimodel system

TABLE V
PERFORMANCE OF DIFFERENT METHODS UNDER DIFFERENT CIRCUMSTANCES

Samples	Type	Method	Dice(%)	Jacc(%)	FP(%)	Kappa	Pr(%)	Re(%)	Infer(ms)
C1	1	Method in [27]	66.32	58.49	55.21	0.32	76.35	91.65	620
		FCN-32s[29]	61.12	58.63	63.21	0.12	40.21	96.23	110
	2	PSPNet [30]	73.24	71.64	62.31	0.47	71.36	94.32	90
		VggNet[28]	82.14	86.35	9.35	0.84	89.36	90.21	965
	3	Grabcut-Yolov3	80.14	78.12	12.47	0.72	92.36	89.01	600
		Grabcut-Faster R-CNN	95.36	92.21	5.86	0.91	94.36	94.54	450
		DBC-GrabCut-YOLOv2	96.25	90.32	7.24	0.88	96.14	96.35	500
C2	1	Method in [27]	56.24	54.20	40.21	0.21	94.45	86.36	570
		FCN-32s	48.26	46.33	62.45	0.12	34.54	88.36	110
	2	PSPNet	65.70	62.38	50.65	0.04	74.23	92.54	120
		VggNet	72.65	88.32	42.02	0.88	88.66	91.21	968
	3	Grabcut-Yolov3	78.62	85.39	36.88	0.75	95.95	85.36	690
		Grabcut-Faster R-CNN	88.36	86.21	12.35	0.92	95.16	95.63	446
		DBC-GrabCut-YOLOv2	90.24	78.71	10.06	0.93	95.25	96.11	480
C3	1	Method in [27]	65.32	52.45	53.23	0.04	49.79	86.34	590
		FCN-32s	56.20	45.54	40.35	-0.05	39.75	91.32	112
	2	PSPNet	70.54	65.32	30.54	-0.36	53.62	89.21	106
		VggNet	82.36	75.36	12.64	0.78	63.32	95.32	978
	3	Grabcut-Yolov3	84.54	76.36	65.34	0.22	94.70	73.80	703
		Grabcut-Faster R-CNN	90.24	88.69	13.86	0.82	60.97	88.18	462
		DBC-GrabCut-YOLOv2	92.36	90.23	10.35	0.88	97.25	96.20	513
C4	1	Method in [27]	52.31	46.35	76.45	-0.08	42.03	92.31	680
		FCN-32s	48.24	33.35	72.58	-0.34	34.38	90.21	124
	2	PSPNet	56.32	52.8	68.87	-0.11	47.40	89.32	131
		VggNet	66.58	62.35	20.65	0.79	56.68	78.14	908
	3	Grabcut-Yolov3	68.32	88.21	68.42	0.32	75.42	98.10	714
		Grabcut-Faster R-CNN	86.64	73.24	25.36	0.63	46.30	90.24	480
		DBC-GrabCut-YOLOv2	88.17	86.56	12.56	0.78	96.92	93.64	472
C5	1	Method in [27]	46.22	42.47	65.28	0.09	38.03	98.20	790
		FCN-32s	42.34	58.14	54.62	-0.12	47.60	96.50	104
	2	PSPNet	60.23	54.78	50.26	0.19	55.20	92.40	164
		VggNet	68.20	48.65	30.24	0.65	60.41	97.92	996
	3	Grabcut-Yolov3	65.24	79.32	18.54	0.42	86.68	89.68	732
		Grabcut-Faster R-CNN	87.64	69.82	3.20	0.62	91.65	56.21	462
		DBC-GrabCut-YOLOv2	90.24	81.20	8.62	0.85	99.96	98.24	450

is proposed, where the segmentation samples generated by DBC-GrabCut are fed into the neural network YOLOv2 for defect localization. A discussion of iteration and learning rates will be presented in order to yield favorable accuracy and recall. The hardware applied for the experiment is an AMD Ryzen 5 3600 6-Core processor at 3.60 GHz coupled with a GPU with 6 GB of graphical memory.

1) *Iterations*: Ten different iterations are designed for model training with a learning rate of 10^{-2} , and the rail data are divided into 144 batches. Notably, 70% of these samples are used as the training set and the other 30% as the test set, and the results are shown in Fig. 9. For a better understanding of the results, both line plot and box plot, which can intuitively present the distribution status of data, are applied for the description.

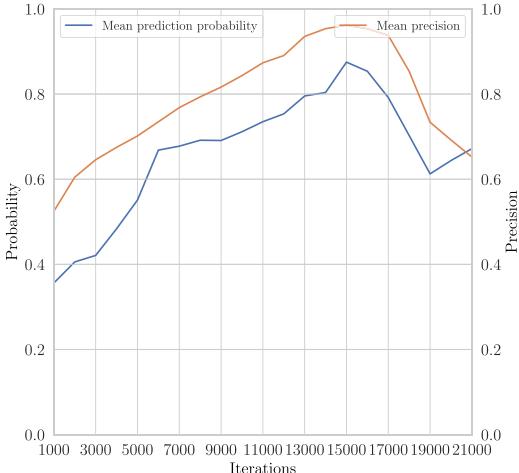
Fig. 9(a) shows that the mean prediction probability (MPP) of the model is only 0.4208 and the mean precision (MP) is only 0.5248 when the number of iterations is 3000. As the number of iterations increases, both MPP and MP increase, reaching a maximum at the same time at 15 000 iterations, 0.8750 and 0.9617, respectively. When the number of iterations continues to increase, these two indicators no longer increase, and when the number of iterations reaches 21 000, the MPP drops to 0.6716 and the MP drops to 0.6521.

