



Context-guided coarse-to-fine detection model for bird nest detection on high-speed railway catenary

Hongwei Zhao^{1,2} · Siquan Wu² · Zhen Tian² · Yidong Li^{1,2} · Yi Jin^{1,2} · Shengchun Wang³

Received: 9 October 2022 / Accepted: 22 May 2023 / Published online: 26 July 2023
© The Author(s) 2023

Abstract

As a critical component of ensuring the safe and stable operation of trains, the detection of bird's nests on the rail catenary has always been essential. Low-resolution images and the lack of labelled data, however, make it difficult to detect smaller bird's nests (those occupying small pixels in the input image). Previous solution relies on manual online patrol or offline video playback, which severely limits the detection efficiency. Previously, this challenge was addressed by manual online patrol or offline video playback, which severely limits detection efficiency. We propose in this work a context-guided coarse-to-fine detection model (CG-CFDM) for solving the bird's nest detection problem. This solution consists of a context reasoning module and a coarse-to-fine detection network. By detecting domains and matching templates, the context reasoning module generates new labelled context bounding boxes, thereby reducing the burden of annotation. As a result of its delicately designed architecture and powerful representation learning ability, this trained coarse-to-fine detection network further facilitates the detection of bird's nests in an efficient and accurate manner. Extensive experiments demonstrate that the proposed approach is superior to existing methods in terms of performance and has a great deal of potential for detecting bird's nests.

Keywords Bird nest detection · Deep learning · Context-guided · Coarse-to-fine · Railway catenary

1 Introduction

The development of railway electrification technology and the rapid expansion of railway mileage have made bird's nest detection on the railway catenary an increasingly important part of railway safety. In high-speed railways, the catenary is a power supply line that transmits electricity from the main railway to the trains. Occasionally, bird activities,

particularly nesting behavior, result in problems such as short circuits of catenary lines, burning, tripping, and damage to electrical components. In spite of this, detecting bird's nests is a challenging task due to the fact that birds prefer to nest on high-speed rail catenary switch bases, steel columns, hard beams, and other hidden areas. Moreover, the image has a low resolution and the nests take up a small portion of the input image, making the detection task more difficult. One of the solutions that can reduce the negative effects caused by the construction of bird's nests on the catenary is to rely on manual inspections such as human on-site inspections or manual online patrols or offline video playbacks. In addition

Communicated by B. Bao.

Siquan Wu, Zhen Tian, Yidong Li, Yi Jin and Shengchun Wang contributed equally to this work.

✉ Yidong Li
ydl@bjtu.edu.cn

✉ Shengchun Wang
wangshengchun@rails.cn

Hongwei Zhao
hwzhao@bjtu.edu.cn

Siquan Wu
18231420@bjtu.edu.cn

Zhen Tian
18301017@bjtu.edu.cn

Yi Jin
yjin@bjtu.edu.cn

¹ Key Laboratory of Big Data and Artificial Intelligence in Transportation, Ministry of Education, Beijing 100044, People's Republic of China

² Beijing Jiaotong University, No.3, Shangyuan Village, Beijing 100044, People's Republic of China

³ Infrastructure Inspection Research Institute, China Academy of Railway Sciences Corporation Limited, 2 Daliushu Road, Haidian District, Beijing 100081, People's Republic of China

to being time-consuming and labor-intensive, this manual method is highly expensive and has a low degree of reliability. Furthermore, the extremely low efficiency makes it difficult to detect potential safety hazards in a timely manner.

In light of this, identifying the bird's nest effectively and rapidly instead of manual operation has been a challenging but rewarding task. In spite of the impressive progress that has been made in the area of object detection in previous studies, the detection of bird's nests remains far from a practical solution. As an example, the Faster-RCNN detection model relies on two-stage detectors, which deliver better performance at the expense of inference time, making it less suitable for real-time applications. The YOLO object detector is one of the most popular single-stage models, but its detection effect is generally lower than that of a two-stage model, particularly when the training data set is small and the object features are not very distinctive. Additionally, each of these methods requires a large amount of annotation data in order to train the model, but the dataset for bird's nest detection is very limited.

As a solution to these challenges, we propose a novel coarse-to-fine detection model (CG-CFDM) for detecting bird's nests. While the low resolution and limited number of pixels make it difficult to detect bird's nests, their presence at railway catenaries can be recognized by considering the context of their location. Hence, we believe that the key to solving this problem lies in how we can incorporate context as additional information to aid in the detection of bird's nests. The first step is to extract the interest domain of bird nest from a small number of annotated images by contextual similarity constraint in order to solve the problem of insufficient training images. We recommend a matching approach for discovering interest domain regions in other images based on proposals generated from selective search methods. Clearly, this is an unsupervised and efficient method of generating a novel bounding box level dataset that is used in real-world scenes for bird's nests. Secondly, we propose a coarse-to-fine detection model for training for bird's nest detection once the novel dataset has been obtained.

In particular, the proposed CG-CFDM is based on a cascaded YOLO network, which combines the advantages of both two-stage and single-stage detection models. For detection of bird's nest, CG-CFDM performs a coarse-to-fine strategy, with coarse-level detection and fine-level detection being calculated simultaneously for different detected images. To speed up the detection process of both detection models, CG-CFDM adopts pipeline technology. Furthermore, we employ multi-threaded parallel computing technology to maximize detection throughput and make full use of computing resources. The parallel pipeline acceleration algorithm that has been optimized for the CG-CFDM network allows for a detection rate that is comparable to that of a single-stage network, which allows for real-time

detection of bird's nests. Furthermore, the proposed CG-CFDM network has the excellent detection rate of the two-stage network, and it has the ability to filter out irrelevant environmental samples effectively.

In summary, the research motivation and corresponding innovative work of this paper are as follows:

- (1) To solve the problem of small object detection in open scenarios, we propose a novel context-guided coarse-to-fine detection model (CG-CFDM) that makes full use of the context information for detecting bird's nests.
- (2) To reduce annotation costs, we propose an efficient strategy for automatically inferring and reasoning context bounding boxes.
- (3) To ensure the real-time performance of the detection model, CG-CFDM uses pipeline technology and multi-threaded parallel computing technology to maximize computing resources.

Experimental results indicate that the recall rate of the CG-CFDM network reaches 93.69%, an improvement of 16.21% and 30.18% when compared with YOLOv3-SPP and YOLOv4 networks, respectively. In addition, we achieve a higher detection frame rate that meets the requirements of rapid and accurate detection of the bird's nest on a high-speed railway.

2 Literature review

The most commonly used methods for reducing the adverse effects of bird's nest construction on the Internet are structural optimization, blocking, and repelling [1]. Furthermore, civilian prevention methods such as visual shock have also been proposed due to their low cost [2]. Regular manual inspections are primarily used to detect foreign objects intruding into high-speed rail catenary in addition to routine operations. Regular patrols are conducted along the railway to detect and remove foreign objects. In light of the fact that the locations of the catenary are relatively scattered, and the locations of foreign objects are not fixed, this manual inspection method is labor-intensive, time-consuming, and expensive.

A video analysis technology has been used recently to detect foreign objects entering the catenary, but the technology still requires a great deal of manual labor and performs poorly in real time. For the purpose of improving the intelligent identification technology for high-speed railway catenary and increasing detection efficiency, the detection of bird's nests on high-speed railways is primarily accomplished by manually judging and marking video transmitted by the inspection vehicle. As a manual method of detection, it is also time consuming and cannot be used to detect

the bird's nest in a timely and accurate manner. Thus, the automatic identification technology for bird's nests on high-speed railways attracted an increasing amount of attention. In China, there have been numerous studies on the detection of contact network status using image processing. In order to solve the problems, the automatic visual detection technology is applied to the video captured along the railway transmitted by the inspection vehicle. In an earlier study [3], it has been suggested to use traditional image processing methods such as local dynamic threshold binarization in order to first obtain a rough image of the suspected bird's nest, and then to extract HOG features to use in a SVM classifier. Using the characteristics of bird nests that are often seen in the cross-arm area of telephone poles, Huang et al. [4] propose establishing a classification model that incorporates color characteristics as well as structural characteristics of catenary towers for detection. By using this method, it is possible to achieve a higher level of detection accuracy and anti-interference. In [5] analyses the bird's nest images on the catenary equipment and concludes that almost all bird's nests are constructed at several fixed locations on the catenary equipment. As a result, the HOG feature is initially extracted from the key area of the suspected bird's nest, and then classified using a SVM classifier, in order to achieve a relatively satisfactory result. Moreover, [6] proposes modeling the bird's nest based on its color, shape, and relative location. When compared to the traditional manual method, the above methods can enable automatic recognition of bird's nests, but each requires the training of a classification model, which is time-consuming. As an alternative to the above recognition method, there have been studies [7] using the tracking method of contact line detection and parameter extraction on a catenary, and the Hough transform method of recording the candidate line parameters is then used to compare the collected infrared video to complete the comparison. In addition, some methods are designed for the detection of objects. Using exhaustive search, for example, we select a window to scan the entire capture scene and then adjust its size to continue scanning [8].

