Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 7534739, 12 pages
https://doi.org/10.1155/2021/7534739

Research Article

YOLO-SD: A Real-Time Crew Safety Detection and Early Warning Approach

Xinwei Lin,[1,2] Shengzheng Wang,[2] Zhen Sun,[2] and Min Zhang[2]

[1]*College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China*
[2]*Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China*

Correspondence should be addressed to Shengzheng Wang; szwang@shmtu.edu.cn

Received 7 September 2021; Accepted 3 December 2021; Published 21 December 2021

Academic Editor: Eneko Osaba

Wearing safety rope while working at the loft and over the side of a ship is an effective means to protect seafarers from accidents. However, there are no active and effective monitoring methods on ships to control this issue. In this article, a one-stage system is proposed to automatically monitor whether the crew is wearing safety ropes. When the system detects that a crew enters the work area without a safety rope, it will warn the supervisor. In this regard, a safety rope wearing detection dataset is established. Then a data augmentation algorithm and a boundary loss function are designed to improve the training effect and the convergence speed. Furthermore, features from different scales are extracted to get the final detection results. The obtained results demonstrate that the proposed approach YOLO-SD is effective at different on-site conditions and can achieve high precision (97.4%), recall rate (91.4%), and mAP (91.5%) while ensuring real-time performance (38.31 FPS on average).

1. Introduction

In order to minimize the human injury and loss of life at sea, the International Maritime Organization (IMO) has adopted the International Safety Management (ISM Code) as a mandatory standard and requires vessel operators to develop, implement, and maintain their Safety Management System (SMS). Generally, it requires that when working aloft, the crew must obtain the captain's permission, and a checklist, two or more staff, and a task manager must be assigned at the same time. However, it cannot guarantee that all the crew can implement these rules. Issued statistics by the European Maritime Safety Agency (EMSA) [1] indicate that, from 2014 to 2019, maritime accidents across Europe caused more than 496 deaths and 6,210 injuries, of which slips, trips, and falls share almost 37% of all accidents. More specifically, during a ship drilling in November 2019, a Japanese cargo ship ORANGE PHOENIX crew died after falling to the deck without a safety rope. Studies of the EMSA show that many casualties at offshore tasks originate from not wearing safety ropes by the crew. Currently, monitoring the safety rope wearing is one of the workers' mutual inspections before the operation [2]. So far, no automatic method has been proposed in this regard so that it is a challenge for the captain or shipping company manager to ensure whether the crew is wearing safety ropes when working at high altitudes. Accordingly, it is of significant importance to establish a method to automatically determine whether the crew is wearing a safety rope when entering the work area.

The crew working at height refers to working on ship masts (see Figure 1(a)), outside chimneys (see Figure 1(b)), or ship crane (see Figure 1(c)). Unlike ordinary construction sites, there are a few crews onboard, which mainly originates from the space limitation. Accordingly, the working crews lack adequate supervision and protection during the work. Moreover, the crew works in a wide range of vertical space on the ship, which requires safety rope detection algorithms, multiscale detection capabilities, and high robustness. On the other hand, conventional target detection algorithms often rely on cloud platforms with substantial computing power or high-performance GPU computer clusters to achieve better detection rates. However, considering onboard requirements, it is a challenge to apply these

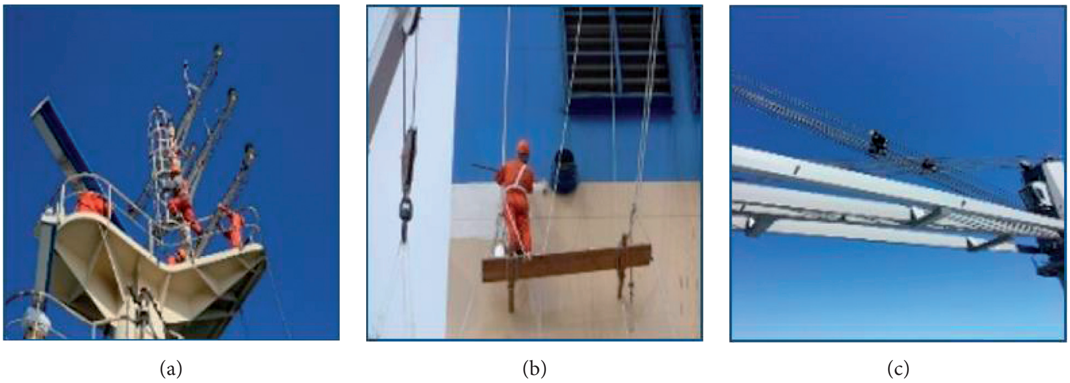


Figure 1: Some offshore tasks at high altitudes.

algorithms. Generally, safety detection and early warning systems for the crews working at heights should have the following capabilities: (1) determine whether all individuals in the working area are wearing a safety rope; (2) detect the surveillance video in real time; (3) issue a warning message when the crew enters the working area without wearing a safety rope. Currently, investigations about safety ropes are mainly focused on the use of visual perception technology [3–6] and signal processing [7, 8] to perform a qualitative or quantitative analysis of the reliability of safety rope. Particularly, detecting the wearing of safety rope on the ship has not yet been carried out.

In the present study, it is intended to specifically address the task of safety rope-wearing detection on the ship. The main objective of this article is to identify whether all crews on the ship's high-altitude operation area wear a safety rope or not; otherwise, a warning message should be issued on the monitoring screen. The difficulty of the above-mentioned tasks makes it problematic to rely on any multistage method with handicraft features. In order to resolve this problem, a convolutional neural network (CNN) is introduced. The proposed method is inspired by the development of a one-stage detector YOLO (You Only Look Once) [9], which implements the object by directly regressing bounding boxes through a single CNN. Compared with conventional target detection methods, the established method can automatically perform feature learning and provide excellent performance in the field of computer vision. The main contributions of this article can be summarized as follows: (1) It is the first method that introduces CNN into the crew safety detection and early warning in surveillance video. The proposed system is end-to-end trainable. (2) Based on the YOLOv5 pipeline and BN-Conv modules, a multiscale CNN framework YOLO-SD is proposed to improve the accuracy and speed of the detection under complex surveillance conditions. (3) Based on the proposed method, Crew Safety Rope Wearing Detection Dataset is established, which contains 3150 images and 6,583 safety ropes instances. These images cover massive changes in diverse scenes and scales as well as examples with occlusion. Each instance in the benchmark is annotated with a class label with a bounding box.

The contents of this article are organized as follows: The literature survey is presented in Section 2, followed by the system overview in Section 3. Then details of the proposed Crew Safety Rope Detection algorithm are discussed in Section 4. The experimental results are presented in Section 5. Finally, conclusions and main achievements are summarized in Section 6.