Fig. 9(b) shows the box plot of the prediction probability at different iterations. The median is small and the rectangle is long with no outliers when the number of iterations is

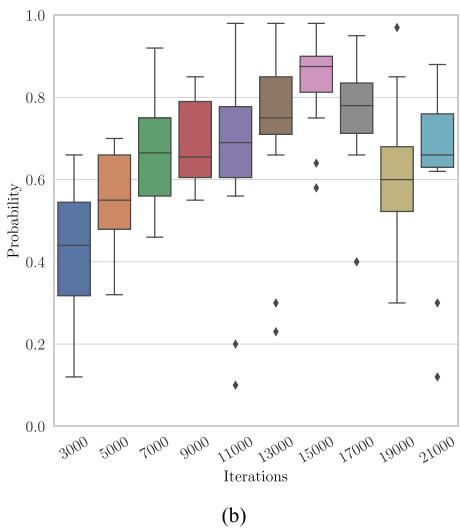
in the range of 3000–9000, indicating that the prediction accuracy of the model is low but reasonably stable. When the number of iterations ranges from 11 000 to 13 000, the median is large yet the rectangle is still long, and the outliers can be found below 0.3, indicating that the model is unstable. At 15 000 iterations, the median is the highest and the rectangle is the shortest. Although there are some outliers, the smallest value of outlier is greater than 0.50, indicating that the model prediction probability is high in most cases now. The indicators deteriorate at iterations 19 000 and 21 000, illustrating that the increase in number will degrade the model performance. The above analysis fully illustrates that the number of iterations of 15 000 is a reasonable option.

2) *Learning Rate*: Eight learning rates are designed to train the model and analyze their effects on the performance. The number of iterations in the training process is set to be 15 000, and the batch size is 144, in which 65% of the samples are served as the training set and the other 35% as the test set, and the results are shown in Fig. 10.

Fig. 10(a) shows that when the learning rate is between 10^{-1} and 10^{-3} , the MPP and MP increase gradually as the learning rate decreases, and both reach the maximum when the learning rate is 10^{-3} , which are 0.9059 and 0.9721, respectively. When the learning rate continues to decrease, MPP and MP no longer increase. When the learning rate is between 10^{-1} and 10^{-5} , the MPP exceeds 0.50 and the MP exceeds 0.84, which indicates



(a)



(b)

Fig. 9. Effect of the number of iterations on the performance of the model. (a) Relationship between the number of iterations and the MPP and MP. (b) Box plot of the relationship between the number of iterations and the MPP.

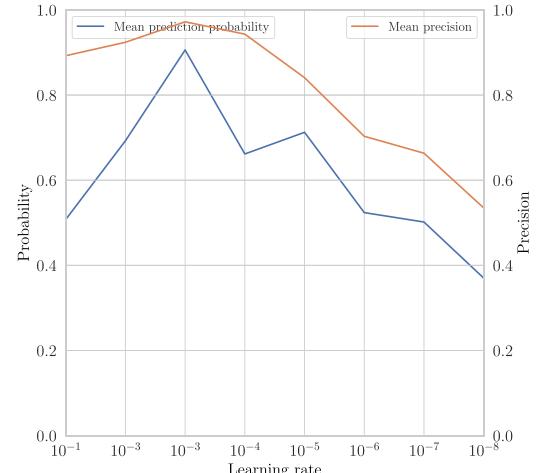
that the model is stable. The model deteriorates gradually in the range of 10^{-6} – 10^{-8} .

The box plots of the prediction probabilities at different learning rates are shown in Fig. 10(b). With learning rates of 10^{-1} and 10^{-2} , both medians are below 0.7 and the boxes are long but without outliers, indicating that the model can predict consistently. At a learning rate of 10^{-3} , the box is shorter and the lower bound is relatively large with no outliers, indicating that the model is performing well in most cases. When the learning rate is 10^{-4} to 10^{-8} , the median decreases, the box length varies, and the minimum values are relatively small, suggesting that the model performance is deteriorating.