In recent years, deep learning-based object detection has made significant progress. In some cases, the excellent YOLO network has been applied to perform the detection, such as road detection and automatic license plate detection [9–12]. In [13], a spatial pyramid matching method is presented, which uses the position context provided by the detector to separate the object from its background, and then constructs a spatial feature histogram for recognition. The use of deep learning technology for traffic sign recognition has also been proposed [14]. Yao et al. [15] introduce a recognition algorithm based on convolutional neural network to solve the problem of vehicle color recognition. Deep neural networks have also been used to detect road cracks [16]. It is obvious that deep learning technology has also been

extensively used in the inspection of high-speed railways. A representative example is the visual integration method based on deep learning [17, 18] in the detection of catenary secant breaks and the detection of contacts in catenary insulators. By using point cloud recognition technology, Lin et al. [19] achieve a higher level of detection accuracy. In [20] proposes to improve Fast R-CNN and Faster R-CNN. Research has been conducted comparing these two models to the detection of railway catenary birds' nests [21], with satisfactory results. In [22] builds a new approach based on Faster R-CNN. In [23] improves the convolutional neural network VGG16 to extract features of the object. Compared to AlexNet [24] and GoogLeNet [25], the improved VGG16 structure uses a batch normalization layer [26] with the best convergence effect to process the results, and then refers the results to an RPN network with sliding windows to distinguish between the different types of objects. In particular, the initial area is derived from the convolutional feature map, and a deconvolution operation is added as a feature map layer to the Conv4 convolutional feature map for transmission to the contact network for the detection of bird's nests.

In combination with various algorithms in the above research and the corresponding scenarios, we propose a target detection model based on cascaded YOLO neural network to realize the automatic identification of the bird's nest of high-speed railway catenary. Compared with traditional image recognition technology, deep learning has obvious advantages in feature extraction and can enrich the types of image recognition with higher precision and speed.

3 CG-CFDM network model construction

3.1 Overall network framework

This section mainly describes the whole deployment environment of Bird's Nest detection from front-end data collection to back-end algorithm analysis (as shown in Fig. 1). Firstly, the high-definition camera equipment is used to capture the railway environment information to form video data, and the video data is transmitted to the intelligent vision sensor equipment through WIFI. After that, the intelligent visual sensor equipment performs real-time processing of video data uploaded by the camera equipment through a two-level network from coarse to fine by calling the cascade depth neural detection network model, and transmits the processing results to the database of the data center server through the Internet. Finally, users can view the detection results through a web browser.

This smart device is capable of running a variety of operating systems, such as Windows, Linux, and Unix. It is particularly advantageous to use Linux systems, which are

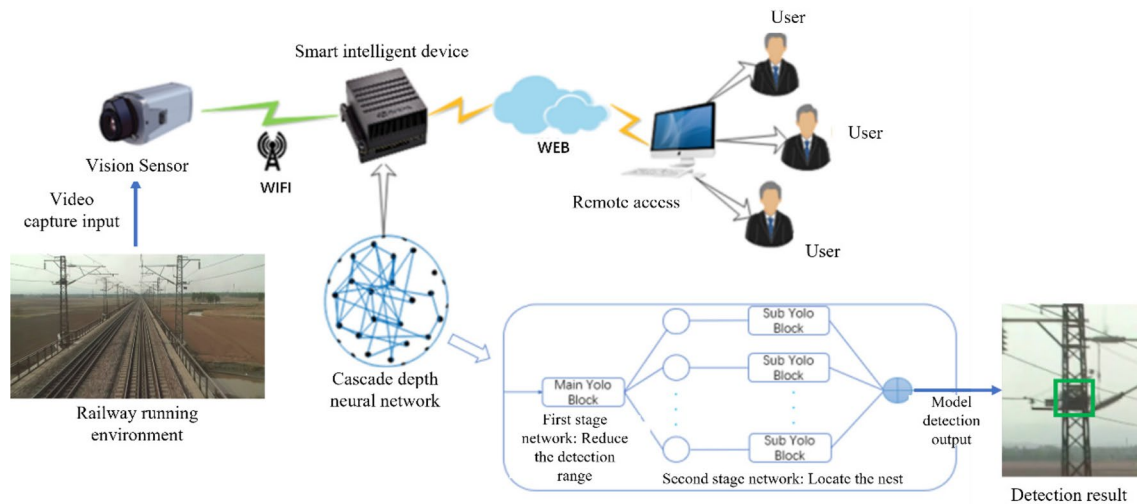


Fig. 1 The deployment of the proposed model

open source, compatible with mainstream deep learning development frameworks, and can easily transfer detection software developed for other platforms to smart devices.

The hardware environment for the CG-CFDM network is the Nvidia series development version, such as the Jetson Xavier NX module. As well as processing data from multiple high-resolution sensors and supporting all popular deep learning frameworks, it offers 21TOPS service-level performance, allows multiple modern neural networks to be run in parallel, and effectively supports the detection pipeline of the CG-CFDM network as well as the parallel detection requirements for the second-level detector. The system is capable of acquiring and processing railway video data in real-time.

3.2 CG-CFDM detection network structure

The main structure of CG-CFDM detection network is shown in Fig. 2. Suppose a piece of railway video data to be detected v can be expressed as $v = \{I_1, I_2, \dots, I_N\}$, where $I_i (i = 1, 2, \dots, N)$ is the video frame, N is the total number of frames of the video. CG-CFDM performs hierarchical detection on each video frame I_i , and the interest domain sub-image of the i -th frame is expressed as $R_i^j (j = 1, 2, \dots, M_i)$, where M_i is the total number of interest domains in the i -th frame image. The bird hazard of the j -th interest domain in the i -th frame is expressed as $B_i^{j(l)} (l = 1, 2, \dots, L_i^j)$, where L_i^j is the total number of bird damage.

A CG-CFDM network consists of two hierarchical detectors. The first-level detector accepts the image frame I_i of the railway video as input, and its output is the interest

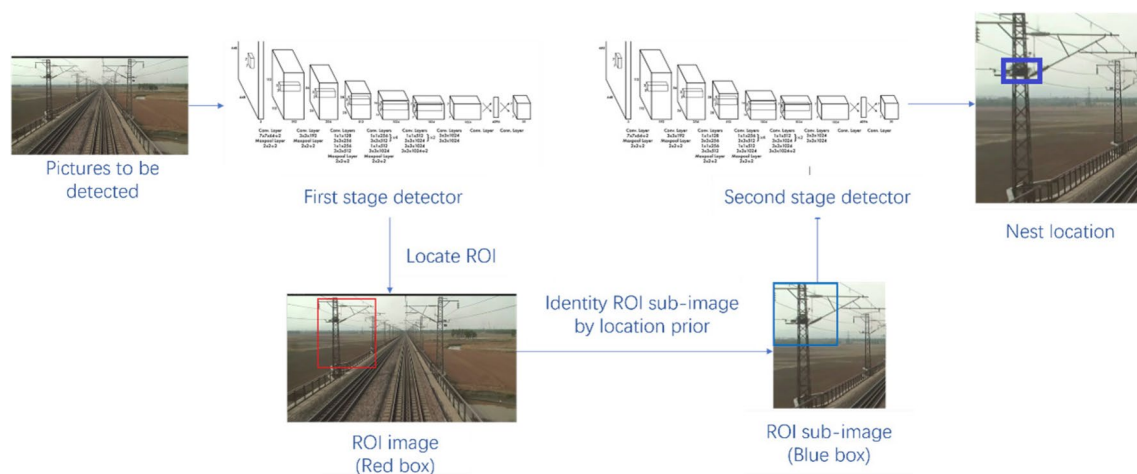


Fig. 2 The construction process of CG-CFDM detection network

domain set $\chi_i = \{R_i^0, R_i^1, \dots, R_i^{M_i}\}, (R_i^0 = \varphi)$. As the detection process of the first-level detector, the CG-CFDM network uses the Yolov4 neural network to analyze the image frame and extract the image domain of interest. The image frame I_i is input to the first-level detector-Yolo neural network for feature extraction and analysis, and the network outputs the result set χ_i , where $R_i^j = (x_i^j, y_i^j, w_i^j, h_i^j)$, the quadruple $(x_i^j, y_i^j, w_i^j, h_i^j)$ represents the abscissa and ordinate of the sub-image of the interest domain relative to the original image, as well as its width and height. The input of the second-level detector is the sub-image $I_i(R_i^j)$ corresponding to the element R_i^j of the result set, and the output result is $B_i^{j(k)}$. The second-level detector still uses the Yolo neural network to analyze and recognize the sub-images of the interest domain. After the feature extraction and analysis of the image $I_i(R_i^j)$ result Yolov4, the result set of bird damage is output $K_i^j = \{B_i^{j(0)}, B_i^{j(1)}, \dots, B_i^{j(L_i^j)}, (B_i^{j(0)} = \varphi)\}$, where $B_i^{j(l)} = (x_i^{j(l)}, y_i^{j(l)}, w_i^{j(l)}, h_i^{j(l)})$, the quadruple $(x_i^{j(l)}, y_i^{j(l)}, w_i^{j(l)}, h_i^{j(l)})$ represents the positional relationship of the bird damage prediction image relative to the sub-image of the interest domain. The result of bird damage recognition is the position (x, y, w, h) , of the bird damage relative to the original image, and the conversion $(x, y, w, h) = T(B_i^{j(l)})$ can be expressed as

$$\begin{cases} x = x_i^{j(l)} + x_i^j \\ y = y_i^{j(l)} + y_i^j \\ w = w_i^{j(l)} \\ h = h_i^{j(l)} \end{cases} \quad (1)$$

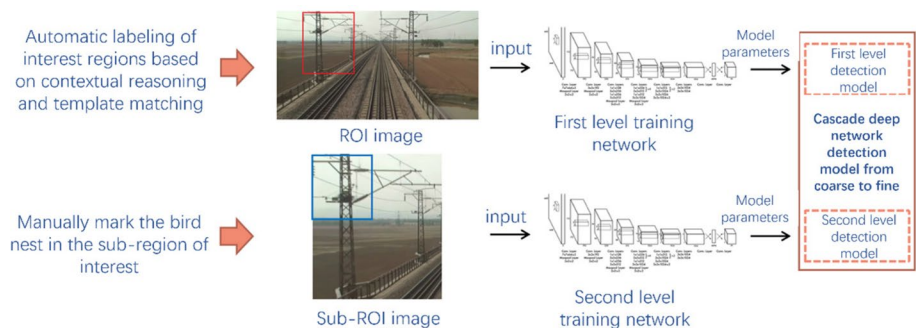
The output of the CG-CFDM network is all the results of identifying bird damage in the i -th frame, and the output set is $\Omega = \{T(B_i^{j(l)}) | 1 \leq j \leq M, 1 \leq l \leq L_i^j\}$.