2. Literature Survey

Aiming at detecting the target precisely, detection and classification with precise target positioning are combined to provide a proper understanding of the image. In conventional target detection techniques, artificial features are mainly applied to extract and shallow trainable structures for target detection. Consequently, these methods may simply fail in complicated scenes, severe climate, and different occlusions.

With the emergence of deep learning technology [10–13], many limitations of conventional target detection technologies have been resolved so that deeper semantics and features can be learned. The deep learning-based target detector has three main modules, including information area selection, feature extraction, and classification. In the information area selection process, a multiscale sliding window is usually applied to scan the entire image and calculate the height process. Moreover, the target recognition process is mainly carried out in the feature extraction process. It is based on visual feature extraction to characterize the semantics of the target. It is worth noting that common features in this regard are SIFT [14], HOG [15], and Haar-like [16]. Finally, classification is defined as the process of distinguishing the target object from all other categories. Some common effective classifiers are AdaBoost [17], support vector machine (SVM) [18], and deformable component model (DPM) [19]. Based on these algorithms, modern target detectors can be mainly divided into two categories: (1) target detectors based on the area proposal such as R–CNN [20], Fast R–CNN [21], Faster R–CNN [22]; (2) detectors based on the target detector for classification/regression such as YOLO [10, 23–25], SSD [26], and EfficientDet [27]. In the former detectors, the area proposal should be initially generated and then the proposal area should be classified. This two-level detector based on the region suggestion extracts the candidate area, which is a complex and time-consuming process to achieve real-time detection. On the other hand, the target detector based on classification/regression has only a single forward CNN network, eliminating the region suggestion generation and subsequent feature resampling. Accordingly, it encapsulates all calculations in a network, thereby making it easier to realize a real-time detection.

Redmon used mapping and direct evaluation of class probabilities to predict candidate frames and proposed a one-stage object detector YOLO [9]. The main idea of this detector was to divide the captured image into specific grid units, where each grid unit is responsible for predicting the candidate anchor and the corresponding confidence value. Further investigations reveal that this method can process 45 frames of images per second in real time, which is much higher than that of other target detectors. However, YOLO cannot effectively deal with objects with unusual aspect ratios or small objects. In order to resolve these shortcomings, Liu used the anchor, RPN, and multiscale representations and proposed a single-shot multibox detector (SSD) [26]. Moreover, Redmon improved the original YOLO and proposed YOLO9000 [23] and YOLO v3 [24] and achieved good results in detecting high-resolution or multiscale targets. Bochkovskiy [25] improved Redmon's YOLO framework and proposed YOLO v4 and YOLO v5. Accordingly, a 43.5% average accuracy and 140 fps were achieved on the MS-COCO data set.

3. System Overview

Figure 2 shows the framework of the proposed algorithm YOLO-SD, indicating that the proposed framework is mainly composed of three networks.

(1) Data Augmentation Network: In the training stage, a data augmentation network is designed to augment the dataset and increase the number of multiscale safety ropes and the number of target-occluded images artificially, to improve the network's ability to detect different scale safety ropes and improve the algorithm robustness.

(2) Feature Extraction Network: Based on the YOLO v5s framework, the feature pyramid network (FPN) and CSPDarknet53 network are employed to extract the image features of different scales. In this regard, the network's main structure involves a bottom-up pathway, a top-down pathway, and lateral connections, which can extract three different resolutions and semantic information and obtain the image features of different scales.

(3) Prediction network: It processes the acquired multiscale image features and generates detection boxes and confidence values for crew members wearing safety ropes and those not wearing safety ropes.

When the prediction network detects a crew member entering the work area without wearing a safety rope, the system marks it out and issues a sound warning to remind the supervisor.

4. Detection Approach

4.1. Data Augmentation Network. During the experiment, it was found that the majority of missing targets were relatively small targets. This phenomenon may be attributed to the following two reasons: (1) The area of the safety rope in the image is relatively small. (2) The safety rope can be easily blocked by other objects. Kisantal [28] showed that the algorithm's detection performance for small targets could be improved through the oversampling and copy-pasting data enhancement methods for images containing targets. Accordingly, a data augmentation strategy is designed to train the algorithm and overcome the foregoing problems. To this end, each image is initially oversampled and then new training images are generated by flipping, random scaling, random cropping, and random arrangement. Figure 3 shows that the bottom image, which is used for the training, is generated by randomly cropping, arbitrarily scaling, flipping, and rearranging the four top images.

4.2. Feature Extraction Network. The extraction of safety rope features is the key to detect and identify the safety rope. In this regard, the Darknet53 network is employed to extract safety rope features, which is based on the Darknet-19 and residual network of YOLO v2. The network is deepened to layer 53 and then shortcut connections are added between layers. It should be indicated that as the deeper network structure deepens, more advanced features of the safety rope can be extracted. However, the corresponding network calculation and the computational complexity dramatically increase.

The Darknet53 network introduces Cross-Stage Partial Networks (CSPNet) [29] to achieve a more decadent combination of gradients while reducing the calculation. In particular, CSPNet divides the feature map of the Darknet53 network base layer into two parts, where the first part passes through a dense block and the other part passes a transition layer and then merges them through a cross-stage hierarchical structure. The main idea is to propagate the gradient flow through different network paths by separating the upper layer of gradient flow. Accordingly, the CSPNet can reduce the number of inference calculations by about 20%. Meanwhile, the memory usage during the feature pyramid generation process can be reduced by up to 75%. The feature extraction network of YOLO v5s uses CSPDarknet53, and its network structure is shown in Figure 4. It should be indicated that CSPDarknet53 is based on the idea of Cross-Stage Partial Networks (CSPNet) to transform the Darknet53 network.

Since the safety rope may have different sizes in different images, the feature extraction network is required to perform the multiscale feature extraction process. In this regard, Feature Pyramid Network (FPN) [30], which is composed of the bottom-up pathway, top-down pathway, and lateral connections, can be applied to generate a high resolution containing low-level safety rope contour features with less semantic information and low resolutions containing high-level safety rope texture feature with rich semantic

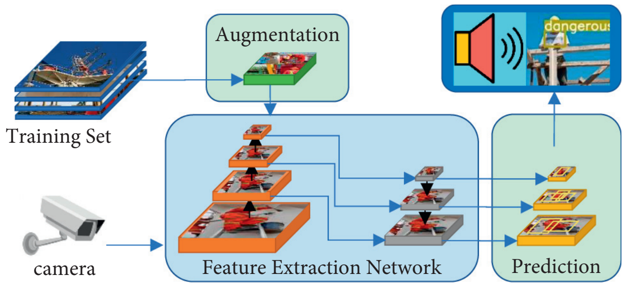


Figure 2: Frame diagram of the YOLO-SD system.