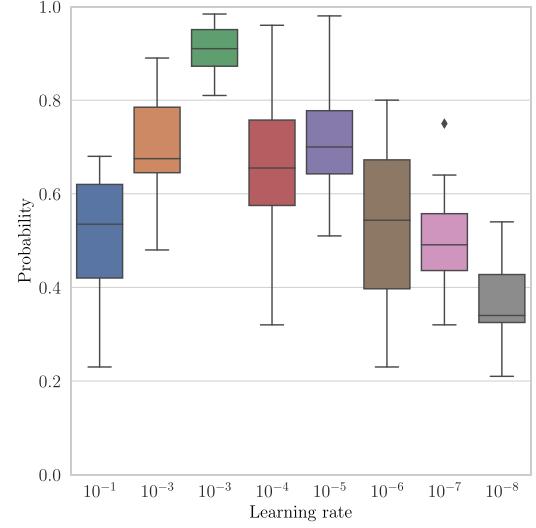
The above analysis shows that a learning rate setting of 10^{-3} is a plausible option.

D. Analysis of the Effect of Locating Surface Defects

1) *Model Performance Comparison*: The three modules of the proposed model, i.e., track pinpointing, defect segmentation, and target localization, are discussed separately



(a)



(b)

Fig. 10. Effect of learning rate on the performance of the model. (a) Relationship between learning rate and the MPP and MP. (b) Box plot of the relationship between learning rate and the MPP.

above. Next, the performance of the whole model is considered.

The semantic segmentation methods for images are broadly classified into three categories, one of which is the traditional image segmentation algorithm, the second is CNN-based method, and the third is a combination of the above two methods [26], [27] describing the application of first category and gives excellent results. VggNet [28], FCN-32 [29], and PSPNet [30] all belong to the second category, and all of them also have promising performance. Methods of the third category are introduced in [31] and [2]. The algorithm proposed in this article also belongs to the third category. This article will illustrate the advantages of the DBC-GrabCut+YOLOv2 method by comparing the performance of the three classes of algorithms for C1–C5 in terms of defect localization. The hardware applied for the experiment is an AMD Ryzen 5 3600 6-Core processor at 3.60 GHz coupled with a GPU with 6 GB of graphical memory. All algorithms are run until convergence.

The results are shown in Table V, where factors of false positive (FP), precision (Pr), and recall (Re) are used to

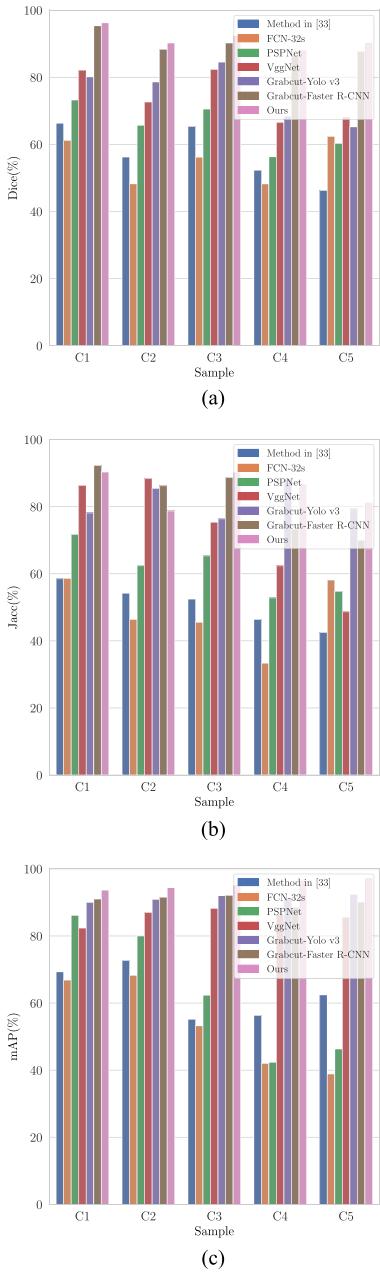


Fig. 11. Accuracy of different methods on various rail samples is (a) pixelwise accuracy, (b) edgewise accuracy, and (c) mean accuracy.

comprehensively evaluate the models' pros and cons, Dice and Jaccard (Jacc) are adopted to assess the accuracy of segmentation edges, and Kappa is the consistency coefficient. The runtime of the model is rounded using Infer. The model performs optimally when FP and Infer are minimized and Dice, Jacc, Kappa, Pr, and Re are maximized which provides excellent suppression of various external noises including rust, oil, and Gaussian and pretzel noise, making the algorithm robust.

As shown in the table, the following factors are used:

- 1) The first category of methods: the segmentation of defects on samples in C1 and C2 is satisfactory, showing that they are tolerant to uneven illumination and mild noise, and are capable of extracting defect edges

TABLE VI
SUMMARY AND COMPARISON OF CURRENT
DETECTION MODEL ABOUT RAIL

Method	Year	Components	Accuracy (%)	Speed (s/frame)
MPCNN in [17]	2012	Rail	92.42	0.00875
Method in [18]	2014	Rail	92.89	N/A
Method in [32]	2015	Concrete ties	93.35	N/A
Method in [31]	2016	Rail	90.77	1.375
MT in [33]	2017	Ties / Fastener	96.74	N/A
ILNET in [34]	2017	Wires	96.54	N/A
DM-RIS[2]	2018	Rail	96.74	0.485
PVANET++in[14]	2018	Split pins	95.28	2.325
Method in [35]	2018	Fastener	92.78	0.00157
Ours	2022	Rail	97.21	0.0064

accurately. However, it is less successful in processing samples in C3–C5, and the FPs are 30% higher in processing C4 samples than in C2, suggesting that it is sensitive to rust, oil, Gaussian and pretzel noise, and cannot segment out the defects accurately. In addition, the efficiency of this type of algorithm is poor.