3.3 CG-CFDM Training Network Structure

The construction process of CG-CFDM training network is divided into the construction of the first level training network and the second level training network, which can be summarized as extracting the data set of interest domain and the data set of bird's nest from a railway video with the bird's nest label to conduct the neural network training of the two level detectors respectively. The processing flow is shown in Fig. 3. The first-level training network automatically marks the domain of interest (red box) in the original video frame image based on context reasoning and template matching algorithm as the training data of the first-level network object coarse positioning to train the domain of interest location model. The second level training network determines the sub-region of interest (blue box) based on geometric position constraints in the region of interest image, and trains the bird nest positioning model by manually marking the bird nest in the sub-region of interest as the training data for the second level network object fine positioning.

Suppose the initial data set is a piece of railway video data $v = \{I_1, I_2, \dots, I_N\}$, which contains the target object's bird damage frame set $\{I_{b_1}, I_{b_2}, \dots, I_{b_s}\}$, where $I_{b_i} (i = 1, 2, \dots, s)$ is the frame number of bird-damaged frames, and s is the total number of bird-damaged frames. The initial data set only marked the position of the bird damage frame B , and the labeled set of the i -th bird damage frame was $L_i = \{l_i^1, l_i^2, \dots, l_i^{k_i}\}$. In the formula, k_i is the number of bird damage frames and l_i^j is the label position, which can be expressed as $l_i^j = (x_i^j, y_i^j, w_i^j, h_i^j)$. The interest domain of the bird-damaged frames is determined by inferentially labelling the bird damage frame set B . The reasoning process begins by segmenting the image using the Selective Search algorithm [8]. The input of the Selective Search algorithm is the bird damage frame I_{b_i} , and the result is the continuous area $Z_i = \{z_i^0, z_i^1, \dots, z_i^{d_i}\}$, where d_i indicates the number of

Fig. 3 The construction process of CG-CFDM training network



rectangular areas output by the algorithm, $z_i^j = (x_i^j, y_i^j, w_i^j, h_i^j)$, which represents the position of the rectangular area relative to the bird damage frame. Finally, here is the bird-damage frame I_{b_i} to find the interest domain set of the image set containing the labeled area, which is defined as

$$H_i = \{I_{b_i}(u) | u \in Z_i \wedge \exists v \in L_i(I_{b_i}(v) \subseteq I_{b_i}(u))\}. \quad (2)$$

By definition, all sub-images of interest domain $I_{b_i}(u)$ contain bird-damaged areas. By analyzing the search results, the following bird-damage annotation sets can be obtained for each image in the interest domain:

$$\text{Label}(I_{b_i}(u)) = \{(v_x - u_x, v_y - u_y, w_v, h_v) | v \in L_i \wedge I_{b_i}(v) \subseteq I_{b_i}(u)\}. \quad (3)$$

The inference results of all bird-damage frames constitute the template library T , namely:

$$T = \bigcup_{i=1}^S H_i. \quad (4)$$

The template library is then used to perform template matching on all frames of the railway video. The template matching matches each image frame I_i . The matching result is the location set as the interest area label of the video frame I_i . Set the binary relationship $a \cong b$ to represent the image a and the image b matches, the obtained video frame label can be expressed by

$$\text{Rlabel}(I_i) = \{(x, y, w, h) | \exists g \in T (g \cong I_i(x, y, w, h))\}. \quad (5)$$

All video frames and their matching results constitute the interest domain data set for the first-level detector to train the Yolo neural network

$$\text{Dataset_ROI} = \{(\mathbf{I}, \text{Rlabel}(\mathbf{I})) | \mathbf{I} \in \mathbf{v}\}. \quad (6)$$

All template libraries and their annotation results constitute a bird pest data set for the second-level detector to train the Yolo neural network

$$\text{Dataset_Birdnest} = \{(\mathbf{I}, \text{Label}(\mathbf{I})) | \mathbf{I} \in T\}. \quad (7)$$

4 Interest domain search based on contextual reasoning

In this section, we focus on the use of the Selective Search algorithm and template matching to construct the first-level detector and the second-level detector data sets for CG-CFDM. In the inherent geometric structure of the catenary, the spatial distribution of the bird's nest is limited to the continuous context space sub-region of the catenary, which has strong contextual location prior information and can determine the detection region of interest based on the context information.

4.1 Contextual reasoning

As a whole, the domain of interest is a contextual region with a high degree of similarity. First, we need to find all regions with a high degree of similarity in the image. An image is segmented and merged as part of the search area.

A preliminary segmentation of the image is performed to obtain a large number of similar regions, and then the regions are merged to produce a series of candidate regions, and the region of interest is determined from the candidate region containing the bird's nest.

An image can be represented by an undirected graph $G = \langle V, E \rangle$. The vertex of the graph represents a pixel of the image, and the weight of edge $e = (v_i, v_j)$ indicates the dissimilarity of adjacent vertex pairs i, j . The color distance of the pixel $\sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}$ and other pixel attributes can be used to indicate the dissimilarity between two pixels $w(e)$. Since an area is a set of points with the least dissimilarity, it always contains the point set's minimum spanning tree. The first step in segmenting the image is to identify the forest composed of the minimum spanning tree in the image. The difference between local areas determines whether or not they should merge. The intra-class difference is defined as:

$$\text{Int}(C) = \max_{e \in \text{MST}(C, E)} W(e). \quad (8)$$

The difference between classes is defined as the smallest connecting edge of the two regions:

$$\text{Diff}(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} W((v_i, v_j)). \quad (9)$$

In particular, if the two regions are not connected by edges, $\text{Diff}(C_1, C_2) = \infty$. Therefore, the following is the basis for merging the two regions:

$$\text{Diff}(C_1, C_2) \leq \min(\text{Int}(C_1) + \tau(C_1), \text{Int}(C_1) + \text{Int}(C_2)). \quad (10)$$

Among them, $\tau(C)$ is the threshold function, which assigns weights to the area formed by isolated points:

$$\tau(C) = \frac{k}{\|C\|'}. \quad (11)$$

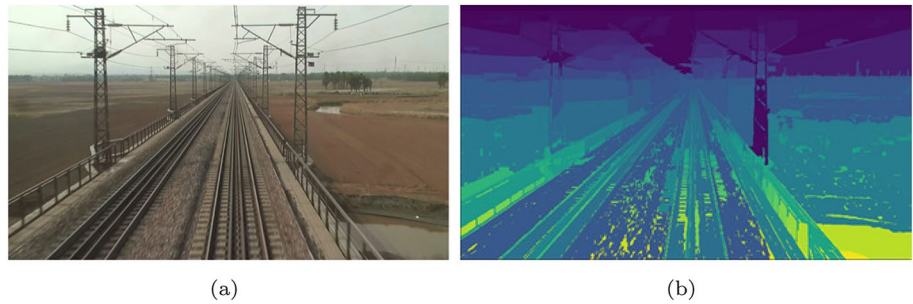
In this way, the image point sets are merged to obtain multiple basic regions [27].

The segmentation result is shown in Fig. 4.

Candidate regions formed above are merged, and their merged results are represented by rectangles. The positions of the rectangles can be identified by the quadruple of (x, y, w, h) . The corresponding identification rectangle position attribute for area C is calculated as follows:

$$x = \min_{v \in C} x(v), \quad (12)$$

Fig. 4 The schematic diagram of segmentation effect: **a** original image; **b** image after segmentation



$$y = \min_{v \in C} y(v), \quad (13)$$

$$w = \max_{v \in C} x(v) - \min_{v \in C} x(v), \quad (14)$$

$$h = \max_{v \in C} y(v) - \min_{v \in C} y(v). \quad (15)$$

The first step is to calculate the differences between regions, which can be evaluated by four indicators. In the color histogram of the two regions, the color distance is determined by the minimum value of each bin.

$$S_{\text{colour}}(r_i, r_j) = \sum_{k=1}^n \min(c_i^k, c_j^k). \quad (16)$$

The texture distance depends on the minimum value of each bin in the fast sift feature histogram of the two regions.

$$S_{\text{texture}}(r_i, r_j) = \sum_{k=1}^n \min(t_i^k, t_j^k). \quad (17)$$

Merging between small areas is given priority, and small areas are given a higher merging weight.

$$S_{\text{size}}(r_i, r_j) = 1 - \frac{\text{size}(r_i) + \text{size}(r_j)}{\text{size}(im)}. \quad (18)$$

Regions with large overlaps between the circumscribed rectangles are first merged.

$$S_{\text{fil}}(r_i, r_j) = 1 - \frac{\text{size}(BB_{i,j}) - \text{size}(r_i) - \text{size}(r_j)}{\text{size}(im)}. \quad (19)$$

Based on the above differences, the total difference between regions is calculated as follows:

$$S(r_i, r_j) = a_1 S_{\text{colour}}(r_i, r_j) + a_2 S_{\text{texture}}(r_i, r_j) + a_3 S_{\text{size}}(r_i, r_j) + a_4 S_{\text{sill}}(r_i, r_j). \quad (20)$$

The position of the bird's nest of the input data set is marked manually, and the area attribute of the marked bird's nest is still represented by a rectangle, whose position attribute is (bx, by, bw, bh) . The interest domain consists of the

candidate area, including the bird's nest area, and its position coordinates should satisfy the following conditions:

$$\begin{cases} x \leq bx \\ y \leq by \\ x + w \geq bx + bw \\ y + h \geq by + bh \end{cases}. \quad (21)$$

In order to prevent excessive merging, the interest domain should also meet the following threshold conditions:

$$\frac{w \times h}{\text{imagesize}} \leq T. \quad (22)$$

In order to obtain candidate regions, the basic regions are merged according to the differences between them. The merge algorithm is as follows [28]. The candidate region where the nest is located is the inferred domain of interest, as shown in Fig. 5. The red box indicates the inferred domain of interest. According to the statistical rules of a large number of data, the bird's nest basically occurs in the upper left part of the region of interest. Therefore, based on the inherent geometric structure of the pole, the target detection range can be further reduced to the blue box area as shown in Fig. 5, which is called the sub-region of interest.