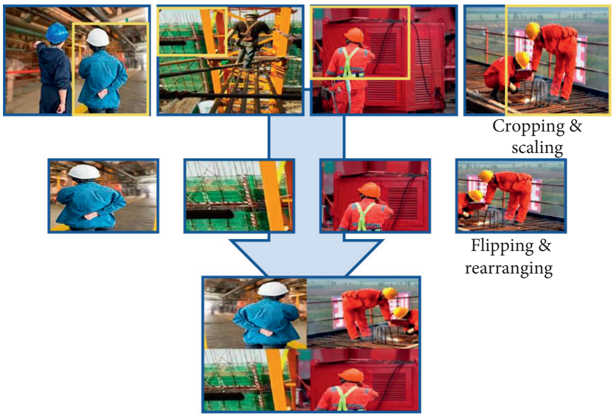


Figure 3: Sample data augmentation.

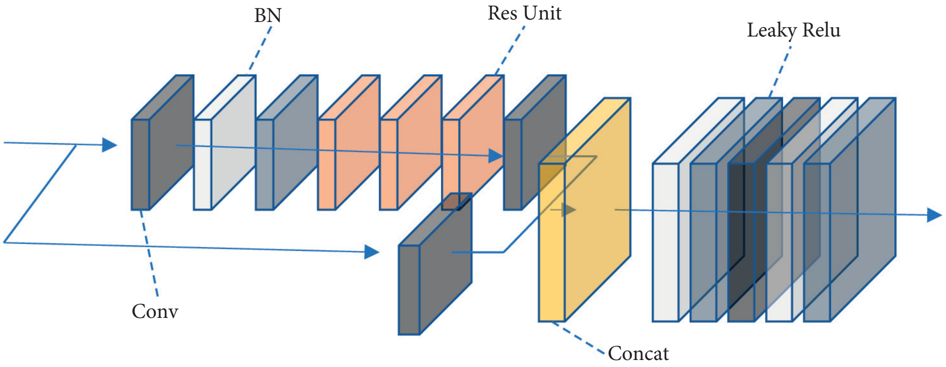


Figure 4: CSPDarknet53 network structure.

information. The bottom-up pathway is the feedforward calculation of the backbone network, which calculates the feature level composed of feature maps of multiple scales with a zoom step of 2. The top-down pathway generates higher resolution features by upsampling spatially coarser but semantically more robust feature maps from higher pyramid levels, enhanced by horizontal connections. Each horizontal connection incorporates feature maps of the same size in the bottom-up path and the top-down path. For example, Figure 5 shows that a $608 \times 608 \times 3$ image goes through a bottom-up path to obtain three different scale feature maps of 76×76, 38×38, and 19×19. The top-down path three different scale feature maps are 19×19, 38×38, and 76×76, and the last three feature maps used for prediction are $76 \times 76 \times 255$, $38 \times 38 \times 255$, and $19 \times 19 \times 255$.

The cross-stage partial network (CSPNet) reduces the number of network calculations. However, the existence of a large number of CBL modules in the entire feature extraction network still increases the computational expense. In particular, there are 48 CBL modules in the YOLO v5s network structure. Meanwhile, each CBL module consists of a convolutional layer, a batch normalization (BN) layer, and an activation function (Leaky Relu). The BN layer can accelerate network convergence in the network training process. Moreover, it also increases the computational expense in the network forward inference process and affects the model performance. Based on the CBL module, the BN layer is merged into the convolution layer to form a new BN-Conv layer to reduce the calculation of forwarding inference of the model.

The BN layer is between the convolution layer and the activation function Leaky Relu. It converts each small batch of data into a standard Gaussian distribution with the zero mean and the variance of 1. This can be mathematically expressed in the form below:

$$\widehat{x}_i = \frac{x_i - 1/m\sum_{i=1}^{m}(x_i)}{\sqrt{1/m\sum_{i=1}^{m}\left(x_i - 1/m\sum_{i=1}^{m}(x_i)\right)^2 + \varepsilon}}. \tag{1}$$

Moreover, the BN layer can be expressed as follows:

$$BN_{\gamma\,,\,\beta}(x_i) = \gamma \cdot \widehat{x}_i + \beta, \tag{2}$$

where ε, γ, and β are the regularization parameter, scaling factor, and the offset term, respectively.

Then the convolutional layer can be written in the form below:

$$x_{\text{conv}} = \sum_{i}^{n}(x_i * w_i), \tag{3}$$

where w_i denotes the weight of the i-th layer.

The BN-Conv layer is formed by merging the BN layer into the convolutional layer, where the weight parameter w_i' and bias term β' can be calculated from the following expressions:

$$\begin{aligned} w_i' &= \frac{\gamma * w_i}{\sqrt{1/m\sum_{i=1}^{m}\left(x_i - 1/m\sum_{i=1}^{m}(x_i)\right)^2 + \varepsilon}}, \\ \beta' &= \beta - \frac{\gamma * \widehat{E}(x_{\text{conv}})}{\sqrt{1/m\sum_{i=1}^{m}\left(x_i - 1/m\sum_{i=1}^{m}(x_i)\right)^2 + \varepsilon}}. \end{aligned} \tag{4}$$

Finally, the BN-Conv layer can be obtained from the following expression:

$$x_{i+1} = \sum_{i=0}^{n}(x_i * w_i') + \beta'. \tag{5}$$

4.3. Prediction Network. In the safety rope prediction network, three different scale feature maps of the feature extraction network are applied to generate prediction boxes and the corresponding category predictions. Accordingly, a unique prediction box can be obtained through the non-maximum suppression (NMS).

Furthermore, the loss function is an essential part of the model training process and the prediction network. It affects the convergence speed and the prediction error of the model. The loss function of YOLO v5s consists of three parts, including classification loss, confidence loss, and bounding box loss. It is worth noting that the bounding box loss function is a Generalized Intersection over Union (GIoU) [31], which is an evolutionary version of Intersection over Union (IoU).

IoU is the most commonly used indicator in target detection tasks. In anchor-based methods, the main role of IoU is to determine the positive and negative samples of the safety rope and evaluate the output box's distance to the ground-truth. IoU can be mathematically expressed in the form below:

$$IoU = \frac{|A \cap B|}{|A \cup B|}, \tag{6}$$

where A and B are the blue target box and yellow prediction box in Figure 6, respectively. Moreover, $A \cap B$ and $A \bigcup B$ denote the intersection and union of A and B, respectively. Equation (6) indicates that the definition of the IoU loss function is relatively simple. Further investigations reveal that this definition has two shortcomings: (1) when the target box and the prediction box do not intersect, equation (6) results in IoU = 0. However, this result cannot reflect the distance between the two boxes so that the loss function cannot be derived and optimized. (2) When the target boxes and the prediction boxes have the same size separately, as long as $A \cap B$ is constant, the corresponding IoU value is the same regardless of A and B relative position. Meanwhile, the IoU loss function cannot distinguish the intersection of the two boxes.

