Research Article

Multiscale Object Detection Method for Track Construction Safety Based on Improved YOLOv5

Zhen Xue1,1, Liangliang Zhang,1 and Bo Zhai2

1School of Science, North University of China, Taiyuan, Shanxi 030051, China
2Department of R & D Technology, Shanxi Shiheng Track Technology Co., Ltd, Taiyuan 030006, China

Correspondence should be addressed to Zhen Xue; xuezhen@nuc.edu.cn

Received 17 February 2022; Revised 9 June 2022; Accepted 18 July 2022; Published 31 August 2022

Academic Editor: Muhammad Haroon Yousaf

Copyright © 2022 Zhen Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In track construction, it is an important and necessary guarantee for production safety to check the number of workers and tools before and after the track maintenance. In view of time-consuming, laborious, and low detection efficiency of traditional manual inspection way, we propose an improved YOLOv5 multiscale object detection algorithm for track construction safety in this paper. We improve, from Generalized Intersection over Union (GIoU) to Distance Intersection over Union (DIoU), the loss function for Bounding Box (BBox) regression to speed up the convergence of the model. And we also improve, from the Non-Maximum Suppression (NMS) to DIoU-NMS, the post-processing method to enhance the model’s detection ability for occluded objects and small objects. The experiment results on the Track Maintenance dataset (our self-prepared dataset) and MS COCO dataset show that the mAP value of our improved YOLOv5 algorithm is 94.8% and 38.7%, respectively. Compared with the original YOLOv5 algorithm, the mAP values on above datasets are increased by 5.1% and 5.4%, respectively. The validation experimental results on MS COCO dataset and Track Maintenance dataset indicate that the detection ability of our improved YOLOv5 algorithm for occluded objects and small objects is enhanced. The proposed algorithm can provide technical support for the real-time accurate detection of track construction workers and tools.

1. Introduction

Up to the end of 2021, the operating mileage of China’s high-speed railway has reached about 41,000 km, ranking the first in the world, which also brings severe challenges to the safety of railway maintenance construction. The inconsistency of the number of workers and tools before and after the track may cause serious traffic accidents. At present, the main checking way for the number of workers and tools is manual inspection, which is time-consuming, laborious, and with low detection efficiency. With the rapid development of deep learning and computer vision, the unmanned intelligent object detection technology is becoming more and more popular for its detection advantages such as low cost and high efficiency.

At present, the deep-learning-based object detection algorithms are mainly divided into two categories: two-stage method and one-stage method. When the two-stage method is applied, the region candidate border is first generated, then the feature of each candidate border is extracted, and finally the ultimate location box is generated, and further the category is predicted. The representative networks of the two-stage method are Region-based Convolutional Neural Networks (R–CNN) series [1–4]. In contrast, when the one-stage method is adopted, the classification and regression are carried out at the same of generating the candidate border. The representative networks of the one-stage method are You Only Look Once (YOLO) series [5–9] and Single Shot Multibox Detector (SSD) series [10–14]. Since the whole network only includes convolutional layers and input images, after convolution operation the object’s category and position are directly returned. To sum up, the one-stage method is of higher detection speed; whereas the two-stage method is of higher recognition accuracy. For track construction, the two-stage detectors are more accurate but fail to meet the needs for real-time detection, so we select the one-stage method with higher detection speed, in which YOLO is widely adopted [5].

In 2021, Zhou et al. [15] presented a safety helmet detection method based on YOLOv5 algorithm in order...
to establish a digital safety helmet monitoring system. Yan et al. [16] proposed a light-weight fruit target real-time detection method for the apple picking robot based on improved YOLOv5. In view of the low detection accuracy of traditional traffic sign recognition algorithms, Lv and Lu [17] presented a traffic sign recognition method based on the improved YOLOv5 algorithm. Huang et al. [18] introduced Convolutional Block Attention Module (CBAM) to the architecture of YOLOv5 algorithm, improved the loss function for regression, and proposed an improved YOLOv5 model which can complete the rapid recognition task of citrus fruit in orchards. Chu [19] presented a target detection algorithm for the tank and armored vehicles based on YOLOv5 algorithm. Aiming at the disadvantage of the traditional ship detection method that requires manual feature selection, time-consuming and laborious, Zhang et al. [20] proposed a remote sensing image ship detection method based on YOLOv5. Yao et al. [21] applied deep learning technology to kiwifruit flaw detection and presented a real-time detection algorithm for kiwifruit flaw based on YOLOv5. Yang et al. [22] proposed a detection algorithm based on YOLOv5 which can recognize whether faces in public places such as malls and factories are wearing masks. Recently, Shu et al. [23] combined the characteristics of DenseNet network and YOLOv5 network and proposed a kind of Dense-YOLOv5 network structure, which has a better recognition effect for the small objects with unclear features. Tan et al. [24] presented a real-time detection method based on YOLOv5 algorithm in view of the deficiency of the existing manual detection mask-wearing situation. In 2020, Liu et al. [25] applied K-means method to find the most appropriate anchors for dataset and proposed a mask-wearing recognition method based on YOLOv5 algorithm. Chen et al. [26] adopted YOLOv5 algorithm for the ship detection and can well detect large and small ship objects, objects occluded by clouds and fog, and objects under complex sea wave background in satellite remote sensing images.