Fig. 5 The domain of interest by reasoning



Fig. 6 Template element image

4.2 Automatic sample labeling by template matching

Export the image of interest sub-region obtained in the previous section as template element image (blue box area in Fig. 5), as shown in Fig. 6, all template element images constitute template library.

The interested domain is inferred from the position of the bird's nest, and all interest domain images form a template library. Every image in the image set is traversed and the normalized correlation coefficient matching is used to identify all the interests of each image. Suppose the template image is T , the image to be matched is I , the width of the cut template image is w , the height is h , and R represents the matching result, then the matching method can be expressed as follows:

$$R(x, y) = \frac{\sum_{x', y'} T'(x', y') * I'(x + x', y + y')}{\sqrt{\sum_{x', y'} T'(x', y')^2 * \sum_{x', y'} I'(x + x', y + y')^2}}, \quad (23)$$

where

$$T'(x', y') = T'(x', y') - \frac{1}{w * h} \sum_{x'', y''} T'(x'', y''), \quad (24)$$

$$I'(x + x', y + y') = I'(x + x', y + y') - \frac{1}{w * h} \sum_{x'', y''} I'(x + x'', y + y''). \quad (25)$$

A larger value indicates a greater similarity between the rectangular area (w, h) and the template at the position (x, y) of the graph, while the maximum value of the template similarity results from template matching. The template matching value must be greater than the threshold value.

$$\text{remember } Rs(T, I) = \max_{x, y \in I} R(x, y). \quad (26)$$

The template matching process begins by comparing the images in the template library to the original images. Each template image corresponds to a best matching value R , and the position of the corresponding rectangular matching box is (x, y, w, h). The first matching result of the template constitutes the result set S :

$$S = \{(x, y, w, h, Rs) | Rs(T, I) = \max_{x, y \in I} R(x, y) \wedge Rs(T, I) > c\}. \quad (27)$$

In this equation, c is the matching threshold parameter. The rectangles in the result set S may intersect if they are arranged in descending order of Rs value. For the intersection of two rectangles s , the conditions are:

$$\max(x(s), x(t)) \leq \min(x(s) + w(s), x(t) + w(t)), \quad (28)$$

$$\max(y(s), y(t)) \leq \min(y(s) + h(s), y(t) + h(t)). \quad (29)$$

The result set S is traversed in turn, and if the current rectangle intersects the marked rectangle, the marking is discarded; otherwise, the current rectangle is marked. The Fig. 7 below shows the result of template matching:

To obtain the interest domain data set, the image set was marked in the VOC format to indicate possible interest domains in the bird's nest. We use the interest domain of bird's nests in the template library as the bird's nests data set.

Fig. 7 The schematic diagram of template matching: **a** original image; **b** matching result



5 Cascaded deep network construction from coarse to fine

This section explains the detection principle of the CG-CFDM network and its high-precision characteristics. We also analyze the parallel computing potential of the CG-CFDM structure, and explain the pipeline parallel acceleration algorithm used by CG-CFDM.

5.1 Detection principle

The bird's nest on the catenary is small in size and lacks significant shape and texture features. For classifying fringe images, existing artificial design features are not sufficient to achieve ideal results. In this regard, deep learning provides a feasible solution. The YOLO neural network has a strong detection performance as a common target detection network. In this study we use a two-level predictive network cascade detection network for detecting bird's nests on the catenary. The following analysis takes YOLOv3-SPP network structure as an example to describe the detection principle as a hierarchical detector.

YOLO neural networks take the entire image as input and predict the bounding box and its category based on the input image. As shown in Fig. 8, the algorithm begins to divide an image into $S \times S$ grids, and the grid where the center of the labeled object is located is responsible for predicting the corresponding labeled object. Each grid needs to predict B bounding boxes, and each bounding box needs to return to its own

relative offset between the center point of the bounding box and the position of the upper left corner of the corresponding grid. t_x and t_y are normalized so that the predicted value of the bounding box is between 0 and 1, thus ensuring that the center point of the bounding box must be in the divided grid.

$$b_x = \sigma(t_x) + c_x, \quad (30)$$

$$b_y = \sigma(t_y) + c_y, \quad (31)$$

$$b_w = p_w e^{t_w}, \quad (32)$$

$$b_h = p_h e^{t_h}, \quad (33)$$

t_x, t_y, t_w, t_h are the predicted output of the model. c_x and c_y represent the coordinates of the grid, p_w and p_h indicate the size of the bounding box before prediction; b_x, b_y, b_w and b_h are the center coordinates and size of the predicted bounding box. There is a correlation between confidence and whether the grid correctly predicts the object to be detected as well as the deviation of the bounding box from the true position of the object. The confidence level can be expressed by the following equation:

$$\text{confidence} \Pr(\text{Object} * \text{IOU}_{\text{pred}}^{\text{truth}}), \quad (34)$$

In the equation, IOU represents the intersection ratio between the bounding box and the object labeling box, and the calculation method is as follows:

$$\text{IOU}_{\text{pred}}^{\text{truth}} = \frac{(\min(tx+tw, px+pw) - \max(tx, px)) * (\min(ty+th, py+ph) - \max(ty, py))}{tx*ty+px*py - (\min(tx+tw, px+pw) - \max(tx, px)) * (\min(ty+th, py+ph) - \max(ty, py))}, \quad (35)$$

position (x, y, w, h) . Through a change in the step size of the convolution kernel, YOLOv3 changes the size of the propagation tensor in the network. It improves the prediction speed of the model by calculating the bounding box in advance. It determines the position of the bounding box by predicting the

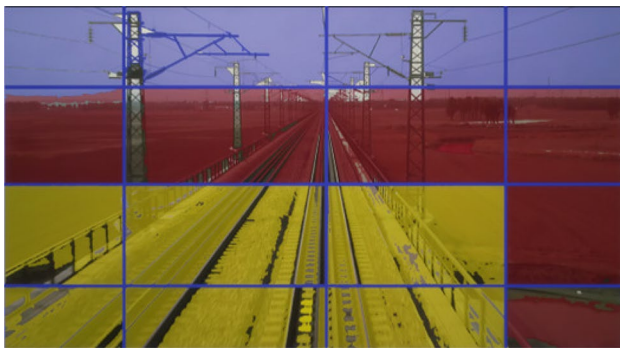


Fig. 8 Division of grid prediction results

where (tx, ty, tw, th) represents the truth position attribute of the label box (x, y, w, h) , and (px, py, pw, ph) represents the predicted position attribute of the bounding box (x, y, w, h) . $\Pr(\text{Object})$ represents the probability that the object to be predicted exists in the grid. During the training process, if no object to be tested falls into the grid, then $\Pr(\text{Object})$ is 0; Otherwise it is 1. In the training process, each grid corresponds to a pixel matrix, and each pixel matrix is used as the input of the neural network. For each grid, the output given by the network for each bounding box is $(x, y, w, h, \text{conf}, c_1, \dots, c_n)$, where (x, y, w, h) gives the position of the bounding box, conf is the confidence of the bounding box, $c_1, \dots, c_n$ represents the class probability of the object.

YOLOv3 uses multi-scale features to detect objects. The receptive field of the feature map obtained after 32 times downsampling is relatively large, making it suitable for detecting large targets in the image; the feature map obtained after 16 times downsampling has a medium-scale receptive field, which is appropriate for detecting

Table 1 Feature map and prior table

Feature map	13*13	26*26	52*52
Feel the wild big middle small	Feel the wild big middle small (116 × 90)	Feel the wild big middle small (30 × 61)	Feel the wild big middle small (10 × 13)
A priori box	(156 × 198) (373 × 326)	(62 × 45) (59 × 119)	(16 × 30) (33 × 23)

medium-sized targets in the image; after 8 times down-sampling, the feature map has a relatively small receptive field, which is appropriate for detecting objects of relatively small size in the image. YOLOv3 uses the K-means clustering method to determine the size of the a priori box, and sets three a priori boxes of different sizes for each down-sampling scale, and clusters an a priori box of nine sizes. In Table 1, the distribution of nine a priori boxes obtained when the input image has a resolution of 416×416 is shown for feature maps of different sizes.

It is not ideal to directly utilize the YOLO network to detect catenary bird's nests, since they occupy a small

$$\text{confidence} = \Pr(\text{birdnest} * \text{IOU}_{\text{pred}}^{\text{birdnest}}). \quad (36)$$

Generally, the IOU value of the large-size anchor frame and the medium-size anchor frame is not very high for small-scale objects, and the overall precision is not very good.

