Real-Time Object Detection for the Running Train Based on the Improved YOLO V4 Neural Network

Yang Liu, Mengfei Gao, Humin Zong, Xinping Wang, and Jinshuang Li

1School of Automation and Electrical Engineering, Dalian Jiaotong University, Dalian 116028, China
2National Marine Environmental Monitoring Center, Dalian 116023, China

Correspondence should be addressed to Yang Liu; muxiaoyi@126.com

Received 18 February 2022; Revised 26 April 2022; Accepted 18 May 2022; Published 9 June 2022

Copyright © 2022 Yang Liu 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.

Matching the detection speed and accuracy in practical applications is considered to improve the speed of video object detection in front of trains. Lightweight convolutional neural network MobileNet and clustering ideas are combined to improve the object detection algorithm, and the MYOLO-lite model object detection algorithm is designed. The self-made object dataset of a forward-moving train is combined with the K-means clustering idea. The transcendental frame is redesigned to enhance scale adaptability. The nonmaximum suppression and loss function improvement methods of the MYOLO-lite network model are proposed for occluded tracks in front of the train, obstruction of trains, low detection accuracy of large object coincidence, and uneven distribution of positive and negative samples. The mean average precision value of the experimentally designed MYOLO-lite model object detection algorithm reaches 95.74%, with 42.04 frames per second. Detecting a picture only takes 0.024 s.

1. Introduction

China’s railway lines have been expanding and developing extensively in recent years. Various types of algorithms based on deep learning are no longer bound as a vital part of the intelligent transportation system. Object detection in auxiliary train automatic driving has become a popular research topic. At present, various detection methods, including sensor fusion and traditional image processing methods, are used in rail transit object detection. Industry scholars have also studied and improved the application of machine learning to orbit detection. Guo et al. [1] proposed to correct the error of the object position by using the least-squares method because of the defects, such as poor adaptability and weak distance discrimination of traditional visual sensors. They also constructed a detection area based on radar parameters to detect obstacles in front of the train. Wang et al. [2] designed a jitter compensation algorithm for the current overlimit in cargo transport train image foreign body intrusion detection method when the train runs high speed. This algorithm is susceptible to the influence of the acquisition frequency and causes missed detection. It also obtains the foreign body intrusion into the object area based on the background difference and morphological corrosion expansion operation. Traditional image processing methods are not realistic enough to fit complex objects and cannot achieve an ideal detection effect for complex scenes, such as occlusion. Moreover, they have certain limitations for real-time requirements in the industry and cannot meet the object detection requirements in front of a running train in the new era.

As an extension of traditional machine learning, deep learning has developed rapidly in recent years in the context of big data. The essence of deep learning is to make the machine learning process close to the level of human intelligence. Its unique advantages in extracting image features have been widely studied and applied in computer vision, pattern matching, and pattern recognition. In 2013, Ross Girshick et al. [3] proposed a region convolutional neural network (R-CNN) object detection algorithm based on deep learning. It is the first deep learning object detection algorithm that can truly achieve industrial-grade applications. Its detection accuracy on the Pascal visual object class (VOC) dataset can reach 53.3%, which is 30% higher than the
Deep learning is a new direction for machine learning research. It excels in various tasks, such as image classification, speech recognition, image segmentation, and object detection. However, the applications in the field of rail transit are few, and practical and feasible results are lacking. Many scholars from other fields of applied research have studied and improved the issues mentioned to meet the requirements of various applications. These studies have great relevance for our research and application. Zhang [11] proposed a Dalation_DenseNet SSD model and improved it to detect and accurately locate small traffic sign objects. Wang et al. [12] determined that the aerial images of drones contain many small objects and are easily obscured. Thus, they added an squeeze-and-excitation (SE) attention mechanism based on the proposed S-YOLO V4 algorithm and other methods to improve accuracy. Li et al. [13] combined the MobileNet architecture with Faster R–CNN in the German Traffic Sign Detection Benchmark database and achieved good results. Liu [14] used the lightweight convolutional neural networks on YOLO V3. The algorithm was optimized to study vehicle object detection in autonomous driving. Du et al. [15] studied infrared object detection and detected occluded objects caused by complications due to obscured vehicles. YOLO based on attention mechanisms was used. Xie et al. [16] applied the combination of YOLO V4 algorithm and MobileNetv3 to flower detection.