In view of the complexity of track construction scene and the requirement of real-time detection of construction workers and tools, it is necessary to solve the recognition problem of the occluded objects and small objects. Therefore, we select YOLOv5 as the principal algorithm and improve it. We first adopt the images both uploaded from operation recording instrument and collected from Internet to make a detection dataset of construction workers and tools. Then we apply the improved YOLOv5 algorithm to recognize the number of workers and tools before and after the track operation so as to get the detection accuracy and efficiency of the proposed model.

The remainder of this paper is organized as follows. Section 2 introduces the network architecture of YOLOv5 algorithm and presents the improved YOLOv5 algorithm. Section 3 demonstrates the experimental results and corresponding analysis. Finally, Section 4 is the conclusion.

2. Method

2.1. Characteristics and Network Architecture of YOLOv5 Algorithm. YOLO (v1-v5) is a series of object detection algorithms based on the deep neural network. YOLOv1 algorithm [5] was proposed in 2016 by the father of YOLO, Joseph Redmon. Henceforth, YOLOv2 (i.e., YOLO 9000) [6] and YOLOv3 [7] were presented in 2017 and 2018, respectively. YOLOv5 [9] is the latest model of YOLO series and was announced by Ultralytics LLC on June 10, 2020, less than 50 days since YOLOv4 algorithm [8] was proposed on April 23, 2020.

The biggest advantage of YOLO series is that they can recognize the category and position of multiple items in the image at one time and complete the recognition task from end to end, causing the speed greatly heightened. YOLOv5 inherits most advantages of YOLOv4 and it is far superior to the latter in speed and flexibility. For instance, the real-time speed of YOLOv5s network can achieve 140FPS (that of YOLOv4 is 50FPS) on Tesla P100 GPU and the size of YOLOv5s network is only 27 MB (that of YOLOv4 is 244 MB), which enables the model to be suitable for deployment to the embedded devices.

The network architecture diagram of YOLOv5 is shown in Figure 1.

As shown in Figure 1, YOLOv5 network architecture is mainly composed of four parts: Input, Backbone, Neck, and Prediction (or Head). The input terminal includes mosaic data augmentation, adaptive anchor computation, and adaptive image scaling. Mosaic data augmentation is applied to enable the model better detect the small object in the image. And adaptive anchor computation is used to make the prediction results more reasonable by updating the area of the prediction box of output per iteration [27]. And adaptive image scaling is applied to scale the original size to the standard one in the unified framework and then is sent to the network for detection, without modifying the image size before detection. The backbone network includes three parts: Focus, Cross Stage Partial connections (CSP) [28], and Spatial Pyramid Pooling (SPP). CSP1_1 and CSP2_1 are applied to backbone and neck, respectively. As for Focus, the technology of slicing the image performs well during the transformation from the input image to feature image, improving the computation efficiency of the algorithm. The CSP structure can be used to extract rich information feature from the input image and is good for enhancing the learning ability of the network. The SPP module with maximum pooling is able to fuse different scale features. The neck is composed of Feature Pyramid Network (FPN) [29] and Path Aggregation Network (PAN) [30] which can effectively improve the feature extraction ability of network. For the prediction end (or output end), GloU_Loss [31] is selected as the loss function for Bounding Box (BBox) regression, and
the final detection box is subjected to non-maximum suppression (NMS) processing to obtain the optimal target box.