- 2) The second category of methods: In terms of segmentation edge accuracy (Dice and Jacc), Pr, and FP, VggNet network is the best performer, followed by PSPNet, and FCN-32s delivers poor results. The networks perform fairly good in the C1 and C2 samples, but not so well in the C3–C5 samples. The performance of the networks decreases significantly as the noise is intensified, illustrating that such algorithms are sensitive to rust, oil, Gaussian, and pretzel noise.
- 3) The third category of methods: For samples in C1 and C2, Dice and Jacc of all three models are around 80%, the FP is less than 15%, and Pr and Re are higher than 90%, with the method in this article being the best performer among these indicators. When processing samples in C3–C5, each indicator of GrabCut-YOLO v3 and GrabCut-Faster-R-CNN decreases, demonstrating that they are sensitive and less robust to rust, oil, Gaussian, and pretzel noise; however, the model in this article still performs well. Since rail defects affect passenger safety, miss detection should be avoided. Therefore, the higher the Re, the better if the Pr of the model satisfies the requirements.

In summary, the third category of methods is better than the other two categories of methods overall when it comes to rail surface defects detection. The performance indicators of the proposed algorithms are among the best two for all models except Infer, showing the optimal overall performance.

- 2) Edge Pixel Accuracy and Overlap Estimation: Fig. 11 statistically shows the segmented edge pixel accuracy and overlap for each model. It can be seen that the trend of Dice, Jacc, and mAP is consistent, in which the mAP of the third category of methods is greater than 88%, with Dice and Jacc mostly exceed 70%. The performance of the second category of methods was unstable, with the best-performing VggNet having a mAP above 82% and Dice and Jacc mostly above 60% except a Jacc of only 48.65% in C5. The mAPs of FCN-32s and PSPNet are more than 65% on C1 and C2 samples, but less than 45% on both C4 and