Some preliminary explorations in the field of rail transit based on deep learning are conducted. Considerable research is combined with traditional methods to achieve effective detection of railway systems. Zhao et al. [17] proposed a forward train identification system based on the SSD network to solve the problem of judging obstacles in front of the subway train operation environment. Xu [18] proposed an orbital semantic segmentation algorithm based on residual SSD network and SegNet network-based encoder-decoder idea. A joint detection model of a shared feature extraction network was designed, and experimental results were presented. Li et al. [19] proposed a model segmentation method based on the existing YOLO V3 model to train obstacle detection and an edge computing-based method, which is used to help execute the object detection algorithm on the trackside. Sun [20] used YOLO V5 algorithm and DeepSORT algorithm to detect and track targets, and realized real-time detection and tracking of train positions in railway scenarios. Given the high requirements on the speed and accuracy of video detection for high-speed trains running ahead and the complex environment of a running train, some problems, such as occlusion, exist. Thus, the following are performed in this research:

(1) First, the production process and model evaluation index of the object dataset in front of the self-made train are introduced. Then, the improved MYOLO-lite algorithm is proposed. Based on YOLO V4, the improved lightweight MYOLO-lite model reduces the model parameters and the calculation complexity. Moreover, the a priori box is redesigned using the idea of the K-means clustering algorithm to enhance the adaptability of the scale. The effectiveness of the improved algorithm is proved through experiments and comparison with other models.

(2) The MV3-YOLO-lite algorithm is optimized for occlusion, the problems of positive and negative datasets, and difficult sample imbalance. Bounding box optimization based on flexible nonmaximum inhibition of Soft-NMS is studied. Based on the idea of focal loss, the complete intersection over union (CIOU)_Loss algorithm is fused to improve the loss function by adding a modulation factor α to the cross-entropy function.

The rest of this paper is organized as follows. Section 2 introduces the related works. The methods are presented in Section 3. The experiments and results are discussed in Section 4, and the conclusions are drawn in Section 5.

2. Related Works

2.1. YOLO V4 Model. In 2020, Alexey Bochkovskiy proposed YOLO V4 [21] based on YOLO V3. The foundation combines many techniques. No revolutionary change is made, but the performance is improved. In the overall detection idea, three feature layers are used for classification and regression prediction. CSPDarkNet-53 is used as the backbone feature extraction network. CSPNet [22] (i.e., cross-stage partial network) is added to each large residual block of DarkNet-53, which effectively improves CNN learning ability. In enhanced feature extraction networks, YOLO V4 combines a spatial pyramid pooling SPP structure with a PANet structure [23] to improve the YOLO V3 network. The SPP structure is in the convolution of the last feature layer on CSPDarkNet-53. After the three convolutions of the last feature layer on CSPDarkNet-53, stacking with four different scales of maximum pooling can increase receptive fields and separate salient features, and the pooled nuclei are $13 \times 13$, $9 \times 9$, $5 \times 5$, and $1 \times 1$ [24]. In addition, various techniques, such as mosaic data enhancement, label smoothing, CIOU, and learning rate cosine annealing attenuation, have been used in training.


2.2.1. MobileNet V1 Network. MobileNet V1 uses depthwise separable convolution instead of ordinary convolution; thus, the model parameters are considerably reduced, and the lightweighting of convolutional neural networks is achieved.
Compared with ordinary convolutions, depthwise separable convolutions split a convolutional kernel into two modules and extract features through $3 \times 3$ depthwise convolution and $1 \times 1$ pointwise convolution, thereby reducing the number of parameters and computational costs. Figure 1 shows a depthwise separable convolutional structure.

2.2.2. MobileNet V2 Network. In 2018, Google team launched MobileNet V2 [28]. After the actual training was completed, researchers determined that although MobileNet V1 considerably reduced the amount of computation, the deep convolution kernel was easily invalidated. Thus, V2 was based on V1. Inverted ResBlock [29] was used. It uses $1 \times 1$ pointwise convolution for the upswing. Then, feature extraction with $3 \times 3$ depthwise separable convolution, followed by $1 \times 1$ pointwise convolution for dimensionality reduction operation, was used. The input and output were directly connected through the residual edge section on the right. The specific structure is shown in Figure 2.