In order to resolve these two shortcomings, the GIoU loss function has been introduced to the YOLO v5s.

$$\text{GIoU} = \text{IoU} - \frac{\left|A_C - A \cup B\right|}{\left|A_C\right|}, \tag{7}$$

where A_c is the minimum area of the two boxes' enclosed area (the red box in Figure 7). Unlike IoU, which only considers overlapping areas, all areas are considered in GIoU. Accordingly, GIoU can better reflect the overlap degree between the two boxes. It is worth noting that when the prediction box is inside the target box, A_c and $A \bigcup B$ are equal, and GIoU reduces into IoU, and their relative positional relationship cannot be distinguished.

Although the foregoing shortcomings of the IoU loss function are resolved in the GIoU function, some other problems, including slow convergence and inaccurate regression [32], should be resolved. Accordingly, the Distance-IoU (DIoU) loss function is proposed in the present article to replace the GIoU loss function and resolve these problems.

$$\text{DIoU} = \text{IoU} - \frac{\rho^2\left(b, b^{\text{gt}}\right)}{c^2}, \tag{8}$$

where b and b^{gt} are the center points of the prediction box and the target box, respectively. Moreover, ρ and c denote the Euclidean and the diagonal distance of the minimum closure area A_c, respectively. In the DIoU function, the overlap area and the center point distance are considered simultaneously. When the target box wraps the prediction box, the distance between the two boxes is directly measured, so the DIoU loss function converges faster. In this regard, the black arrow in Figure 8 represents the diagonal distance c of the minimum closure area A_c, and the red arrow

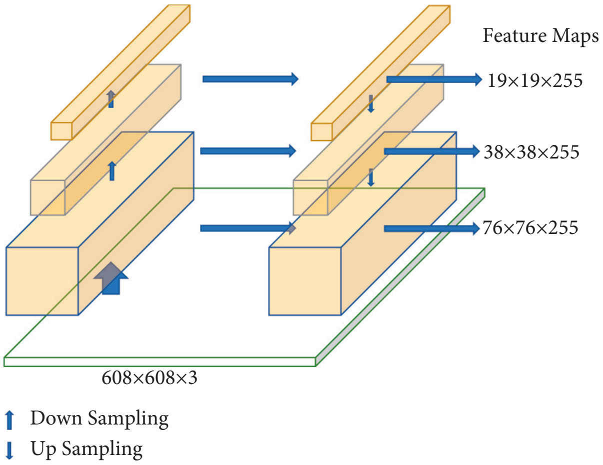


Figure 5: Feature pyramid network.

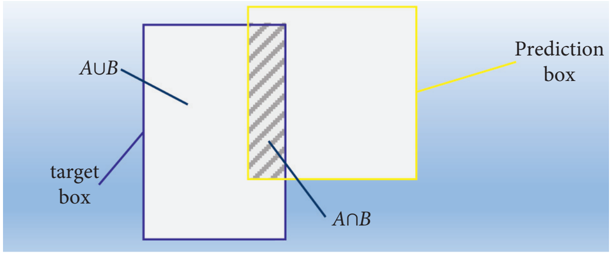


Figure 6: Schematic diagram of IoU.

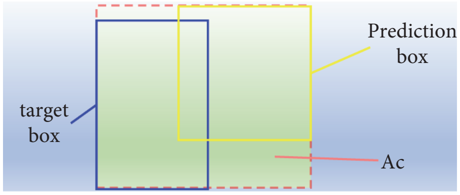


Figure 7: Schematic diagram of GIoU.

represents the Euclidean distance between the center point of the prediction box and the center point of the target box $d = \rho(b, b^{gt})$.

In summary, the overlapping area and the distance between the center points of the two boxes are considered in the proposed DIoU loss function. This loss function is expected to improve the accuracy of the prediction box's regression and increase the convergence rate.

5. Experimental Result

5.1. Dataset. The training and testing result of the algorithm depends on the data set. However, no public crew safety rope data set is available for use. Therefore, a self-made crew safety rope data set is created in this study. In this regard, two data sources, including the surveillance video onboard and web crawlers, have been prepared prior to the test. In total, 3,150 images and 6,583 targets are gathered in the dataset. After extracting images from the ship's surveillance video, some images including crew are screened out and labelled. Then the labelled targets are divided into two categories, including safety harness with 1,250 images and dangerous cases with 1,900 pictures. During training the algorithm, the data set is divided into the training set, the validation set, and the test set at the ratio of 16 : 4:5. Figure 9 shows some images of the prepared dataset.

5.2. Evaluation Indicators and Experimental Platform. The performance of the algorithm depends on the objective evaluation indicators. In the present article, precision, recall, mean average precision (mAP), and frame rate are considered as the primary performance evaluation indicators.

Precision and recall are used to evaluate the algorithm's accuracy in classifying the target and its capability to find the target. These indicators are defined in equations (9) and (10), respectively. The distribution of the precision-recall rate (P-R) can be drawn by taking the recall rate as the X-axis and the precision as the Y-axis. Average precision (AP) is applied to calculate the area under the precision-recall (P-R) rate curve of a specific category. Moreover, the mean average precision (mAP) is used to calculate the average of the area under the P-R curve of all categories. It should be indicated that mAP reflects the average detection accuracy of all categories and can be used to evaluate the performance of the proposed DIoU loss function.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{9}$$

$$recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{10}$$

where TP (True Positive) refers to the number of positive classes predicted as positive, FP (False Positive) refers to the number of false positives that predict a negative class as a positive class, and FN (False Negative) refers to the number of false negatives that predict a positive class as a negative class.

All simulations are carried out on a personal PC with an Intel(R) Core(R) 7 @ 3.60 GHz CPU, 16G running memory, NVIDIA Geforce GTX1070 graphics card, CUDA v10.1 software, Window7 64-bit operating system, and Pytorch 1.7 deep learning framework.

5.3. Analysis of Experimental Results. In order to evaluate the advantages of the proposed algorithm, obtained results are compared with those of the standard YOLO v5s, SSD, and EfficientDet algorithms on the same platform.