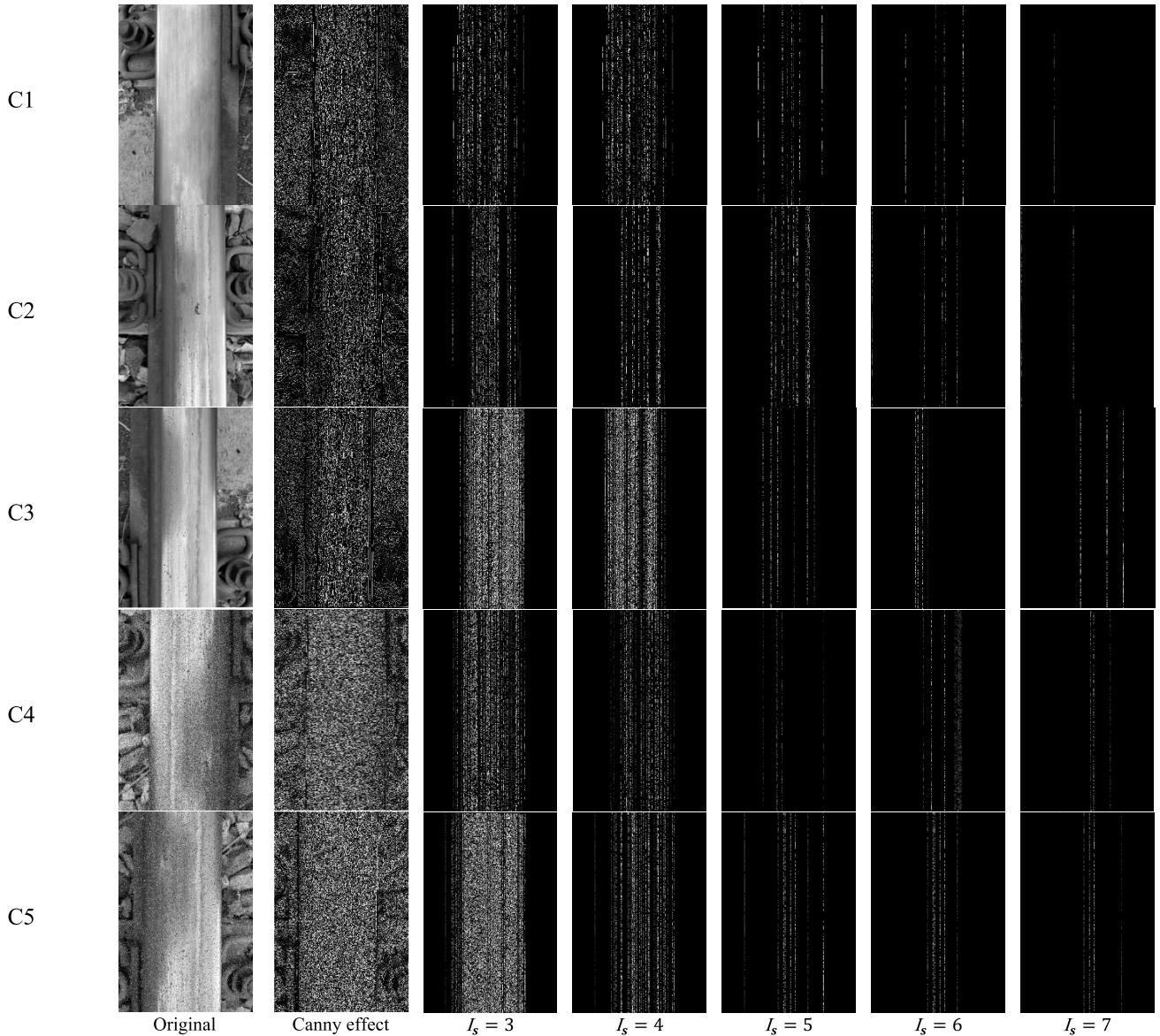


Fig. 12. The result of the edge extraction algorithm varies with the factor I_s . Each row from top to bottom represents the samples in C1 to C5 and their processing results sequentially.

C5. Although the mAP of GrabCut-FasterR-CNN exceeds that of the proposed model in C1, the proposed model is higher than other models in all other samples: in C3–C5 samples, the mAP of the proposed model is greater than 90% and Jacc and Dice are greater than 80%. It can be seen that the algorithm proposed in this article has optimal comprehensive performance.

3) Comparison With Existing Models: The rail detection models proposed in recent papers are listed in Table VI, where “N/A” means not recorded in the article. Since the experiments were conducted under different conditions, with different samples and computational configurations, and each article has a distinct research direction and focus, a direct comparison in terms of “Accuracy” and “Speed” is not meaningful and not the real purpose of the table. However, it can still be concluded from the comparison that the proposed model

(accuracy = 97.21, speed = 0.0064) is capable of performing the rail detection task.

IV. CONCLUSION AND PROSPECTS

In this article, a multilevel, end-to-end accurate and fast method for detecting defects is proposed for complex and diverse rail surfaces. First, the rails in the image are accurately extracted, and after several experiments and comparisons, it is found that the extraction accuracy is the highest when the dynamic factor of the preliminary pseudo-edge algorithm is $I_s = 5$ and the dynamic threshold of the secondary pseudo-edge algorithm is $c = 20$. Second, the DBC algorithm is used to identify the Gaussian model parameters to enable GrabCut to perform automatic segmentation. And the entropy penalty factor is employed to enable the improved algorithm to quickly segment the rail defects. As the result, such algorithm