2.2. Improved YOLOv5 Algorithm. In track construction, when using the original YOLOv5 algorithm to detect the multiscale objects and occluded objects, the missed and false detection occur sometimes, and the accuracy does not meet the actual requirements. In order to promote the detection ability of multiscale objects and occluded objects, we improved the loss function and NMS for BBox regression of YOLOv5 algorithm in this paper.

2.2.1. Improvement of Loss Function from GIoU to DIoU. The quality of loss function directly affects the training speed and performance of the object detector. During the regression of the detection box of YOLOv3 algorithm, Intersection over Union (IoU) metric [32, 33] is applied to measure the overlapping area between the predicted BBox and the ground-truth BBox divided by the area of union between them, and the corresponding loss function [31] is defined as follows:

\[
L_{\text{IoU}} = 1 - \text{IoU} = 1 - \frac{|A \cap B|}{|A \cup B|} \tag{1}
\]

where \(A\) denotes the predicted BBox; \(B\) denotes the ground-truth BBox; \(|A \cap B|\) denotes the area of intersection of \(A\) and \(B\); \(|A \cup B|\) denotes the area of the union of \(A\) and \(B\). IoU can better reflect the overlapping degree between the two BBoxes, has the feature of scale invariance, and can be applied to measure the accuracy of prediction. Nevertheless, in reality there is a case that the predicted BBox and the ground-truth BBox do not intersect. At this time, the value of IoU is equal to 0 and the value of the loss function is always equal to 1, so learning cannot continue.

To solve the gradient vanishing problem for nonoverlapping cases, Generalized Intersection over Union (GIoU) loss function is adopted in YOLOv5 algorithm, which is defined as follows:

\[
L_{\text{GIoU}} = 1 - \frac{|A \cap B|}{|A \cup B|} + \frac{|D|}{|C|} \tag{2}
\]

where the definitions of \(A\) and \(B\) are the same as that of (1); \(C\) is the smallest region that simultaneously covers \(A\) and \(B\); let \(D = C / (A \cup B)\), then \(C = A \cup B \cup D\); \(|C|\) and \(|D|\) denote the area of \(C\) and \(D\), respectively. If the predicted box is inside the ground-truth box and the size of all predicted boxes is equivalent, then the difference set between the ground-truth box and each predicted box is identical, GIoU is degenerated into IoU, and we cannot distinguish relative position relationship.

In fact, both IoU and GIoU only consider the overlapping area, and the corresponding loss functions have two disadvantages such as slow convergence and inaccurate regression. Nevertheless, Distance Intersection over Union (DIOU) also considers the normalized distance between the central points of predicted BBox and ground-truth BBox. The corresponding loss function is defined as follows [32, 34]:

\[
L_{\text{DIOU}} = 1 - \frac{|A \cap B|}{|A \cup B|} + \frac{d^2}{c^2} \tag{3}
\]

where \(d = \rho(b, b^{gt})\) is the Euclidean distance of central points of predicted BBox and ground-truth BBox; \(c\) is the diagonal length of the smallest enclosing box covering two BBoxes. If the two boxes are in the horizontal or vertical direction at the same time, then DIOU loss will make the model get rapid regression. DIOU loss can directly minimize the normalized distance between central points, achieving faster convergence speed and more accurate regression [34].

The IoU, GIoU, and DIOU, as stated above, are applied to describe the similarity between predicted BBbox and
ground-truth BBox. The comparison among their calculation methods is shown in Figure 2.

2.2.2. Improvement of NMS from Original NMS to DIoU-NMS. The NMS is applied to search the local maximum and suppress the non-maximum element. NMS is the final step for most object detection algorithms and usually applied to select the predicted BBox in the reasoning process, i.e., to remove the redundant boxes. The original NMS is based on the classification confidence score, only the predicted BBox with the highest confidence score can be preserved. In most cases, there is not a strong correlation between IoU and classification confidence score, thus the location of many classification labels with high confidence scores is not very accurate. In applying the YOLOv5 algorithm with original NMS the analysis is done only through overlapping areas and the missed and false detection are more likely to emerge, especially in scenes with highly overlapping objects.