2.2.3. MobileNet V3 Network. Google proposed MobileNet V3 [30] in 2019. The network architecture, which uses a special bottleneck structure [31], combines the following features: the depthwise separable convolution from MobileNet V1 and a linear bottleneck of the inverted residual structure from MobileNet V2. The addition of attention mechanism after the depthwise separable convolution and the introduction of the SE module improves the accuracy of the network model without increasing time consumption; a new activation function, h-swish [32], is used instead of the swish function, thereby reducing the number of operations and improving performance. The swish and h-swish formula are defined as

$$\text{swish}(x) = x \cdot \sigma(x),$$

$$h - \text{swish}(x) = x \frac{\text{Relu6}(x + 3)}{6}. \quad (1)$$

Sigmoid calculation takes a long time, particularly on the mobile side. Thus, ReLU6 $(x + 3)/6$ is used to approximately replace sigmoid. Swish can improve efficiency by approximately 15% in deep networks using h-swish substitution.

3. Methods

3.1. MYOLO-Lite Series Model. Based on the YOLO V4 network, a lightweight convolutional neural network model is used in this study to optimize the basic structure of the YOLO V4 network. This study also aims to balance the efficiency and speed of the object detection algorithm. For YOLO V4, the network structure can be divided into three parts: backbone feature extraction network, corresponding to CSPDarkNet-53; enhanced feature extraction network, corresponding to SPP and PANet; and prediction network YoloHead, using the obtained features for prediction.

3.1.1. Backbone Feature Extraction Network. The MobileNet series network can be used for classification, which uses the trunk part for feature extraction. Using the three neural networks, MobileNet V1, MobileNet V2, and MobileNet V3, we can obtain each MobileNet network corresponding to...
YOLO V4 of the same three effective feature layers. We can use these three effective feature layers to replace the feature layer of the original YOLO V4 backbone network CSPDarkNet-53. The backbone structure of the MYOLO-lite series model network is shown in Figure 3.

3.1.2. Enhanced Feature Extraction Network. The depthwise separable convolutional block and activation function are defined based on the information circulation in the PANet network using depthwise separable convolution to reconstruct the enhanced feature extraction network structure. First, the semantic information of the top features is transmitted to the bottom network by upsampling. Then, the high-resolution information of the bottom features is fused to improve the effect of small object detection. The information transmission path from the bottom layer to the top layer is added, and the feature pyramid is strengthened by downsampling. Finally, the feature map of different layers is used to make predictions. The reconstructed network structure of the MYOLO-lite series model is shown in Figure 4.

The 3 × 3 convolution in PANet is replaced with a depthwise separable convolution block to ensure recognition accuracy. This block can be obtained by looking at the number of parameters before and after the replacement. Table 1 shows that the replaced model has a considerable effect on reducing the number of parameters.

3.2. Designed Anchor Box Based on K-Means Clustering Algorithm. In the object detection network, an anchor box must be used to detect the object in the image. The detection object is changed from traversing the entire picture to detecting the anchor box in each grid, considerably reducing the calculation complexity. The anchor box of YOLO V4 is obtained by clustering on the COCO dataset annotations. Thus, some mismatches with the self-made dataset used in this study exist. Given that the K-means algorithm clustering works well, it is used in this study to obtain the anchor box that matches the object of the self-made dataset. Therefore, the newly generated anchor box matches well with the real box in the dataset.

The K-means clustering algorithm is the most famous division clustering algorithm. It is widely used because of its simple and efficient implementation. The IOU is widely used to measure whether it is similar. Thus, the YOLO V4 algorithm uses the redefined distance formula as follows:

\[ d(\text{box, centroid}) = 1 - \text{IOU (box, centroid)} \]

where the box is a real box representing the label in the dataset, centroid is the clustering center, and the IOU is the intersection between the real box and the anchor box. \( d \) is small when the IOU value is large, and the real box matches well. The anchor values are (14, 68), (32, 39), (45, 144), (52, 193), (62, 203), (89, 69), (114, 247), (149, 249), and (169, 188).

For the three valid feature maps generated by YOLO V4, three anchor boxes are used for each size. The largest sensory field exists on the 13 × 13 feature map. The three sizes of anchor frames, namely, 114 × 247, 149 × 249, and 169 × 186, are used to detect the largest object. Moreover, 52 × 193, 62 × 203, and 89 × 69 are used on the feature map of 26 × 26. These three sizes of anchor frames are used to detect large objects. Furthermore, 32 × 39 and 45 × 144 are used on the feature plot of 52 × 52. These three sizes of anchor frames are used to detect medium objects.