5.3.1. Training Results. Figure 10 shows sample results of the algorithm training process. It is observed that, after 250 iterations, the average loss value remains constant and approaches an asymptotic value, indicating rapid convergence of the algorithm. The precision-recall rate (P-R) curve in Figure 10(b) shows that, for the recall rate of 96%, the accuracy can reach 90%. Moreover, Figure 10(c) shows the falling curve of bounding box loss of YOLO v5s and

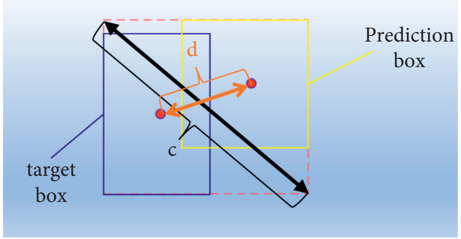


Figure 8: DIoU schematic diagram.

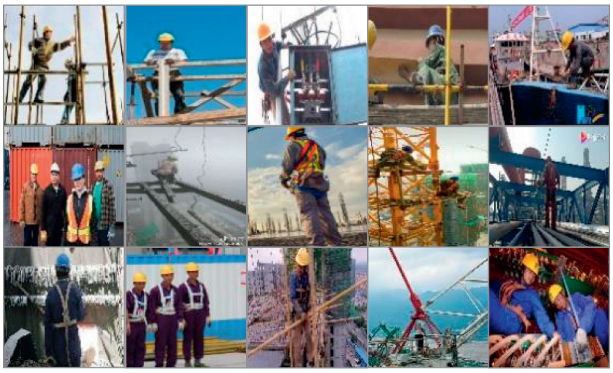


Figure 9: Some images of the prepared data set.

improved YOLO v5s. It is observed that the modified algorithm outperforms the bounding box loss from the aspects of convergence rate and regression accuracy.

5.3.2. Comparison with Standard Algorithms. In order to verify the detection effect of the proposed algorithm, the detection results obtained from the standard YOLO v5s and the proposed YOLO-SD algorithms are compared. Table 1 shows that proposed YOLO-SD algorithm improves the accuracy, recall, and average accuracy. More specifically, the detection speed is increased by 7.3%.

5.3.3. Comparisons with Other Algorithms. To further evaluate the performance of the proposed algorithm, the detection efficiencies of YOLO-SD, SSD, and EfficientDet [27] algorithms are compared. In this regard, the test results are presented in Table 2. It is observed that the SSD algorithm performs well, where the mAP value reaches 89.3%, but the detection frame rate in the SSD algorithm is only one-third that of the proposed YOLO-SD algorithm. Meanwhile, it is found that the EfficientDet-d0 algorithm has the highest detection speed, up to 59.30 fps, while that of the proposed YOLO-SD algorithm is 38.31 fps. However, mAP value of the YOLO-SD algorithm is much higher than that of the EfficientDet-d0 algorithm. It should be indicated that the EfficientDet-d1 algorithm has a higher mAP value than the EfficientDet-d0 algorithm, but there is still a remarkable gap with that of the proposed YOLO-SD algorithm. Moreover, it is found that the detection speed of EfficientDet is lower than that of the proposed one. Since the proposed YOLO-SD algorithm considers both the detection precision and detection speed, it can better complete the crew safety rope's detection task.

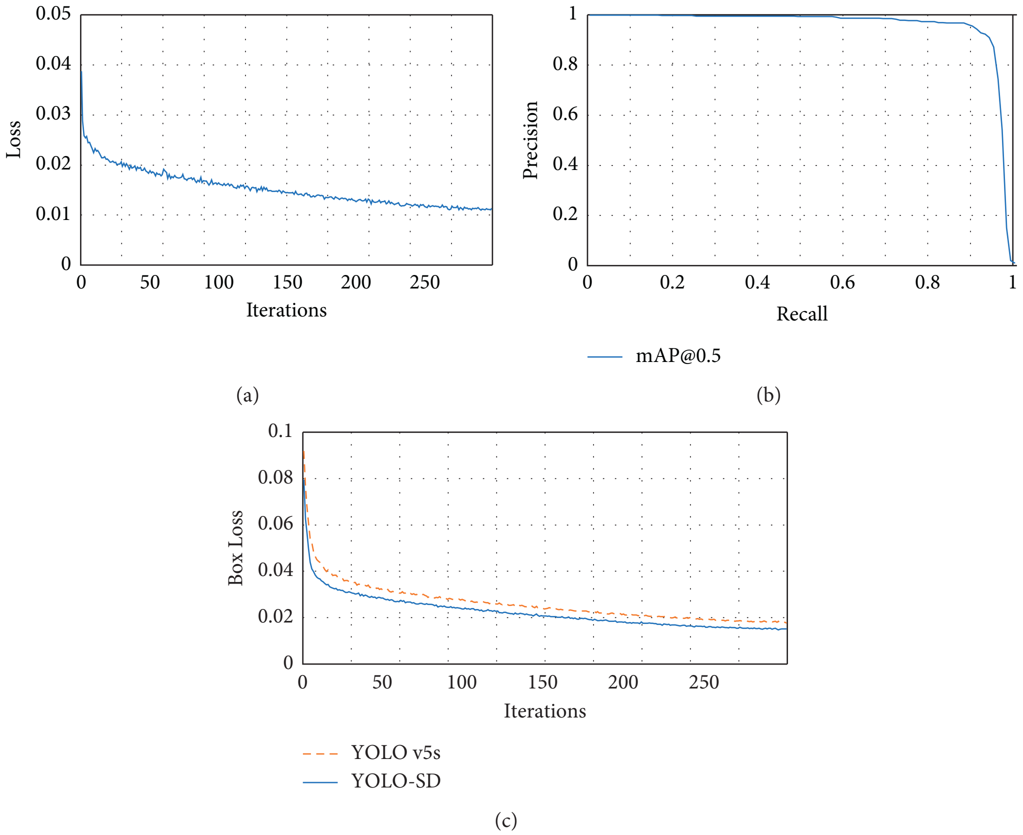
Figure 11 shows some of the crew safety detection results. From left to right, the crew's distance and the camera become farther and the crew's imaging size gradually decreases. It should be indicated that each image has a different resolution. It is found that the SSD algorithm has one misdetection and two fraudulent detections. Moreover, the EfficientDet-d1 algorithm has four missing detections. Meanwhile, the proposed algorithm has only one misdetection. Table 3 indicates that the average confidence values of the SSD, EfficientDet-d1, and YOLO-SD algorithms are 78.33, 47.13, and 78.63, respectively. It is concluded that the proposed algorithm has the highest average confidence value and has the most concentrated distribution of confidence values. Meanwhile, it has the best detection stability in different image resolutions and target sizes and has a good detection effect. Accordingly, it is inferred that the proposed algorithm has good robustness.