As shown in the Fig. 9, there is only one grid in the upper left corner for valid calculations. The forecast expectation of a single network is:

$$E(\text{birdnest}) = \sum_{i=0}^{S^2} \sum_{j=0}^B \Pr_i(\text{birdnest}) * \text{IOU}_{\text{pred}_i}^{\text{birdnest}}. \quad (37)$$

Note: $\text{IOU} = \frac{E(\text{birdnest})}{\sum_{i=0}^{S^2} \sum_{j=0}^B \Pr_i(\text{birdnest})}$ is a valid forecast average IOU of the box

$$E(\text{birdnest}) = \frac{BS^2 * I(\text{birdnest}) * \text{IOU}}{I(\text{image})}, \quad (38)$$

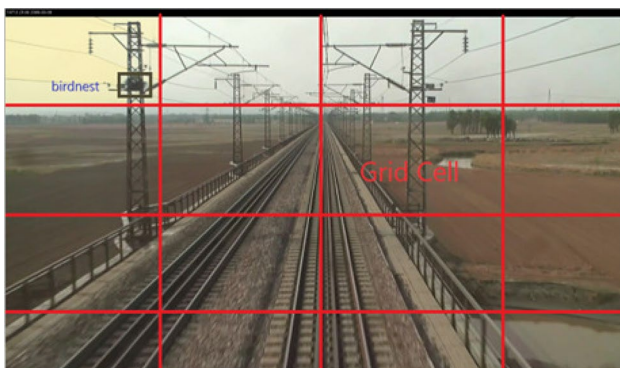
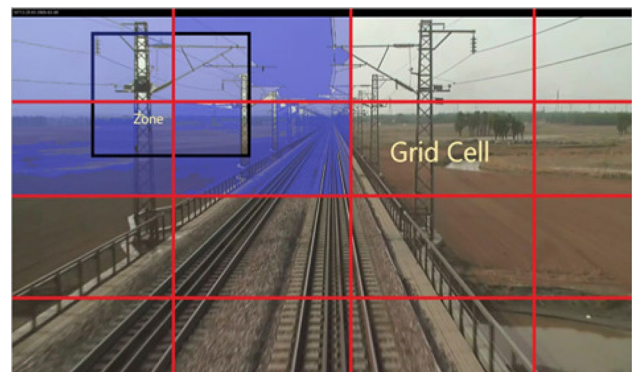
where $\frac{I(\text{birdnest})}{I(\text{image})}$ is the proportion of bird's nest images.

The distribution of the bird's nest in the catenary is affected by physical factors, i.e., its position within the catenary is bound. A bird's nest is defined as all possible areas within the interest area in Sect. 4.1. The equation can be expressed as follows:

$$\begin{aligned} P(\text{distribution}) &= P(\text{Bird's nest exists in the interest domain of the image} \\ &\quad | \text{The image contains bird's nest}), \\ &= 1. \end{aligned} \quad (39)$$

proportion of the image, resulting in a large number of grids and a waste of computing resources. In addition, the YoloV3 neural network uses an anchor box that is pre-determined based on a priori clustering of the data,

Based on the prior condition of bird nest distribution, the cascaded YOLO neural network can be used to perform the first level of “coarse detection” to identify the region of interest, and then perform the second level of “fine detection” to identify the location of bird nest from the predicted sub-region image of the region of interest given by the network.

**Fig. 9** Single-stage network prediction**Fig. 10** Cascade network-first-level network prediction

5.2 Detection precision analysis

First level prediction:

The segmentation result is shown in the Fig. 10. The black box is the interest domain given by the template matching, and the red segmentation line is the grid divided by the YOLO network. Only the four grids in the upper left corner are responsible for predicting the interest domain with confidence, and the confidence of other grids is 0. The confidence and expectations of the first level network detection are as follows:

$$\text{confidence}(\text{Zone}) = \Pr(\text{Zone}) * \text{IOU}_{\text{pred}}^{\text{zone}}, \quad (40)$$

$$E(\text{Zone}) = \frac{BS^2 * I(\text{Zone}) * \overline{\text{IOU}}_{\text{zone}}}{I(\text{image})}. \quad (41)$$

Second-level network:

The second level YOLO network takes the sub-image set of interest region predicted by the first level network as the input, and performs YOLO cascade detection on the sub-image set, whose essence is the secondary division of the grid. During the training process, increase the IOU of the boundary box as much as possible to improve the training precision and carry out as many effective calculations as possible. The segmentation effect is shown in Fig. 11. The confidence level of the second-level network is:

$$\begin{aligned} \text{confidence}(\text{Birdnest}) &= \Pr(\text{Birdnest}) * \text{IOU}_{\text{pred}}^{\text{birdnest}} * \text{confidence}(\text{Zone}) \\ &* P(\text{distribution}), = \Pr(\text{Birdnest}) * \text{IOU}_{\text{pred}}^{\text{birdnest}} * \text{confidence}(\text{Zone}). \end{aligned} \quad (42)$$

The bird's nest prediction expectation in the interest domain is:

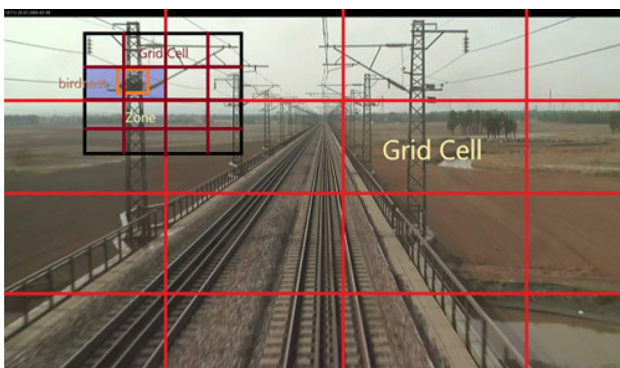


Fig. 11 Cascade network-second-level network prediction

$$\begin{aligned} \hat{E}(\text{birdnest}) &= \sum_{i=0}^{S^2} \sum_{j=0}^B \Pr(\text{birdnest}) * \text{IOU}_{\text{pred}_i}^{\text{birdnest}} * \text{confidence}(\text{Zone}) \\ &= \text{confidence}(\text{Zone}) * \sum_{i=0}^{S^2} \sum_{j=0}^B \Pr(\text{birdnest}) * \text{IOU}_{\text{pred}_i}^{\text{birdnest}}. \end{aligned} \quad (43)$$

The expectations of the cascade forecast are:

$$\begin{aligned} E(\text{birdnest}) &= \frac{BS^2 * I(\text{birdnest}) * \overline{\text{IOU}}_{\text{birdnest}}}{I(\text{zone})} * \frac{BS^2 * I(\text{Zone}) * \overline{\text{IOU}}_{\text{zone}}}{I(\text{image})} \\ &= \frac{(B^2 S^4 * I(\text{birdnest}) * \overline{\text{IOU}}_{\text{birdnest}} * \overline{\text{IOU}}_{\text{zone}})}{I(\text{image})}, \end{aligned} \quad (44)$$

where $\overline{\text{IOU}}$ represents the average IOU value of the grid prediction anchor frame, because the anchor frame size is given a priori by the data set clustering, and it is positively correlated with the size of the input image. The closer the ratio of the object to be detected to the anchor frame, the larger the average IOU value, so the $\text{IOU}_{\text{birdnest}}$ and IOU_{zone} given in the formula are both greater than the original IOU cascade prediction precision.

A single-stage neural network's average prediction precision is positively correlated with the number of training samples. Suppose that the average prediction precision of the neural network is $F(\text{Object}, \text{Base}, n)$ when the training sample is labeled Object in the Base image, and the number is n .

CG-CFDM is expected to have a higher precision than the single-stage network based on the discussion above:

$$F(\text{birdnest}, \text{Zone}, n) * F(\text{Zone}, \text{image}, n) > F(\text{birdnest}, \text{image}, n). \quad (45)$$

In the data set of catenary bird's nest, there are very few sample data sets with bird's nest annotation, and most of the images do not contain bird's nest. However, the number of samples in the interest domain data set is extremely large, and the interest domain that has nothing to do with the bird's nest of the object to be measured offers information about the distribution of the bird's nest, as well as increasing the overall detector's precision by training on the interest domain. Suppose the number of samples in the data set is M , and the number of samples containing bird's nests is N .

Then, the overall precision of CG-CFDM is:

$$\begin{aligned} P &= F(\text{birdnest}, \text{Zone}, N) * F(\text{Zone}, \text{image}, M) \\ &> F(\text{birdnest}, \text{image}, N). \end{aligned} \quad (46)$$

The interest domain data set is relatively easy to obtain, $M \gg N$. It is possible for the first-level detector to distinguish all the interest domains of the image when there are enough samples.

$$F(\text{Zone}, \text{image}, M) \rightarrow 1. \quad (47)$$

In the case of CG-CFDM, the precision reaches the extreme value after the first-level detector is amplified to the extreme:

$$P_{\max} = F(\text{birdnest}, \text{Zone}, N). \quad (48)$$

The precision of CG-CFDM is determined by the precision of the second-level detector. The first-level detector plays a role in object amplification, and the second-level detector detects the bird's nest of the object to be tested in the interest domain. The average IoU value of the object is significantly greater than that of the single stage.

5.3 Detection efficiency analysis

CG-CFDM is a two-stage detection network that cascades two one-stage detection networks. The two-stage detection has a clear physical meaning: the first level identifies the catenary, while the second level identifies the damage that birds cause to the catenary. Therefore, the CG-CFDM detection process can be flexibly controlled: since the second-level detector uses the first-level detector's results, a data correlation exists between the two detectors, so a detection pipeline can be developed between the first and second-level detections.

It is necessary to latch the images recognized by the first-level detector and the intermediate results of the first-level recognition because the pipeline structure inevitably involves latching data. Due to the large amount of data in the image, the infinite accumulation of latches will cause the program to consume an excessive amount of memory. The circular buffer queue can therefore be used to latch the first-level results. Several second-level detectors retrieve results from the buffer queue and save the final results after secondary identification. Using computer clusters, the parallel calculation of the second-level detector can be implemented using the OpenMP programming model, or by using the MPI programming model.