3.3. MYOLO-Lite Model Optimization for Object Occlusion and Uneven Sample Distribution. The algorithm does not directly reject boxes with a degree of coincidence greater than the threshold. However, it calculates an attenuation coefficient according to the degree of coincidence, adjusts the score of adjacent boxes that coincide, and adjusts to a low score if the detection box has most of the coincidence with M. If the amount of coincidence is small, the original detection score does not change much, and the NMS operation continues after adjustment.

The improved Soft-NMS does not simply intend to suppress directly the detection score of the highly overlapping adjacent box bi, but it means to attenuate it. The higher the overlap degree is, the more severe the attenuation is. Thus, the chance of the correct candidate box to be filtered out is reduced. The resulting improved Soft-NMS score reset function is written as follows:

\[
S_i = \begin{cases} 
S_i, & \text{IOU (M, bi)} < N_t, \\
S_i(1 - \text{IOU (M, bi)}), & \text{IOU (M, bi)} \geq N_t.
\end{cases}
\]

Equation (4) is linearly weighted and has the possibility of mutation. Thus, a continuous Gaussian weighted score reset function is proposed to solve the situation where the degree of overlap exceeds the threshold when the mutation \( N_t \) causes the detection result to change, as shown in the following:

\[
S_i = S_i e^{-\frac{(\text{IOU (M, bi)})^2}{\sigma}}, \quad \forall bi \notin D. 
\]

On the standard datasets PASCAL VOC 2016 and MS-COCO 2018, the experiments using the object detection algorithm added to Soft-NMS determine that the algorithm performance can be considerably improved in cases where the objects overlap.

3.4. Sample Distribution Imbalance Improvement. For the one-stage object detection algorithm, the input picture is divided into several grids. Several anchor boxes are generated for each grid. Thus, the object detection algorithm contains a large number of anchor boxes. For the YOLO V4, each image contains 10647 anchor boxes. In the end, only three or four anchor boxes contain the object correctly. The anchor boxes that correctly contain the object are positive samples, whereas the other anchor boxes are negative samples. This scenario leads to an imbalance of positive and negative samples when the object detection algorithm is trained. This problem can be solved using a new weighting method to control the loss calculation scheme of positive and negative samples. Focal loss is combined with the
CIOU_Loss algorithm to introduce the MYOLO-lite loss function of the object detection algorithm.

3.4.1. CIOU Loss Function. In YOLO V4, the use of CIOU_Loss algorithms for the calculation of the object detection loss function is proposed, and the CIOU formula is defined as

\[
CIOU = IOU - \frac{\rho^2(b, b^{gt})}{l^2} - \alpha \nu, \tag{5}
\]

where \(\rho^2(b, b^{gt})\) is the European distance between the center point of the prediction box and the real box, and \(l\) is the diagonal distance of the minimum closure area containing both the prediction box and the real box. \(\alpha\) is a positive trade-off parameter, and \(\nu\) is used to measure the consistency.

Figure 3: Schematic diagrams of the backbone structure of MYOLO-lite series. (a) Schematic diagram of MV1-YOLO-lite structure. (b) Schematic diagram of MV2-YOLO-lite structure. (c) Schematic diagram of MV3-YOLO-lite structure.
of the aspect ratio between the prediction box and the real box. $\alpha$ and $\nu$ are defined as

$$\alpha = \frac{\nu}{1 - IOU + \nu}$$

$$\nu = \frac{4}{\pi} \left( \frac{\omega_{gt}}{h_{gt}} - \frac{\omega}{h} \right)^2,$$  \tag{6}

The CIOU Loss is defined as

$$L_{CIOU} = 1 - IOU + \frac{\rho^2(b, b'_{gt})}{\rho^2} + \alpha \nu.$$  \tag{7}

CIOU achieves fast convergence by minimizing the normalized distance between the prediction box and the object box. It has no deformation of the scale, making

\begin{table}
<table>
<thead>
<tr>
<th>Object detection model</th>
<th>Number of parameters before modifying PANet</th>
<th>Number of parameters after modifying PANet</th>
</tr>
</thead>
<tbody>
<tr>
<td>MV1-YOLO-lite</td>
<td>41005757</td>
<td>12754109</td>
</tr>
<tr>
<td>MV2-YOLO-lite</td>
<td>39124541</td>
<td>10872893</td>
</tr>
<tr>
<td>MV3-YOLO-lite</td>
<td>40043389</td>
<td>11791741</td>
</tr>
</tbody>
</table>
\end{table}
the regression much accurate and fast when overlapping or even containing the object box.