5.3.4. Surveillance Video Detection Test. Figure 12 shows sample results of the surveillance video detection process. It is observed that the proposed YOLO-SD algorithm achieves good results in the detection process. Although the crew's back safety rope was partially obscured by the iron bar at certain moments (see Figure 12 Frame 350), YOLO-SD algorithm can still detect the safety rope.

5.3.5. Anti-Interference Test. To evaluate the algorithm's detection stability, tests are conducted on images taken at different working conditions such as low-light, night, rainy, and foggy days. The obtained results are presented in Figure 13. To quantitatively analyse the anti-image pollution ability of the algorithm, Perlin Noise [33] is added to test images. It is worth noting that applying Perlin Noise is a widely adopted method to generate specific texture noises or augmentation data sets [34, 35]. Figure 14 reveals that the test set images are polluted by 10%, 20%, 30%, 40%, 50%, and 60% coverage rates of Perlin Noise. The test results are shown in Table 4. It is found that when 40% noise is added to the original image, the decline of recall and mAP is less than 6%. Moreover, when the noise coverage increases to 50% and 60%, the corresponding drop of recall and mAP increases to 21% and 35%, respectively. However, the detection precision remains at a high level.

5.3.6. Detection Demo Platform. Figure 15 shows the detection demo software based on the pyQt5 framework. A surveillance camera is used at the port to test the software's functions. When the start button is pressed, the program detects the crew in the surveillance video, marks the detected crew with a mark box, and records the detection information in the log box. Then when a detected crew with no safety rope enters the surveillance area, an alarm will be issued to notify the supervisor.

5.4. Discussion. An automatic system for identifying non-safety-ropes-use provides an effective means to reduce the risk of falling and improve safety issues in a ship's high-



(a)

(b)

(c)

Figure 10: Training results. (a) Train loss. (b) P-R curve. (c) Train box loss.

Table 1: Comparison of detection results

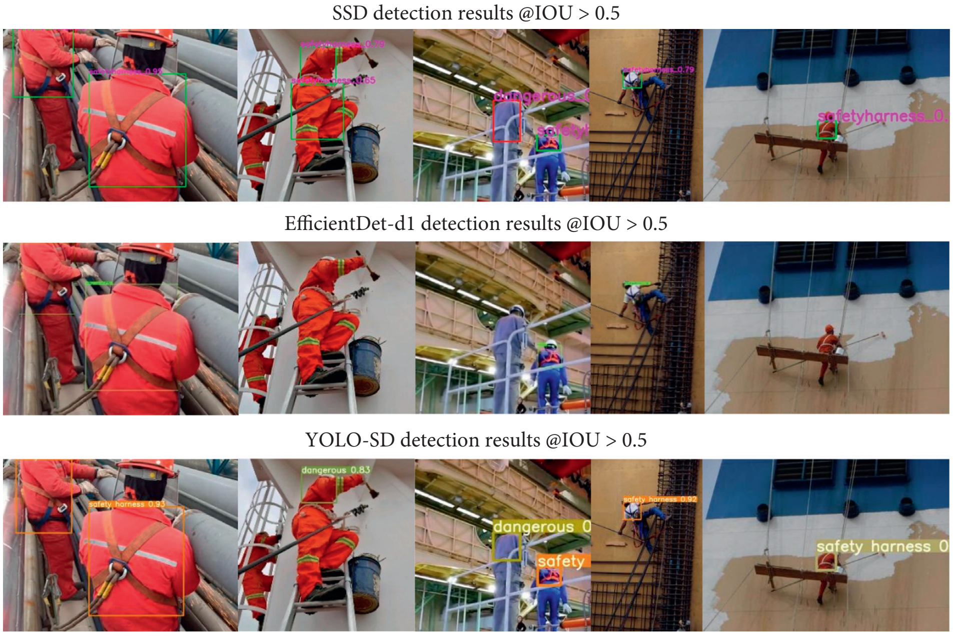
Method	Precision (P) (%)	Recall (R) (%)	mAP@0.5 (%)	Speed/fp (s)
YOLO v5s	97.0	89.1	89.1	35.71
YOLO-SD	**97.4**	**91.4**	**91.5**	**38.31**

Table 2: Obtained results from different algorithms.

Method	Precision (P) (%)	mAP@0.5 (%)	Speed (fps)	Weights (MB)
SSD	89.5	89.3	9.29	91.1
EfficientDet-d0	/	15.2	**59.30**	15.0
EfficientDet-d1	/	48.0	26.93	25.6
YOLO-SD	**97.4**	**91.5**	38.31	**14.1**

altitude operations. The main objective of the present article is to employ the target recognition method to detect whether the safety rope is worn or not. In this regard, extensive experiments on the self-built dataset have been carried out, which demonstrate the effectiveness of the proposed method.

It is found that CNN-based safety ropes wearing detection methods are reliable and stable in a wide range of on-site conditions such as visual range, individual posture, and occlusions. Through a series of improvements to the original algorithm, the YOLO-SD algorithm has improved the mAP (91.5% vs. 89.1%). To apply the new BN-Conv modules, the network inference speed increases by 7.3% (38.31 fps vs. 35.71 fps). Experimental results show that, for the same training and testing dataset, the proposed YOLO-SD algorithm outperforms the SSD method by 8.8% (97.4% vs. 89.5%) in the detection precision and 312.4% (38.31 fps vs. 9.29 fps) in the detection speed. Moreover, the SSD model has a larger weight file and a larger parameter amount so that it requires more memory space and more powerful equipment. Further investigations show that the proposed YOLO-SD algorithm is slower than the EfficientDet-d0 algorithm (38.31 fps vs. 59.30 fps) and faster than the EfficientDet-d1 algorithm (38.31 fps vs. 26.93 fps). However, the mAP value of the YOLO-SD algorithm is significantly higher than that of EfficientDet-d1 (91.5% vs. 48.0%) and EfficientDet-d0 (91.5% vs. 15.2%) algorithms.



SSD detection results @IOU > 0.5

EfficientDet-d1 detection results @IOU > 0.5

YOLO-SD detection results @IOU > 0.5

(a) Large targets

(b) Large targets with occlusion

(c) Medium targets

(d) Small target

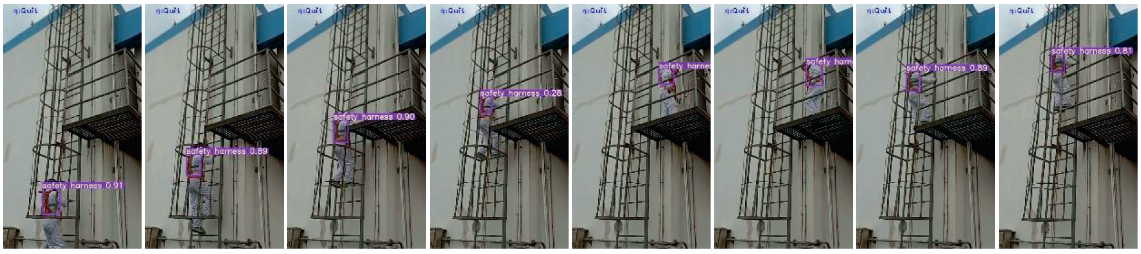
(e) Small target with occlusion

Figure 11: Partial test results.