In order to improve the detection performance for occluded object, we improve the NMS as DIoU-NMS. Different from the original NMS whose criterion is IoU, DIoU-NMS takes DIoU as the criterion for suppressing the redundant boxes. It considers not only the overlapping area, but also the distance between central points of two BBoxes. DIoU-NMS is conducted by simultaneously considering IoU and the distance between central points of the two BBoxes. DIoU-NMS is defined as

\[
s_i = \begin{cases} 
  s_i, & \text{IoU} - R_{DIoU}(M, B_i) < \varepsilon \\
  0, & \text{IoU} - R_{DIoU}(M, B_i) \geq \varepsilon,
\end{cases}
\]

where \(s_i\) is the classification confidence score; IoU is defined in Equation (1); \(\varepsilon\) is the NMS threshold; \(M\) is the predicted BBox with the highest score; \(B_i\) is the pending BBox. DIoU-NMS is conducted by simultaneously considering IoU and the distance between central points of two BBoxes. The distance is denoted by \(R_{DIoU}\) and the corresponding formula is as follows:

\[
R_{DIoU} = \frac{\rho^2(b, b^*)}{c^2},
\]

where the definitions of \(b, b^*, \rho\) and \(c\) are the same as that of (3).

We suggest that two BBoxes with bigger IoU and bigger central points’ distance probably locate two different objects and should not be removed [34]. Therefore, DIoU-NMS is more robust than original NMS for suppressing redundant boxes. And it is very flexible to be incorporated into any object detection algorithm [34].

3. Experimental Results and Corresponding Analysis

3.1. Experimental Environment. The software and hardware configurations in the experiment are shown in Table 1.

3.2. Training, Testing, and Evaluation

3.2.1. Experiment Dataset. To verify the effectiveness of our improved YOLOv5 algorithm, we trained and tested the model on two datasets. One is the popular object detection benchmark dataset Microsoft Common Objects in COntext (MS COCO) [35] proposed and constructed by the Microsoft team. It is a large dataset used for classical computer vision tasks such as object detection, semantic segmentation, key point detection. The other is Track Maintenance dataset, a self-prepared dataset, which contains data both uploaded from the operation recording instrument and collected from Internet by crawlers. The Track Maintenance dataset contains various scenes of the construction site and has 7268 images in total. 32 images of which are shown in Figure 3.

The MS COCO2017 dataset involves 80 object categories. There are more than 330k images in it, and more than 200k

![Figure 2: The calculation method comparison of IoU, GIoU, and DIoU.](image)

Table 1: Experimental platform configuration.

<table>
<thead>
<tr>
<th>Experimental environment</th>
<th>Experimental side</th>
<th>Deployment side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Framework</td>
<td>PyTorch 1.7.1</td>
<td>PyTorch 1.7.1</td>
</tr>
<tr>
<td>Data processing</td>
<td>Python 3.8.5</td>
<td>Python 3.8.5</td>
</tr>
<tr>
<td>Operating system</td>
<td>Windows 10</td>
<td>Windows server</td>
</tr>
<tr>
<td>CPU</td>
<td>17–9700K 3.6 GHz</td>
<td>E5_4607 2.6 GHz</td>
</tr>
<tr>
<td>GPU</td>
<td>NVIDIA</td>
<td>—</td>
</tr>
<tr>
<td>GPU processor</td>
<td>GTX1080Ti</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>CUDA10.1, cudnn7.6</td>
<td>—</td>
</tr>
</tbody>
</table>
images of which have been annotated. For Track Maintenance dataset, we obtained YOLO format label by applying LabelImg software to annotate the objects of the image. There are 22 kinds of objects, of which 21 kinds are tool objects. We annotated the construction worker as "person," and the tool objects as "gongjv 1," "gongjv 2," etc.

Given the hardware performance of the deployment side is relatively low (No GPU acceleration), we preferred applying YOLOv5s with the lowest depth and width. In addition, among the four versions of YOLOv5, YOLOv5s has the smallest network structure and the most considerable detection speed [27]. Improved YOLOv5s algorithm models for the normalized distance between the predicted box and ground-truth box with DIoU loss function and can directly minimize the distance between the two boxes. Therefore, the improved YOLOv5s algorithm converges faster than the original one with GIoU loss function.

For Track Maintenance dataset with 7268 images, based on the experience of selecting the proportion of the training/test set, we divided the dataset into a training set (5814 images) and a test set (1454 images) at a ratio of 8:2 by applying the hold-out method [36]. The training set includes samples with different resolutions to ensure the generalization ability of the model. For MS COCO2017 dataset, we took train2017 with 118k images as training set and test-dev with 20k images as test set.