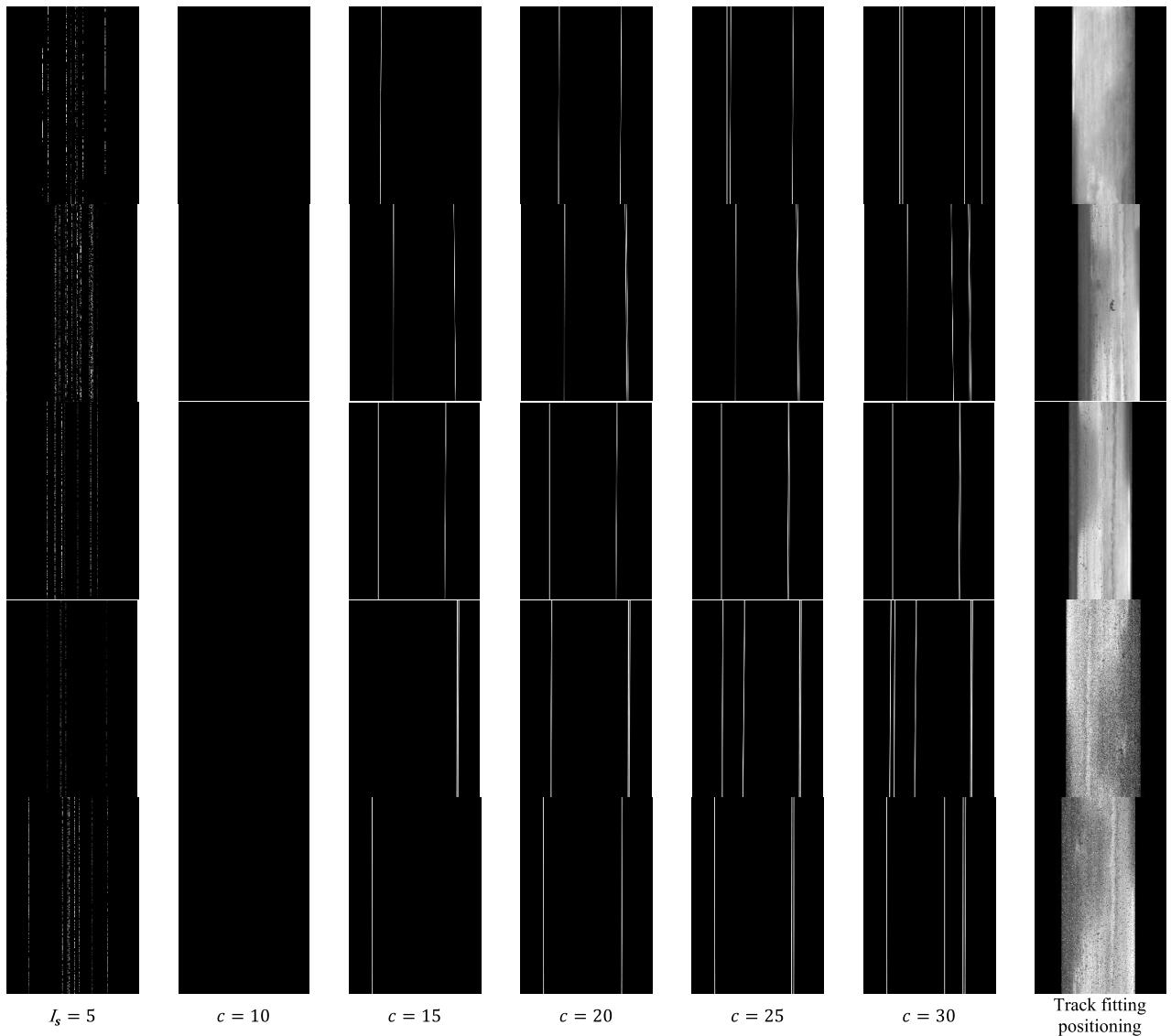


Fig. 13. The results of the edge extraction algorithm varies with the factor c : Each row from top to bottom represents the samples in C1 to C5 and their processing results sequentially.

can accurately segment C3–C5 samples containing intensive Gaussian and pretzel noise. Several sets of experiments verify that the algorithm proposed in this article can realize fast and high-precision localization with an average accuracy of 97.11%, an average recall of 96.10%, and an average frame rate of 0.0064. Compared with similar methods, the method in this article features the robustness and outperforms other models in terms of key indexes such as PSNR and accuracy, and can effectively carry out rail defect detection.

There are still some drawbacks in the article, which will be further studied in the future work.

- 1) Defects are not classified. In fact, some defects are not necessary to be processed, and if they are not distinguished, it will add a lot of ineffective work for maintenance.
- 2) Limited by the chosen edge extraction algorithm and pseudo-edge elimination algorithm, when the two edges of the rail in the image are not parallel to the edge of the image, it may not be able to accurately position

the rail. This situation is difficult to completely avoid because many reasons, such as the installation error or the vibration of camera may cause incorrect view angle for the rails. Under such circumstances, the extraction algorithm applied in this article should be improved to be more adaptive with the ability to correct the skewed edges.

- 3) The types of noise in the rail images not only include Gaussian noise and pretzel noise, but also a small amount of uncommon noise such as Rayleigh noise and gamma noise, and the noise in each image is not exactly of a single type but a mixture of multiple types. In this article, no purposeful treatment of uncommon noise is performed, yet the performance of the algorithm under mixed noise is not examined.

APPENDIX A

See Fig. 12.