Table 3: Target confidence value in Figure 11.

Method	Figure 11(a)		Figure 11(b)		Figure 11(c)		Figure 11(d)	Figure 11(e)
	Left	Right	Left	Right	Left	Right		
SSD	**0.98**	0.95	/	**0.79** **0.65**	**0.97**	**0.99**	0.79	**0.93**
EfficientDet-d1	0.85	**0.99**	/	/	/	0.95	**0.98**	/
YOLO-SD	0.84	0.93	/	**0.83**	0.90	0.94	0.92	**0.93**

Note. The SSD algorithm incorrectly detects the crew on the right side of Figure 11(b), and the crew has two wrong classification boxes.



0.91 @Frame50

0.89 @Frame150

0.90 @Frame250

0.28 @Frame350

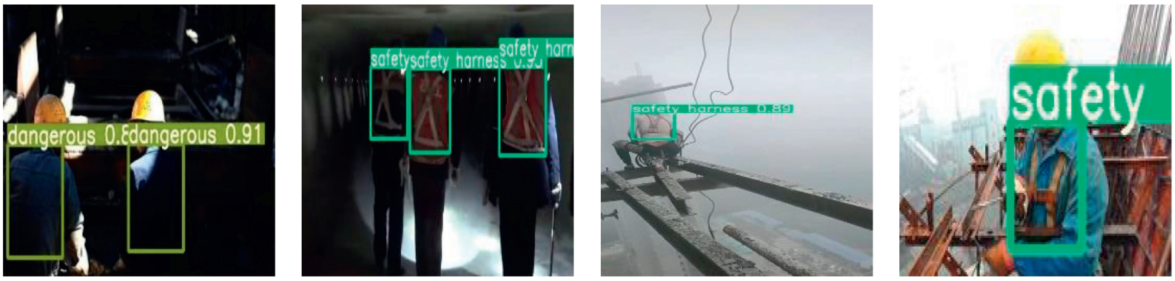
0.84 @Frame450

0.73 @Frame550

0.89 @Frame650

0.81 @Frame750

Figure 12: Surveillance video detection results.



(a) (b) (c) (d)

Figure 13: Various conditions. (a) Low-level light. (b) Dim light of night. (c) Rainy or foggy. (d) Low-resolution.

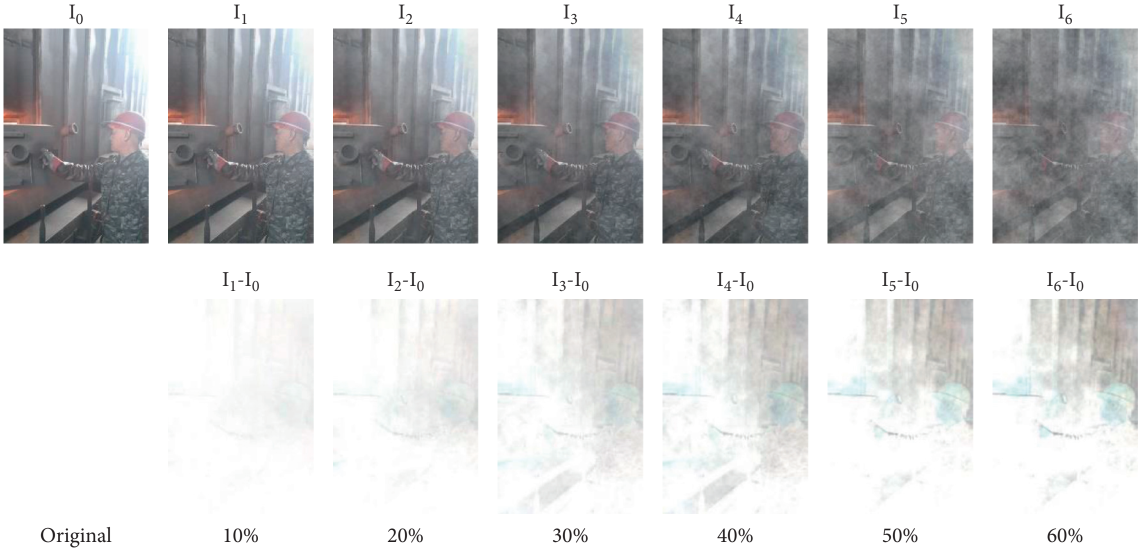
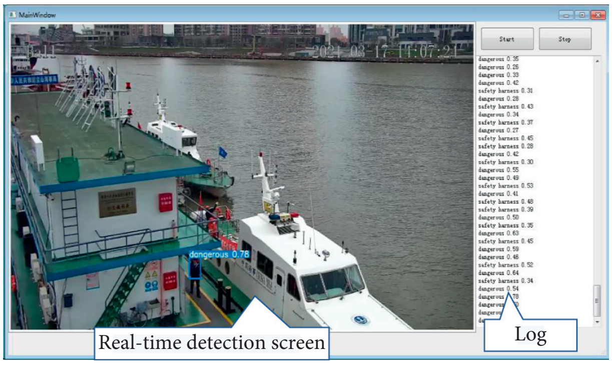


Figure 14: Pictures covered with different coverage rates of Perlin Noise.

Table 4: Obtained results from different coverage rates of Perlin Noise.

Coverage rates	Precision (P) (%)	Recall (R) (%)	mAP@0.5 (%)	mAP@.5:.95 (%)
Original	97.4	91.4	91.5	59.2
10%	97.4	91.0	91.1	59.0
20%	97.1	90.4	90.5	58.8
30%	97.9	89.2	89.3	57.5
40%	97.8	86.4	86.5	55.5
50%	98.0	79.4	79.5	49.7
60%	97.8	65.4	65.5	40.2



```
dangerous 0.35
dangerous 0.26
dangerous 0.33
dangerous 0.42
safety harness 0.31
dangerous 0.28
safety harness 0.43
dangerous 0.34
safety harness 0.37
dangerous 0.27
safety harness 0.45
safety harness 0.28
dangerous 0.42
safety harness 0.30
dangerous 0.55
dangerous 0.49
safety harness 0.53
dangerous 0.41
safety harness 0.48
safety harness 0.39
dangerous 0.50
safety harness 0.35
dangerous 0.63
safety harness 0.45
dangerous 0.59
dangerous 0.46
safety harness 0.52
dangerous 0.64
safety harness 0.34
dangerous 0.54
dangerous  78
dangerou  0
dangerou
da
```

Figure 15: Detection demo software.