3.4.2. Focal Loss Improved Loss Function. Focal loss combines two methods to control the balance of positive and negative samples, namely, controlling the weights of positive and negative samples and controlling the weights of easy-to-classify and hard-to-classify samples.

Controlling the weights of positive and negative samples: the traditional two-classification cross-entropy loss function is defined as

\[
CE(p, y) = \begin{cases} -\log p, & \text{if } y = 1, \\ -\log(1 - p), & \text{otherwise}, \end{cases}
\]  
(8)

Focal loss adds a coefficient before the conventional loss function \(\alpha_t\) to solve the problem of imbalance between positive and negative samples and reduce the effect of negative samples.

\[
CE(p_t, y) = -\alpha_t \log(p_t),
\]

\[\alpha_t = \begin{cases} \alpha, & \text{if } y = 1, \\ 1 - \alpha, & \text{otherwise}, \end{cases}
\]

\[p_t = \begin{cases} p, & \text{if } y = 1, \\ 1 - p, & \text{otherwise}. \end{cases}
\]  
(9)

When the label is 1, \(\alpha_t = \alpha\). When the label is 0, \(\alpha_t = 1 - \alpha\). The range of the \(\alpha\) value is 0–1. We can control the effect of the number of positive and negative samples on loss by setting the value.

It is introduced in (10) to be written as follows:

\[
CE(p, y, \alpha) = \begin{cases} -\log p \cdot \alpha, & \text{if } y = 1, \\ -\log(1 - p) \cdot (1 - \alpha), & 0, \end{cases}
\]  
(10)

where \(y\) is the actual label, \(p\) is the predicted probability value, and \(\alpha\) is a coefficient inversely proportional to the predicted object probability. For a small number of positive samples, its coefficients are large, increasing the contribution to the model. For a large number of negative samples, its coefficients are small, attenuating their contribution to the model. This approach is used to adjust the weights of positive and negative samples to allow the model to learn useful information.

Controlling the weights of easy-to-classify and hard-to-classify samples: for the problem of uneven samples, modulating factor is introduced to solve \((1 - p_t)^Y\). The function can be written as

\[
FL(p_t) = -(1 - p_t)^Y \log(p_t).
\]  
(11)

The object detection network can correctly and easily identify the sample, which is the easy sample, when the input sample is equal to a positive sample, \(p_t = p\), or when \(p\) is close to 1. Moreover, the modulating factor becomes close to 0. When \(p\) is close to 0, the object detection network does not correctly identify the sample, which is the difficult sample. Moreover, the modulating factor becomes close to 1. This approach controls the weights of easy-to-classify and hard-to-classify samples.

The formula for calculating focal loss is written as follows by combining the two methods of heavy control:

\[
FL(p_t) = -(1 - p_t)^Y \log(p_t).
\]  
(12)

Improved loss function: the loss function of the YOLO object detection algorithm consists of bounding box loss, confidence loss, and classified loss. The loss function is written as follows:

\[
Loss = bbox\_loss + conf\_loss + class\_loss
\]

\[
= \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \sum_{i=j=0}^{B} I_{ij}^{obj} \left( (x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2 \right)
\]

\[
+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left( (\sqrt{w_i} - \sqrt{\tilde{w}_i})^2 + (\sqrt{h_i} - \sqrt{\tilde{h}_i})^2 \right)
\]

\[
- \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[ C_i^{obj} \log(C_i) + (1 - C_i^{obj}) \log(1 - C_i) \right]
\]

\[
- \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{noobj} \left[ C_i^{obj} \log(C_i) + (1 - C_i^{obj}) \log(1 - C_i) \right]
\]

\[
- \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[ \tilde{p}_i^{obj} \log(p_i) + (1 - \tilde{p}_i^{obj}) \log(1 - p_i) \right].
\]  
(13)

where \(bbox\_loss\) is the bounding box loss, consisting of lines 1 and 2 of the formula. The mean square error is used. \(conf\_loss\) is the confidence loss, consisting of lines 3 and 4 of the formula. \(class\_loss\) is the classification loss, consisting of the last line. The category and confidence level use cross-entropy as a loss function. Based on the idea of focal loss, the CIOU_Loss algorithm is introduced into the loss function. The CIOU loss is used as the loss of the object box regression. For confidence loss, focal loss is used to replace cross-entropy loss. Classify loss still uses cross-entropy loss, as written in

\[
CE(c, \tilde{c}) = -\tilde{c} \log(c) - (1 - \tilde{c}) \log(1 - c),
\]  
(14)

where \(\tilde{c}\) is the true category and \(c\) is the forecast category.