During training, we took epoch as 2000 and batch size as 16. In order to accelerate the convergence process, we set the original learning rate to 0.02 and then dynamically adjust it in the follow-up training process. The hyperparameters during training are shown in Table 2.

![Figure 3: Some images of track maintenance dataset.](image)

### Table 2: Hyperparameters during training.

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original learning rate</td>
<td>0.02</td>
</tr>
<tr>
<td>Cyclical learning rate</td>
<td>0.2</td>
</tr>
<tr>
<td>Weight decay</td>
<td>0.0005</td>
</tr>
<tr>
<td>Momentum epoch</td>
<td>0.937 2000</td>
</tr>
<tr>
<td>Batch size</td>
<td>16</td>
</tr>
</tbody>
</table>

During training, the loss function can intuitively reflect whether the network model can stably converge as the number of iterations increases. The loss functions adopted in YOLOv5 algorithm and our improved one during training consists of localization loss ($L_{box}$), confidence loss ($L_{conf}$), and classification loss ($L_{cls}$). The overall loss ($Loss$) is the weighted sum of the above three losses and the corresponding formula is shown in (6).

$$Loss = \lambda_{box}L_{box} + \lambda_{conf}L_{conf} + \lambda_{cls}L_{cls}, \quad (6)$$

where $\lambda_{box}$, $\lambda_{conf}$, and $\lambda_{cls}$ represent the weighted value of localization loss, confidence loss, and classification loss, respectively.

At the end of training, we extracted the loss function value in the training log to draw the curve. The training loss curves of the original YOLOv5 algorithm and our improved one are shown in Figure 4.

As shown in Figure 4, with the increasing of epoch, the value of the loss function of our improved YOLOv5 algorithm declines faster than the original one at the initial stage of training. And the overall fluctuation amplitude of our improved YOLOv5 algorithm is smaller. After 600 rounds of training, the loss curve of the improved YOLOv5 model tends to be stable. The loss values of the original algorithm and the improved one converge to about 0.038 and 0.065, respectively. To sum up, the improved YOLOv5 model has a better training effect.

#### 3.2.2. Evaluation Metrics.

In the study, evaluation metrics such as Precision ($P$), Recall ($R$), mean Average Precision (mAP), and Frames per Second (FPS) are applied to evaluate the model’s performance. Precision is the ability of a model to identify only relevant objects and represents the percentage of truly positive predictions among the positive classes [37]; Recall is a measure of coverage and represents the percentage of truly positive predictions among all given ground-truth BBoxs [37]; mAP is applied to measure recognition accuracy; FPS is employed to evaluate the speed of object detection, i.e., the number of frames that can be
processed per second. The calculating formulas of $P$, $R$, and $mAP$ are shown in Equations (7–10).

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{all detections}}$$ \quad (7)

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{all ground truths}}$$ \quad (8)

$$AP = \int_0^1 P(R)dR,$$ \quad (9)

$$mAP = \frac{\sum_{i=1}^C AP(i)}{C},$$ \quad (10)

where $TP$, $FP$, and $FN$ are the number of true positive samples (i.e., a correct detection of a ground-truth BBox), false positive samples (i.e., an incorrect detection of a nonexistent object or a misplaced detection of an existing object), and false negative samples (i.e., undetected ground-truth BBox), respectively [37]; the P-R curve can be obtained by the highest precision at different recall, for each different recall value, select the maximum precision whose recall value is greater than or equal to the given recall value, and then calculate the area under the P-R curve as the $AP$ value. Average Precision ($AP$) is a common metric for evaluating the accuracy of object detector and $AP(i)$ is $AP$ of the $i^{th}$ category; $mAP$ is the mean of $AP$s for all categories, and the bigger the value is, the better the detection performance of the algorithm; $C$ is the total number of detection categories being evaluated, for Track Maintenance dataset the value of parameter $C$ is 22 and for COCO dataset that is 80; $mAP@0.5$ refers to the average $AP$ of all categories when IoU threshold is set to 0.5; $mAP@0.5:0.05:0.95$ refers to the average $AP$ at different IoU thresholds and the IoU threshold ranges from 0.5 to 0.95 with a step size of 0.05, i.e., $mAP@0.5:0.05:0.95 = (mAP@0.5 + mAP@0.55 + \ldots + mAP@0.95)/10$.