6. Conclusion

In the present study, an algorithm is designed to resolve shortcomings of the manual supervision of safety rope wearing in ship operation sites and has high detection accuracy and speed under different resolutions and target sizes. The performed experiments show that, compared with the state-of-the-art target detection algorithms, the YOLO-SD algorithm has a good detection effect, while it is robust in real time. The present article is only focused on the accuracy and speed problems in the detection process of crew safety rope wearing and has not researched algorithm transplantation to artificial intelligence chip or edge computing device.

Data Availability

Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key Research and Development Project of China (Grant no. 2019YFB1600605), National Natural Science Foundation of China (Grant nos. 51709167 and 61701299), Shanghai Science and Technology Innovation Action Plan (Grant no. 18DZ1206101), Shanghai Pujiang Project (Grant no. 18PJD017), and Shanghai National Natural Science Foundation (Grant no. 18ZR1417100).

References

[1] Daily international maritime news, “Why real-time data matters to the maritime,” 2020, https://www.MaritimeCyprus.com.

[2] X. Y. Wu, “Innovative application of intelligent networking + big data in the management of high-risk construction work,” *Building management*, vol. 47, no. 2, pp. 66-67, 2020.

[3] P. Zhou, G. Zhou, H. Wang, D. Wang, and Z. He, “Automatic detection of industrial wire rope surface damage using deep learning-based visual perception technology,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, Article ID 5002911, 2021.

[4] P. Zhou, G. B. Zhou, Z. Z. He, C. Tang, Z. Zhu, and W. Li, “A novel texture-based damage detection method for wire ropes,” *Measurement: Journal of the International Measurement Confederation*, vol. 148, Article ID 106954, 2019.

[5] J. S. Li, S. Z. Yang, W. X. Lu, and Y. S. Wang, “Space-domain feature-based automated quantitative determination of localized faults in wire ropes,” *Materials Evaluation*, vol. 48, no. 3, pp. 336–341, 1990.

[6] H. Shi, L. Zheng, S. Sun, L. Zhang, and L. Wang, “Research on wire rope wear detection based on computer vision,” *Journal of Engineering*, vol. 2020, no. 13, pp. 517–519, 2020.

[7] L. Xuan, “Steel wire rope section damage detection system based on wavelet Multi-Resolution Analysis,” *Advanced Materials Research*, vol. 142, pp. 184–187, 2011.

[8] J. Tian, H. Wang, B. Sven, and x. Wu, “Application of a morphological non-sampling wavelet method to the online detection signal processing of coal mine wire rope,” *Insight: Non-Destructive Testing and Condition Monitoring*, vol. 61, no. 9, pp. 521–527, 2019.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USAarXiv:1506.02640, Las Vegas, USA, July 2016.

[10] X. Chen, L. Qi, Y. Yang et al., “Video-based detection infrastructure enhancement for automated ship recognition and behavior analysis,” *Journal of Advanced Transportation*, vol. 2020, Article ID 7194342, 12 pages, 2020.

[11] J. Zhao, J. Lu, X. Chen, Z. Yan, Y. Yan, and Y. Sun, “High-fidelity data supported ship trajectory prediction via an ensemble machine learning framework,” *Physica A: Statistical Mechanics and Its Applications*, vol. 586, Article ID 126470, 2022.

[12] X. Chen, Z. Li, Y. Yang, L. Qi, and R. Ke, “High-resolution vehicle trajectory extraction and denoising from aerial videos,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3190–3202, 2021.

[13] W. L. Shang, J. Chen, and H. Bi, “Impacts of user behaviors and environmental benefits of bike sharing: a big data analysis,” *Applied Energy*, vol. 285, Article ID 116429, 2020.

[14] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, September 1999.

[15] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, San Diego, USA, July 2005.

[16] C. Papageorgiou, M. Oren, and T. Poggio, “General framework for object detection,” in *Proceedings of the Sixth International Conference on Computer Vision*, Bombay, India, January 1998.

[17] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[18] J. A. K. Suykens, “Support vector machines: a nonlinear modelling and control perspective,” *European Journal of Control*, vol. 7, no. 2-3, pp. 311–327, 2001.

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

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

[21] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448arXiv:1504.08083, Santiago, Chile, June 2015.

[22] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[23] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6517–6525, Honolulu, USA, July 2017.

[24] J. Redmon and A. Farhadi, “YOLOv3: an incremental improvement,” *arXiv e-prints, arXiv:1804.02767*, 2018.

[25] A. Bochkovskiy, C. Y. Wang, and H. Liao, “YOLOv4: optimal speed and accuracy of object detection,” Arxiv.arXiv: 2004.10934, 2020.

[26] W. Liu, D. Anguelov, D. Erhan et al., “SSD: single shot multibox detector,” in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Berlin, Heidelberg, October 2016.

[27] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: scalable and efficient object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10778–10787, Seattle, USA, July 2020.

[28] M. Kisantal, Z. Wojna, and J. Murawski, “Augm`entation for small object detection,” in *Proceedings of the 9th International Conference on Advances in Computing and Information Technology*arXiv:1902.07296, Sydney, Australia, December 2019.

[29] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Chen, J.-W. Hsieh, and I.-H. Yeh, “CSPNet: a new backbone that can enhance learning capability of CNN,” in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1571–1580, Seattle, USA, June 2020.

[30] T. Lin, P. Dollar, and R. Girshick, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, January 2017.

[31] H. Rezatofighi, N. Tsoi, and J. Gwak, “Generalized intersection over union: a metric and A loss for bounding box regression,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 658–666, Long Beach, USA, July 2019.

[32] Z. Zheng, P. Wang, W. Liu, and J. Li, “Distance-IoU loss: faster and better learning for bounding box regression,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12993–13000, 2019.

[33] K. Perlin, “Improving noise,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 681-682, 2002.

[34] M. R. Gamarra Acosta, J. C. Schettini Castro, and N. S. Castro, “An innovative image-processing model for rust detection using Perlin Noise to simulate oxide textures,” *Corrosion Science*, vol. 88, no. 6, pp. 141–151, 2014.

[35] H. J. Bae, C. W. Kim, N. Kim et al., “A Perlin Noise-based augmentation strategy for deep learning with small data samples of HRCT images,” *Scientific Reports*, vol. 8, Article ID 17687, 2018.