In the event that multiple second-level detectors write to the same memory, errors will occur, but if the circular buffer queue is set up as a critical resource for mutually exclusive access, it will also limit the detection rate and result in greater performance consumption. Therefore, the asynchronous detection of the second-level detector can be used to circumvent this problem. We will first describe the structure of the circular buffer queue. The circular buffer queue is composed of three arrays: one is the image array, the result array, and the signal array. The algorithm allows users to set the number of second-level detectors and set the number of second-level detectors to k (k limits the number of first-level detectors to identify the catenary, and generally does not exceed 10). Its structure is described as Fig. 12 (When k is 3): (When k is 3).

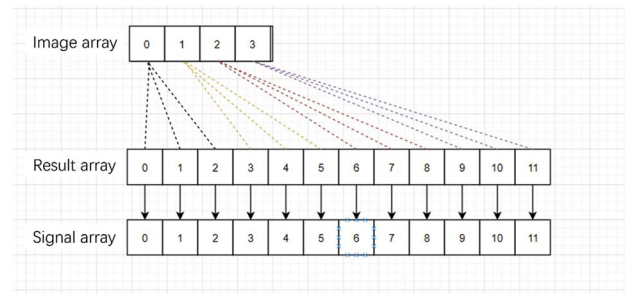


Fig. 12 The Structure of the circular buffer queue array

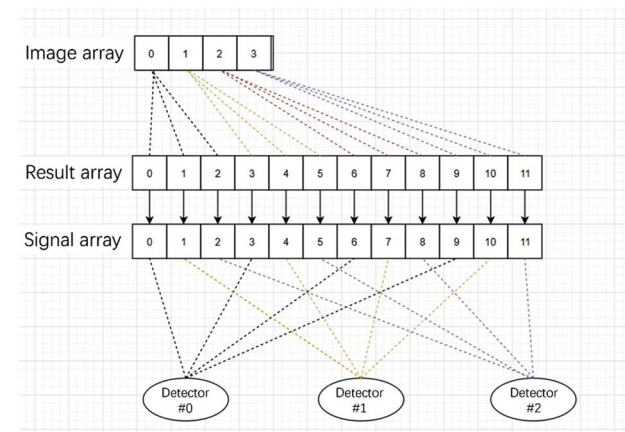


Fig. 13 The structure of the second-level detector

The image array unit stores the images analyzed by the first-level detector. The result array unit stores the intermediate results produced by the first-level detector, that is, the location information (x, y, w, h) of the domain of interest and the type of the catenary. The signal array stores the status of the result array, which is an integer, 0: undetected, 1: invalid detection, 2: valid detection. In the $[ik, (i+1)k]$ -th unit, the validity of its unit is identified by the $[ik, (i+1)k]$ -th unit of the signal array. The length of the result array or signal array is defined as the length of the buffer queue $buffer_size$. According to the above definition and mapping relationship, $k|buffer_size$ is required to ensure the memory alignment and the continuity of the results. Since k is a variable defined by the user dynamically, and the number of interest domains for each image generally does not exceed 10, it can be set as $buffer_size = \prod_{i=1}^1 0i = 362880$.

With the circular buffer queue structure introduced, the asynchronous detection of the second-level detector can be designed: A total of k second-level detectors are used to analyze the data of the result array, where the number is $0 \sim k-1$, and the i -th detector is responsible for the $i, i+k, i+2k, \dots, i+nk$ data unit detection tasks $i = 0, 1, 2, \dots, k-1$.

Its structure definition description can be represented by Fig. 13.

It is important to note that in this detection structure, different detectors can only read the same image array, and write different signal arrays, without conflict or mutual exclusivity.

The overall description of the pipeline parallel acceleration algorithm is given as follows. The signal array is initialized as undetected, and the first-level detector continuously reads and analyzes the video stream data. After reading the i -th frame, the image is put into the $i \bmod buffersize$ unit of the image array (mod is the remainder operation). Then the image is analyzed, and q valid results are generated. If $q \leq k$, the analysis result (x, y, w, h) is stored in $ik + j \bmod buffersize$ ($j = 0, 1, 2, \dots, q - 1$) of the result array. The $ik + j \bmod buffersize$ ($j = 0, 1, 2, \dots, q - 1$) unit setting of the signal array has been checked, and the $ik + j \bmod buffersize$ ($j = q - 1, \dots, k - 1$) setting is invalid. If $q > k$, the first k results are taken according to the confidence level of the first-level detector's detection results, and they are treated as $q \leq k$. The x -th second-level detector initially fetches data from the x -th unit of the signal array, and if its value is not detected, it waits until this unit is invalid or valid; if this unit is invalid, continue to detect the $x + k$ th unit $x = x + k$; if it is a valid detection, take the intermediate result (x, y, w, h) in the x -th unit of the result array, take the image matrix in the $x \div k$ -th unit of the image array, and analyze the (x, y, w, h) area; if there is any problem, record the result of bird damage and continue to detect unit $x + k$ ($x = x + k$). According to the pipeline, the detection rate of the second-stage detector must not be lower than that of the first-stage detector. In the MPI computing cluster environment, this algorithm can significantly increase the detection rate, with a limit rate close to the single-stage single-network detection rate, which can be sufficient to meet real-time detection requirements.

The algorithm description is shown in Fig. 14.

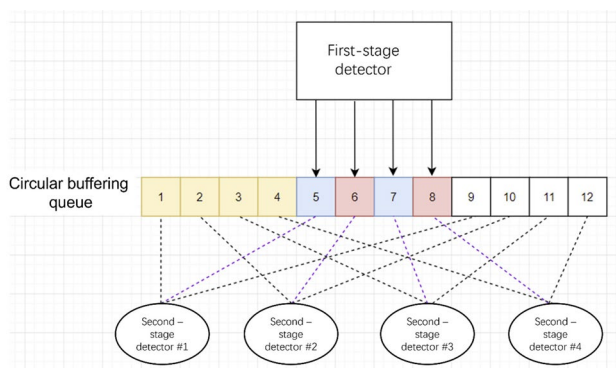


Fig. 14 Diagram of the pipelined parallel acceleration algorithm

6 Experimental analysis

This section describes the process and results of constructing the CG-CFDM network data set, and then conducts a comparison test experiment between different traditional deep learning networks and the CG-CFDM network. Finally, the optimal structure of the CG-CFDM network is explored by changing the structure of the first-level detector and the second-level detector.

6.1 Experimental environment

The experimental hardware environment of this research is the Intel Core i9-10900K processor with a main frequency of 3.70 GHz, RAM is 64GB, Nvidia GeForce RTX 3090 graphics card; The experimental software environment is Win10 Professional Edition, Python3.8, Pytorch1.7.1, CUDA 11.

6.2 Data annotations

6.2.1 Data set for first level interest domain detection

The first level network data set is composed of 3900 images containing catenary, and the interest domain is automatically marked according to contextual reasoning and template matching algorithm. Due to the environmental error of the template library, some samples have marked deviation, which requires manual verification. The detection frame with matching errors shall be removed, and the images with less matching quantity shall be supplemented and marked. Figure 15 shows the template matching results.

Because of the limitations of template matching, which may not accurately mark all interest regions for unknown samples. The template matching result after



Fig. 15 Template matching result



Fig. 16 The template gallery

manual verification is used as the data set of the first level detector training. The image features of the interest region are learned by the deep learning algorithm to identify the interest region of unknown samples. It has strong generalization ability and can identify all contact network areas along the railway line that may have bird damage more accurately.

6.2.2 Data set for second level nest detection

In the test data set, there are 126 images to be tested including the bird's nest, which is the sub-area of interest of the catenary where the bird's nest is located. Some of the images have many environmental background factors, and the environmental noise is large, which need to be removed manually. The manually deleted images constitute the training data set of the secondary network, as shown in Fig. 16.

6.3 Bird nest detection under different models

6.3.1 Evaluating indicator

Firstly, all candidate boundary boxes of network prediction are arranged in descending order of classification probability value for each type of object, then the IOU threshold is set to filter out the valid bounding boxes. The number of valid bounding boxes noted denoted as True Positive (TP), the number of invalid bounding boxes denoted as False Negative (FN) and the number of undetected objects denoted as False Positive (FP) will be counted to calculate the precision and recall. These indicators are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%, \quad (49)$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\%. \quad (50)$$

The precision is measured by the ratio of predicted true positive samples to the total number of positive samples. The higher the precision, the less false detection of the mode. The recall is the ratio of the number of predicted true positive samples to the total number of true positive and false negative samples. The higher the detection recall, the less missing detection of the model.

To ensure real-time detection Recall while maintaining a relatively high detection precision, the detection speed of the model should be increased as much as possible. To measure the speed of detection, the number of processed images per second (FPS per second) is generally used.

6.3.2 Testing process and results

The data set of the first-level detector is 3900 images, and the target detection algorithm based on deep learning, such as Faster R-CNN, YOLOv3-spp, and YOLOv4, is used to identify and train the interest domains along the railway.

The data set of the second level detector is 126 images, and the bird nest in the interest domain obtained by the contextual reasoning algorithm is recognized and trained by using Fast R-CNN, YOLOv3-spp, YOLOv4 and other target detection algorithms based on deep learning.

In addition, Faster R-CNN, YOLOv3-spp, YOLOv4, and other deep learning-based object detection algorithms are used to directly detect and recognize bird damages along the railway. The experiment contains 400 images of bird's nests, of which $240 + 80 = 320$ are taken as the training set and the validation set are used for deep learning model training, and the remaining 80 test images are obtained by data enhancement and other methods to obtain 126 images as the test set for testing the model.