### Table 3: YOLOv5 ablation experiment. %.

<table>
<thead>
<tr>
<th>DIOU</th>
<th>DIOU-NMS</th>
<th>Precision</th>
<th>Recall</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>91.2</td>
<td>85.6</td>
<td>90.2</td>
</tr>
<tr>
<td>√</td>
<td>-</td>
<td>90.3</td>
<td>86.7</td>
<td>91.5</td>
</tr>
<tr>
<td>-</td>
<td>√</td>
<td>92.8</td>
<td>87.2</td>
<td>94.1</td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>96.7</td>
<td>86.8</td>
<td>94.8</td>
</tr>
</tbody>
</table>

### Table 4: Performance comparison between our improved YOLOv5 model and other state-of-the-art models on Track Maintenance dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP/%</th>
<th>Detection speed/FPS</th>
<th>Memory size/MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R–CNN</td>
<td>91.3</td>
<td>10</td>
<td>195.3</td>
</tr>
<tr>
<td>YOLOv3</td>
<td>82.6</td>
<td>20</td>
<td>242.2</td>
</tr>
<tr>
<td>YOLOv5</td>
<td>90.2</td>
<td>26</td>
<td>14.5</td>
</tr>
<tr>
<td>Our YOLOv5</td>
<td>94.8</td>
<td>29</td>
<td>14.6</td>
</tr>
</tbody>
</table>

### Table 5: Performance comparison between our improved YOLOv5 algorithm and other state-of-the-art models on COCO dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP/%</th>
<th>Detection speed/FPS</th>
<th>Memory size/MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R–CNN</td>
<td>36.8</td>
<td>9</td>
<td>315</td>
</tr>
<tr>
<td>YOLOv3</td>
<td>33</td>
<td>20</td>
<td>235</td>
</tr>
<tr>
<td>YOLOv5</td>
<td>36.7</td>
<td>140</td>
<td>27</td>
</tr>
<tr>
<td>Our YOLOv5</td>
<td>38.7</td>
<td>147</td>
<td>28</td>
</tr>
</tbody>
</table>

### Figure 4: The training loss curves of YOLOv5 and YOLOv5-Ours.
carried out on the dataset to judge the effectiveness of each improvement strategy. The DIoU loss function and DIoU-NMS are introduced to the original model in turn, "√" indicating that improvement is adopted, and "−" indicating that improvement is not adopted. The same parameter configuration is adopted during training and the results are shown in Table 3.

As shown in Table 3, after the introduction of DIoU-NMS, the precision, recall, and mAP value are improved by 1.75%, 1.87%, and 4.32%, respectively. After the introduction of DIoU loss function, the recall and mAP values are improved by 1.29% and 1.44%, respectively. After the simultaneous introduction of DIoU and DIoU-NMS, the precision and mAP value are improved by 6.03% and 5.1%, respectively, and the recall value also gains little improvement by 1.4%.

In order to further verify the better effect after applying our improved YOLOv5 algorithm, we selected the state-of-the-art object detection algorithms such as Faster R–CNN, YOLOv3, and YOLOv5 to carry out performance comparison experiment, under the same configuration environment and on the same dataset. The FPS is measured on a NVIDIA GTX1660Ti GPU. The corresponding results on Track Maintenance dataset and COCO dataset are shown in Table 4 and Table 5, respectively.

The results in Table 4 show that the recognition accuracy of our algorithm is higher than that of Faster R–CNN, YOLOv3, and YOLOv5. The mAP (i.e., mAP@0.5:0.05:0.95) value is improved by 5.1% from the original YOLOv5 to our improved YOLOv5 model. The detection speed of our improved model can achieve 29 FPS, which is 1.12, 1.45, and 2.9 times of YOLOv5, YOLOv3, and Faster R–CNN, respectively, indicating that our proposed model can meet the demand of real-time detection. The memory size of our improved algorithm is slightly bigger than the YOLOv5 algorithm and is reduced to one-seventeenth of YOLOv3 and one-thirteenth of Faster R–CNN, respectively.