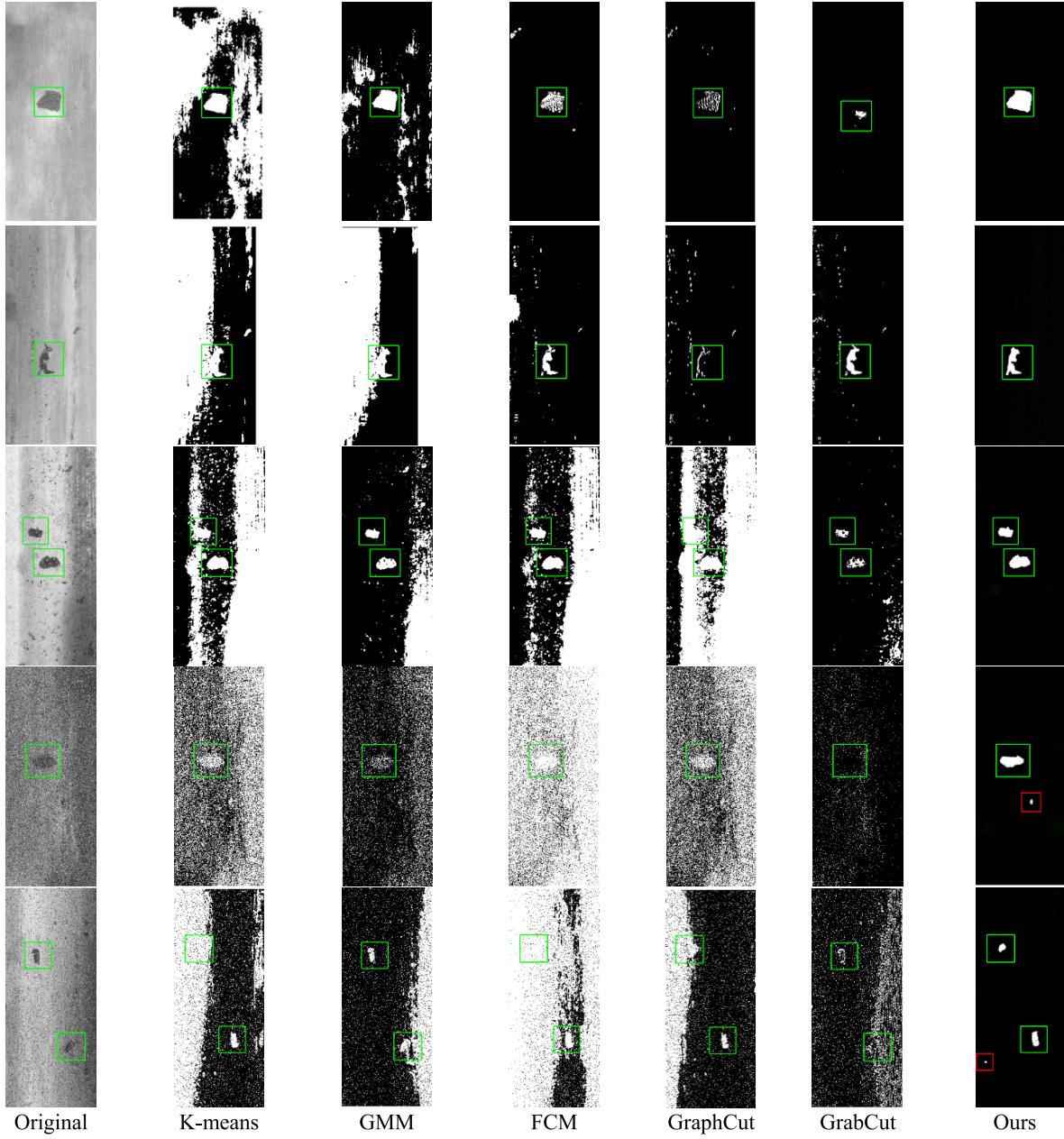


Fig. 14. The results of different mixed models on different samples: Each row from top to bottom represents the samples in C1 to C5 and their processing results sequentially.

APPENDIX B

See Fig. 13.

APPENDIX C

See Fig. 14.

REFERENCES

- [1] A. Jamshidi *et al.*, “A big data analysis approach for rail failure risk assessment,” *Risk Anal.*, vol. 37, no. 8, pp. 1495–1507, 2017.
- [2] X. Jin *et al.*, “DM-RIS: Deep multimodel rail inspection system with improved MRF-GMM and CNN,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1051–1065, Apr. 2020.
- [3] 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.
- [4] O. Orringer, W. R. Paxton, D. E. Gray, and P. K. Raj, “Residual stress and its consequences on both sides of the wheel-rail interface,” *Wear*, vol. 191, nos. 1–2, pp. 25–34, Jan. 1996.
- [5] J. Lundberg, A. Bohlin, and M. Syk, “Capacity test of ultrasonic equipment used for crack detection in railway application,” *Int. J. Syst. Assurance Eng. Manage.*, vol. 2, no. 2, pp. 163–168, Jun. 2011.
- [6] A. K. Soni, B. Sasi, S. Thirunavukkarasu, and B. P. C. Rao, “Development of eddy current probe for detection of deep sub-surface defects,” *IETE Tech. Rev.*, vol. 33, no. 4, pp. 386–395, Jul. 2016.
- [7] C. Lu, Y. Wei, and W. Xu, “Study on B-scan imaging detection for rail tread tilted cracks using ultrasonic surface wave,” *Chin. J. Sci. Instrum.*, vol. 31, no. 10, pp. 2272–2278, 2010.
- [8] A. G. Verkhoglyad *et al.*, “IR-termography methods for the defects detection IA railway wheels,” in *Proc. 7th Int. Conf. Actual Problems Electron. Instrum. Eng.*, Novosibirsk, Russia, 2004, p. 217.
- [9] T. Hackel, D. Stein, I. Maindorfer, M. Lauer, and A. Reiterer, “Track detection in 3D laser scanning data of railway infrastructure,” in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (IMTC)*, May 2015, pp. 693–698.