6.3.3 First-level detector training

The first-level detector uses Faster R-CNN, yolov3-spp and yolov4 detection models for training interest domain recognition. Different network structures are used to detect interest domains.

YOLOv3-SPP has 225 layers and 62.5 million parameters, while YOLOv4 has 327 layers and 64 million parameters. There is little difference in the backbone network structure between YOLOv3 and YOLOv4, and YOLOv4 has optimized and improved the training process: The prediction box regression of YOLOv3 uses GIOU-Loss, which does not distinguish the relative position relationship of objects, while YOLOv4 uses CIOU-Loss, which takes into account

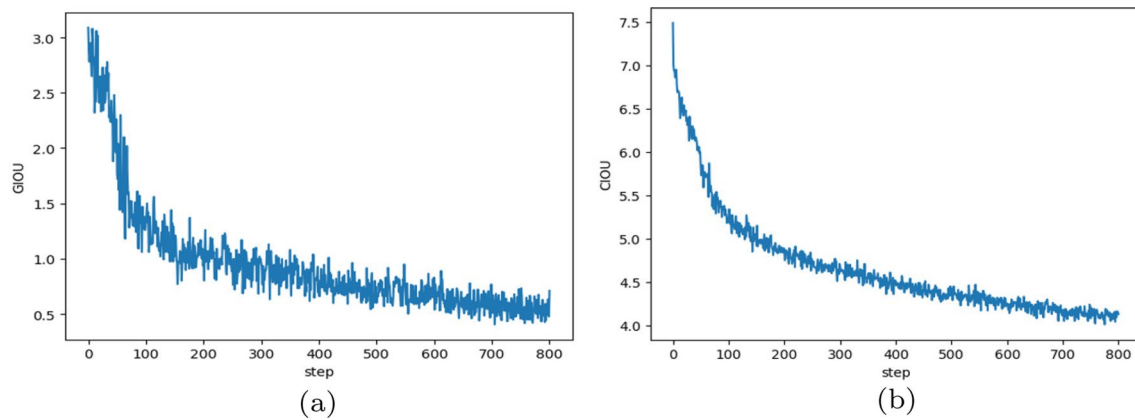


Fig. 17 IOU curves: **a** YOLOv3-GIOU, **b** YOLOv4-GIOU

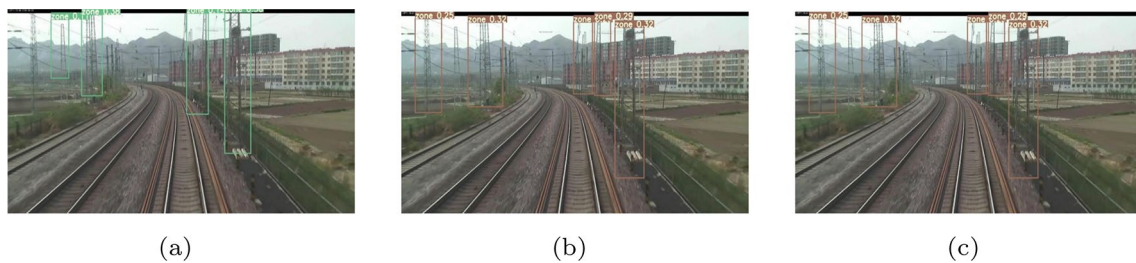


Fig. 18 Result comparison: **a** YOLOv3-spp, **b** YOLOv4-spp, **c** Faster R-CNN

Table 2 Comparison of the number of interest domain detections in the first-level network model

The first-level network model	The number of detections of interest domain	FPS
Faster R-CNN	570	5
YOLOv3-spp	485	68
YOLOv4	521	43

the scale information of the center point of the bounding box and the aspect ratio of the bounding box on the basis of GIOU-Loss, which greatly improves the prediction precision of the box regression.

The IOU curves during YOLOv3-spp and YOLOv4 training are as shown in Fig. 17.

The Fig. 17 illustrates that when the training progress is the same, the CIOU of YOLOv4 is much larger than the GIOU of YOLOv3. In addition, it provides more information about the center position and size of the interest domain, as well as better information about the regional characteristics of the interest domain.

Faster R-CNN compares the YOLO series with two-stage detection. The algorithm generates candidate regions first, generates more prediction frames than the YOLO series

network, and then classifies and regresses the candidate frames. In contrast, the one-stage algorithm of the YOLO series directly classifies and regresses the input image without generating candidate regions. Thus, Faster R-CNN has a lower error rate and missed recognition rate, but its recognition speed is slower than the one-stage algorithm.

The interest domain recognition of different models is compared as Fig. 18.

A total of 126 images of the catenary test data set along the railway were examined. The prediction results of different network models are as shown in Table 2.

The first-level network uses the Faster R-CNN detector to identify the interest domain most accurately. On 126 test set images, all interest domains on the contact web could be accurately marked and the first extreme condition of the network could be reached. It can be seen that the number of candidate frames for Faster R-CNN interest domain detection is greater than that of the YOLO series network. There is a large number of interest domains in the data set, and their shapes are standardized and easy to identify. Although the Faster R-CNN network performs better in identifying interest domains, its detection speed is slower. As the FPS of the cascade detector is only affected by the first-level detector, it is difficult to meet the conditions for real-time detection. As a first-level detector, YOLOv4 has a high degree

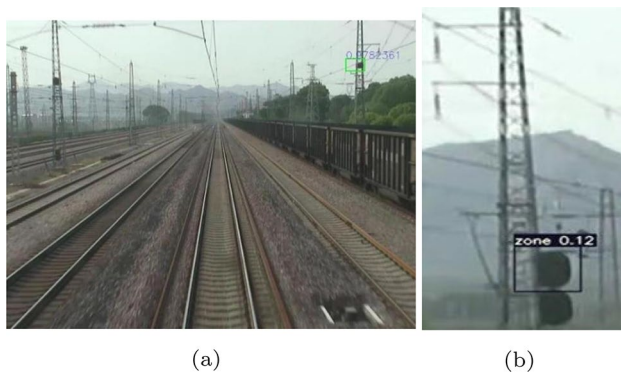


Fig. 19 Label condition: **a** missing label, **b** mislabeling

Table 3 Comparison of direct recognition results of different models

Recognition model	Recall	Precision
Faster R-CNN [28]	61.71%	83.54%
YOLOv3-spp [29]	77.48%	100%
YOLOv4 [30]	63.51%	97.24%

of precision and can meet the needs of real-time detection. Therefore, it is an ideal choice as the first-level detector.

6.4 Result comparison of catenary bird's nest detection

6.4.1 Direct detection

If the data set is small, both one-stage algorithms, such as the YOLO series, and two-stage algorithms, such as Faster R-CNN, are limited in their ability to recognize small targets. In addition, the catenary bird's nest has a single body

feature that is difficult to differentiate from environmental samples, and it is difficult to train and test.

When using Faster R-CNN with small training samples, it is difficult to learn the overall characteristics of the bird's nest sample due to its small size. There is a lack of expression ability, and it is prone to missing the label, which affects the detection rate. In comparison to the Faster R-CNN network, the YOLO series network is capable of multi-scale object recognition. The bird's nest, however, has relatively simple physical characteristics, making it easy to confuse it with environmental samples and prone to mislabeling, which reduces the detection precision. Examples of missing and incorrect labels are shown in Fig. 19.

Recent major representative models such as Faster R-CNN, YOLOv3-spp and YOLOv4 [28–30] are used for bird nest recognition, and the results are as shown in Table 3.

Based on the table, it is evident that, whether it is a two-stage network, such as the YOLO series, or a one-stage network, such as Faster R-CNN, when the bird's nest data set is small, the volume is small, and the physical characteristics are single, the network has a poor learning effect, and it is difficult to determine whether the bird nest exists. Bird's nests on catenaries along railway lines cannot be accurately identified.

6.4.2 Second-level detector training

For the second-level detector model, 126 bird's nest images were trained from the template library. The detection task of the second-level detector is to identify bird's nests in the domain of interest. The images in the data set are amplified by bilinear interpolation. Even small bird's nests can occupy a large area in the image, and it effectively eliminates the interference factors of environmental noise, and it has a very strong anti-interference capability. By enlarging the bird's nest through bilinear interpolation along with the original

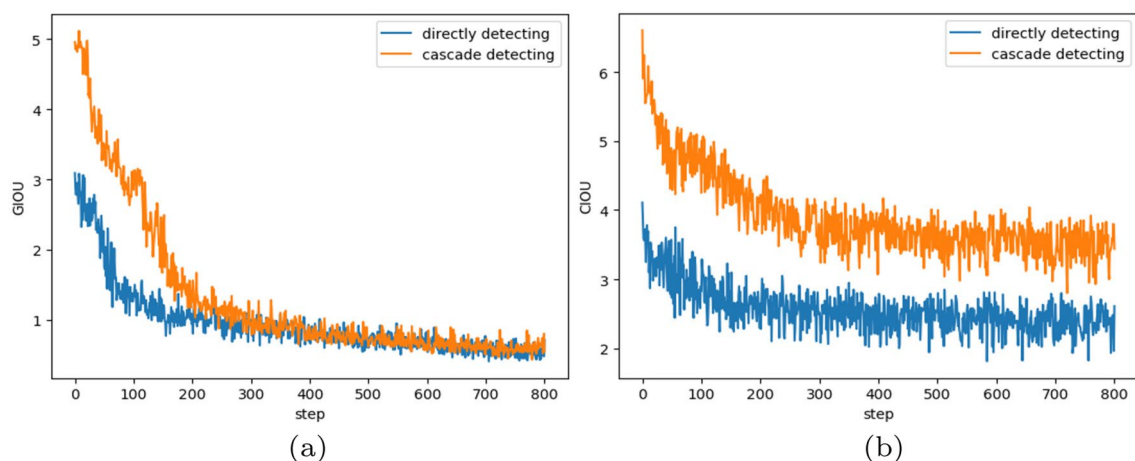


Fig. 20 IOU comparison: **a** YOLOv3-spp GIOU, **b** YOLOv4 GIOU

Table 4 Performance comparison of different cascaded models

First-level detector	Second-level detector	Recall	Precision	FPS
Faster-RCNN	Faster-RCNN	85.13%	98.95%	3.89
Faster-RCNN	YOLOv3-SPP	98.13%	95.91%	4.43
Faster-RCNN	YOLOv4	90.99%	66.23%	4.29
YOLOv3-SPP	Faster-RCNN	84.68%	99.47%	5.93
YOLOv3-SPP	YOLOv3-SPP	92.79%	96.71%	32.91
YOLOv3-SPP	YOLOv4	89.64%	75.95%	26.17
YOLOv4	Faster-RCNN	86.49%	99.48%	5.69
YOLOv4	YOLOv3-SPP	93.69%	97.65%	26.05
YOLOv4	YOLOv4	92.79%	70.55%	21.11

image, the trained IOU value has also been enhanced. The comparison of IOU values of the same detection model with the same training progress is shown in Fig. 20.