The results in Table 5 show that the average recognition accuracy of our improved algorithm is higher than that of other algorithms listed in the table. The mAP value, from original YOLOv5 to our improved YOLOv5 model, is improved by 5.4%. The detection speed of our improved model can achieve 147 FPS, which is 1.05, 7.35, and 16.33 times of YOLOv5, YOLOv3, and Faster R–CNN, respectively, indicating that our proposed model is faster than the three models. The memory size of our improved algorithm is
slightly bigger than YOLOv5 algorithm and is reduced to one-eleventh of YOLOv3 and one-eighth of Faster R–CNN, respectively.

The improved YOLOv5 model presented in this paper reduces many additional computing power, maintains high recognition accuracy and detection speed, and occupies less memory resources. It is suitable for deployment on the mobile embedded device platform and has obvious advantages over the other three models.

3.3. Visualization Analysis of Experimental Result

3.3.1. On MS COCO Dataset. To compare the detection result between the original YOLOv5 model and our improved one more intuitively, we randomly selected some images from the test-dev set of COCO dataset for experiment verification. The results are shown in Figure 5. Figures 5(a)–5(c) are the original image, detection results of the original YOLOv5 algorithm, and the improved one, respectively. Each output box is associated with a category label and a confidence score in [0, 1]. A score threshold of 0.6 is applied to display these images.

As shown in Figure 5, there are some missed detections in using the original YOLOv5 model. It does not perform well in processing occluded objects and small-scale objects. In contrast, the improved YOLOv5 model reduces the influence of scale invariance, has higher detection rate, and effectively enhances the recognition accuracy of multiscale object.

3.3.2. On Track Maintenance Dataset. To compare the detection accuracy between the original YOLOv5 algorithm and our improved YOLOv5 one more intuitively, we also selected some images from the test set of the Track Maintenance dataset. The detection results are shown in Figure 6. Figures 6(a)–6(c) are the original image, detection results of the original YOLOv5 algorithm, and our improved YOLOv5 one, respectively. Each output box is associated with a category label and a confidence score in [0, 1]. To examine the missed and false detections, a score threshold of 0.35 is adopted to display these images.

As shown in Figure 6, the worker object “person” and tool object “gongjv” can be effectively detected by using the original YOLOv5 algorithm and the improved one. Judging from the first row of Figure 6, we can obtain that there are some false detections in using the original YOLOv5 algorithm whereas these objects are accurately detected in using the improved YOLOv5 algorithm. As seen from the second
row of Figure 6, we can obtain that YOLOv5 algorithm cannot judge whether the middle object is a worker or a tool which is a difficult sample, whereas the improved YOLOv5 algorithm can detect that it is a tool and the convergence process is accelerated. The experimental results indicate that in applying the improved YOLOv5 algorithm, the detection ability of occluded object and samples which are hard to detect is heightened, thus the detection accuracy is improved.

In order to evaluate the generalization ability and robustness of the improved YOLOv5 algorithm, we selected different scenes and different type of tools for testing, the results are shown in Figure 7.

From Figure 7, it can be known that the improved YOLOv5 algorithm has strong detection ability in different scenes and can quickly judge the object of the worker and multiscale tool. As a consequence, we can draw the following conclusion: for the improved YOLOv5 algorithm, the generalization ability and robustness are strong and they are less affected by the factors such as scene, light, and color; the overall detection effect is satisfactory and both accuracy and detection speed has been improved effectively.

4. Conclusion

In this research, we apply the deep learning technology to track construction and propose an improved YOLOv5 algorithm for detecting the number of workers and tools in track construction. Taking the YOLOv5 algorithm as the main body, we improve the loss function and NMS for the BBox regression. In using the improved YOLOv5 algorithm, the convergence is accelerated, and the detection accuracy of occluded objects and small objects is enhanced. The experimental results show that the improved YOLOv5 algorithm has strong robustness. By applying this algorithm, we can solve the problem of low detection accuracy for sophisticated scene issues such as occluded objects and small objects, and effectively inspect the construction workers and tools, meeting the practical requirements for track construction safety detection. The research work of this paper is helpful to the profound study and development of safety detection technology of track construction, and further promotes the engineering application of intelligent detection equipment.

Data Availability

During research, the data come from three parts: the images uploaded from the operation recording instrument, the images collected from Internet, and the popular benchmark dataset MS COCO.

Conflicts of Interest

The authors declared that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61801437), Research Topics of Social and Economic Statistics in Shanxi Province (KY2021142), and Research Topics in machine learning (2110800005HX).

References