- [10] C. Sacchi, C. Regazzoni, and G. Vernazza, "A neural network-based image processing system for detection of vandal acts in unmanned railway environments," in *Proc. 11th Int. Conf. Image Anal. Process.*, 2001, pp. 529–534.
- [11] C. Aytekin, Y. Rezaeitabar, S. Dogru, and I. Ulusoy, "Railway fastener inspection by real-time machine vision," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 7, pp. 1101–1107, Jul. 2015.
- [12] Y. He, K. Song, H. Dong, and Y. Yan, "Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network," *Opt. Lasers Eng.*, vol. 122, pp. 294–302, Nov. 2019.
- [13] V. Natarajan, T.-Y. Hung, S. Vaikundam, and L.-T. Chia, "Convolutional networks for voting-based anomaly classification in metal surface inspection," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2017, pp. 986–991.
- [14] 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.
- [15] Y. He, K. Song, and Q. Meng, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1493–1504, Apr. 2020.
- [16] S. Wang, Y. Yin, G. Cao, B. Wei, Y. Zheng, and G. Yang, "Hierarchical retinal blood vessel segmentation based on feature and ensemble learning," *Neurocomputing*, vol. 149, pp. 708–717, Feb. 2015.
- [17] J. Masci *et al.*, "Steel defect classification with max-pooling convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2012, pp. 1–6.
- [18] D. Soukup and R. Huber-Mörk, "Convolutional neural networks for steel surface defect detection from photometric stereo images," in *Proc. Int. Symp. Visual Comput.*, vol. 8887. Cham, Switzerland: Springer, 2014, pp. 668–677.
- [19] K. Wei, T. Zhang, X. Shen, and J. Liu, "An improved threshold selection algorithm based on particle swarm optimization for image segmentation," in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, vol. 5, 2007, pp. 591–594.
- [20] L. Zhou and J. He, "Improved image segmentation algorithm based on GrabCut," *J. Comput. Appl.*, vol. 33, no. 1, pp. 49–52, Sep. 2013.
- [21] M. Shamsi, P. B. Saad, S. B. Ibrahim, and A. R. Kenari, "Fast algorithm for iris localization using Daugman circular integro differential operator," in *Proc. Int. Conf. Soft Comput. Pattern Recognit.*, Dec. 2009, pp. 393–398.
- [22] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 29, no. 3, pp. 433–439, Jun. 1999.
- [23] Z. Ding, Z. Hu, J. Yang, and Z. Tang, "FCM algorithm for the research of intensity image segmentation," *Acta Electronica Sinica*, vol. 25, pp. 39–43, May 1997.
- [24] G. McLachlan and D. Peel, *Finite Mixture Models* (Wiley Series in Probability and Statistics). Hoboken, NJ, USA: Wiley, 2000.
- [25] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *Int. J. Comput. Vis.*, vol. 70, no. 2, 2006, pp. 109–131.
- [26] A. Xu, L. Wang, S. Feng, and Y. Qu, "Threshold-based level set method of image segmentation," in *Proc. 3rd Int. Conf. Intell. Netw. Intell. Syst.*, Nov. 2010, pp. 703–706.
- [27] B. Bhanu, S. Lee, and J. Ming, "Adaptive image segmentation using a genetic algorithm," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 12, pp. 1543–1567, Dec. 1995.
- [28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2014, pp. 1–14.
- [29] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
- [30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2017, pp. 6230–6239.
- [31] S. Faghri-Roohi, S. Hajizadeh, A. Nunez, R. Babuska, and B. De Schutter, "Deep convolutional neural networks for detection of rail surface defects," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 2584–2589.
- [32] X. Giben, V. M. Patel, and R. Chellappa, "Material classification and semantic segmentation of railway track images with deep convolutional neural networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 621–625.
- [33] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multitask learning for railway track inspection," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 153–164, Jan. 2017.
- [34] Z. Liu, L. Wang, C. Li, and Z. Han, "A high-precision loose strands diagnosis approach for isoelectric line in high-speed railway," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1067–1077, Mar. 2018.
- [35] 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.



Hongfei Yang received the master's degree in mechanical engineering from Jilin University, Changchun, China, in 2020, where he is currently pursuing the Ph.D. degree in test and measurement technology and instrumentation.

His main research interests include industrial defect detection, environmental identification of engineering vehicles, and intelligent sensing instrumentation.



Yanzhang Wang received the B.S., M.S., and Ph.D. degrees from Jilin University, Changchun, China, in 2002, 2005, and 2010, respectively.

He is currently a Professor with the School of Instrumentation and Electronic Engineering, Jilin University. His research interests are infrared spectral detection, intelligent sensing, and intelligent signal processing.



Jiyong Hu received the B.Sc. and M.Sc. degrees from Jilin University, Changchun, China, in 2009 and 2012.

He is currently a Planning Engineer with FAW-VW Changchun. His main work focuses on conveyor system of automobile assembly shop.



Jiatang He received the B.Sc. degree in electrical engineering and automation from Wismar University Germany, Wismar, Germany, in 2007.

He is currently a Supervisor Planning Engineer with FAW-VW, Changchun, China. His main work focuses on conveyor system of automobile assembly shop.



Zongwei Yao received the B.Sc. and Ph.D. degrees in engineering from Jilin University, Changchun, China, in 2008 and 2013, respectively.

He is currently an Associate Professor with Jilin University, where he is heading a group working on intelligent of engineering vehicles. He has been focusing on research areas of 3-D construction of environment, autonomous operation of construction vehicles, mechanical system dynamics.



Qiushi Bi received the M.Eng. and Ph.D. degrees in engineering from Jilin University, Changchun, China, in 2014 and 2019, respectively.

He is currently a Postdoctoral Fellow with Jilin University. His research interests are in the areas of unmanned engineering vehicles, finite element analysis, trajectory optimization, and discrete element analysis.