In the curve, it can be seen that the detection model under the same YOLO series has a higher IOU value for the bird's nest amplified by bilinear interpolation, and can more accurately locate the bird's nest.

A total of 210 bird's nests were detected in 126 images in the test set using different first-level detectors, testing the detection performance of second-level detector and calculating serially using both first and second-level detectors. The test results are shown in Table 4:

According to the table, the worst detection Recall of the cascade results of the CG-CFDM is 84.68%, still higher than the best result 77.48% of existing direct detection shown in Table 3.

The first level of the network induces and recognizes an interest domain a priori, and the second level recognizes the bird's nest of the object to be tested within the interest domain, and the distribution of the bird's nest within the interest domain is relatively straightforward. It can reduce the interference caused by other environmental samples, improve the discrimination of the bird's nest, and simplify the training process. The cascaded network is therefore not highly dependent on the data set of the second-level object to be measured, and only a few images are required to learn the characteristics of the distribution of the bird's nest.

The first-level detector uses Faster-RCNN to detect the largest number of interest domains, and so it can reach the extreme conditions of the cascade network, making it the detector with the highest detection rate. Despite this, Faster-RCNN has a low detection rate, and the detection rate of the overall detector is limited by the first-level detector, which cannot provide real-time detection. As a result, the expressive ability of the YOLOv4 network is greater than that of the YOLOv3, and it's able to perform well in the recognition training of interest domains where there are many samples and the features of the objects to be measured can be

Table 5 Comparison of bird nest recognition results of different models

Recognition model	Recall %	Precision %
Faster R-CNN [28]	61.71	83.54
YOLOv3-spp [29]	77.48	100
YOLOv4 [30]	63.51	97.24
HOS+HLS [31]	88.42	62.15
PWA+RWP+PF+IBF [32]	91.38	88.69
CG-CFDM(YOLOv4 + YOLOv3-spp)	93.69	97.65

distinguished easily. In training, however, the precision of performance is not high because the number of samples is small and the characteristics of the samples are not clearly evident. As shown in Table 5, the cascaded YOLOv3-SPP network of YOLOv4 offer a higher recall and precision, as well as a relatively higher frame rate, which can detect real-time video along the railway.

7 Conclusion

In this paper, a solution based on machine vision is proposed for the bird nest detection on the catenary of high-speed railway. Based on the artificial priori, the possible interest regions of the bird's nest can be found through the context-inspired reasoning algorithm, which greatly reduces the search scope of foreign object detection. In addition, the template matching algorithm is used to mark the position of the interest region in the data set to obtain the data set for in-depth learning, so as to improve the generalization ability of the interest region recognition. Then, the second-level detector is used to locate the bird's nest in the domain of interest. For a few data sets containing bird's nests, the detector cascades two one-stage detection models to explore the parallelism of detection algorithms at different stages, and realized the rapid and accurate detection of the bird nest on the catenary of high-speed railway.

Acknowledgements This work was supported in part by the National key research and development program of China, under Grant 2020YFB2103800 and National Natural Science Foundation of China under Grants U1934220, U2268203 and U1934215.

Data availability The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes

were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Liu, H.: Discussion on safety protection measures of overhead catenary. *Intell. City* **7**, 65–67 (2021)
- Huang, M.: Research on prevention and control measures of bird damage in catenary. *Technol. Innov. Appl.* **18**, 234 (2017)
- Duan, W., Tang, P., Jin, W., Wei, P.: Bird nest detection of railway catenary based on hog characteristics in key areas. *China Railway* **8**, 73–77 (2015)
- Huang, Y., Yuan, T., Yang, J.: Research on identification method of catenary fault based on SVM. *Comput. Simul.* **35**, 145–152 (2018)
- Zhu, Z., Xie, L.: Detection of birds' nest in catenary based on relative position invariance. *J. Railway Sci. Eng.* **14**, 1043–1049 (2018)
- Ge, W., Gong, T., Wang, Y., Hu, A.: Target recognition algorithm based on deep learning. *Microprocessors* **40**, 29–33 (2019)
- Li, P., Long, Y.: A method of high speed railway catenary target detection and tracking. *Electron. Eng. Product World* **28**, 49–52 (2021)
- Uijlings, Jasper RR, et al.: Selective search for object recognition. *Int. J. Comput. Vis.* **104**, 154–171 (2013)
- Liu, Z., et al.: Application of combining YOLO models and 3D GPR images in road detection and maintenance. *Remote Sens.* **13**, 1081–1081 (2021)
- Laroca, R., et al.: An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. *IET Intel. Transport Syst.* **15**, 483–503 (2021)
- Yang, Y., Li, L., Gao, S., Bai, H., and Jiang, W.: Objects detection from high resolution remote sensing imagery using an training-optimized yolov3 network. *Laser Optoelectron. Progress* 1–12 (2021)
- Zhao, J. et al.: Detection of passenger flow on and off buses based on video images and YOLO algorithm. *Multimed. Tools Appl.* 1–24 (2021)
- Zhou, H., Zhu, G., Zhang, Y., Ren, S.: Image classification based on region of interest detection and spatial pyramid matching. *Comput. Eng. Appl.* **54**, 206–211 (2018)
- Chen, C., Wang, H., Zhao, Y., Wang, Y., Li, L., Li, K., Zhang, T.: A novel traffic sign recognition algorithm based on deep learning. *Telecommun. Eng.* **61**, 76–82 (2021)
- Yao, G., Zhang, Z., Li, X., Zhang, J.: Vehicle color recognition under road monitoring system based on convolution neural network. *Technol. Innov. Appl.* **8**, 86–89 (2021)
- Han, H., Chi, F.: Application of convolution neural network in road crack detection. *Technol. Innov. Appl.* **5**, 176–178 (2021)
- Ye, H., et al.: Deep learning-based visual ensemble method for high-speed railway catenary clevis fracture detection. *Neurocomputing* **396**, 556–568 (2020)
- Ye, H., et al.: Computer vision-based automatic rod-insulator defect detection in high-speed railway catenary system. *Int. J. Adv. Rob. Syst.* **15**, 3 (2018)
- Lin, S., et al.: LiDAR point cloud recognition of overhead catenary system with deep learning. *Sensors* **20**, 8 (2020)
- Ren, S.Q., He, K.M., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1137–1149 (2017)
- He, D., Jiang, Z., Chen, K., Yang, Y., Yao, X.: Research on detection of bird nests in overhead catenary based on deep convolutional neural network. *Electric Drive Locomotives* **4**, 126–130 (2019)
- Wang, J., Luo, H., Yu, P., Liu, Y.: Detection of bird's nest in overhead catenary system images for railway based on faster r-cnn. *Railway Locomotive Car* **40**, 78–81 (2020)
- Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *Comput. Sci.* **1409**, 1–8 (2014)
- Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet classification with deep convolutional neural networks. *Int. Conf. Neural Inf. Proc. Syst.* 1097–1105 (2012)
- Szegedy, C., Liu, W., Jia, Y., et al.: Going deeper with convolutions. *IEEE Conf. Comput. Vis. Pattern Recogn.* **2015**, 1–9 (2015)
- He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. *IEEE Conf. Comput. Vis. Pattern Recogn.* **2016**, 770–778 (2016)
- Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **59**, 167–181 (2004)
- Li, F., Xin, J., Chen, T., et al.: An automatic detection method of bird's nest on transmission line tower based on faster_RCNN. *IEEE Access* **8**, 164214–164221 (2020)
- Liu, B., Huang, J., Lin, S., Yang, Y., Qi, Y.: Improved YOLOX-S abnormal condition detection for power transmission line corridors. In *Proceedings of the 2021 IEEE 3rd International Conference on Power Data Science (ICPDS)*, 13–16 (2021)
- Yang, Z., Xu, X., Wang, K., et al.: Multitarget detection of transmission lines based on DANet and YOLOv4. *Sci. Programm.* 1–12 (2021)
- Wu, X., Yuan, P., Peng, Q., et al.: Detection of bird nests in overhead catenary system images for high speed rail. *Pattern Recogn.* **51**, 242–254 (2016)
- Lu, J., Xu, X.Y., Xin, L., et al.: Detection of bird's nest in high power lines in the vicinity of remote campus based on combination features and cascade classifier. *IEEE Access* **6**, 39063–39071 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.