The improved algorithmic loss function CFLoss is written in

\[
CFLoss = \sum L_{CIOU} + \sum FL(p) + \sum CE(c).
\]  
(15)

The loss function of the MYOLO-lite algorithm uses CIOU as the loss of the object box coordinate regression added to focal loss, thereby alleviating the impact of sample imbalance on the detection algorithm and effectively improving the detection accuracy of the algorithm.

4. Experiments and Results

In this study, the idea of the MobileNet series network is used for reference, and it is fused with the YOLO V4 network. MobileNet series network replaces CSPDarkNet-53 as the YOLO V4 backbone network. Depthwise separable convolution is used to reconstruct the network structure of the enhanced feature extraction. MYOLO-lite series networks are designed to achieve object detection lightweight
4.1. Experimental Environment and Parameter Settings. The hardware setup in the laboratory configured for this study is as follows: processor, Intel Core i5-9400F CPU @ 2.90 GHz; graphics card, NVIDIA GeForce GTX 1660; memory, 8 GB; hard disk, Samsung SSD 860 EVO500 GB; software environment, Windows 10; programming language, Python 3.6.7; and deep learning frameworks and dependency libraries, Anaconda 4.3.21, TensorFlow 1.14, CUDA 10.0, CU DNN7.4.

The training parameters were critical to model performance. The experiment adopted a cosine annealing learning rate strategy. The batch size, total number of iterations, and initial learning rate were set to 6, 100, and 0.0001, respectively. After a linear rise, a maximum learning rate of 0.001 was reached. Then, it descended to a 0.000001 stop through cosine annealing.

The limited data samples were considered in this study. The idea of transfer learning was used in this study to reduce the network training time and avoid network overfitting or underfitting. The model pretrained by MobileNet was used on the VOC2007 public dataset as the model of this study. The model was pretrained through the method of pretraining weights shown in Figure 6.

4.2. Constructing the Object Dataset for Forward-Moving Trains. In this study, a dataset was gathered by collecting visual videos of trains moving forward and converting the videos into frame pictures [32]. Different scenes of the train were selected as many as possible to improve the generalization ability of the model, and all the states of the object in the video screen were covered as comprehensively as possible. Finally, 3722 images were obtained as the dataset, and 70% of the dataset was randomly selected at training time for the training set; 30% of the dataset was selected as the test set. The categories of the dataset can be divided into four categories: tracks, signals, tunnels, and trains. The number of various objects in the training set and test set is shown in Table 2.

The dataset was enhanced to improve the robustness of the network by reducing the influence of additional factors on the recognition. After the transmission to the object detection network, effective data enhancement of the dataset requires not only the original image brightness, size, contrast ratio, and other parameters to be changed but also the bounding box of the images to be adjusted to correspond to the original image position. When scaling the original image, in order to prevent the size of the input network image from changing, it is necessary to add a gray bar to the edge of the image with the changed size, so that the enhanced image is consistent with the size of the input detection network. In this way, the network can also improve the detection ability of targets of different sizes. The original image is shown in Figure 7 [33]. Figure 8 shows the image after data enhancement.

4.3. Algorithm Evaluation. This study uses average precision (AP), mean average precision (mAP), and frames per second (FPS) for detection speed to evaluate the performance of the algorithm model.

First, the union ratio (i.e., IOU) must be calculated; in particular, the overlapping area between the prediction box and the real box is compared with the total area occupied by the prediction box and the real box. The threshold for the cross-to-side ratio was set to 0.5. This setting was evaluated in this study. At the time, it was considered a positive sample; at the time, it was considered a negative sample. $IOU \geq 0.5IOU < 0.5$.

Precision represents the ratio of true positives in the identified images. Recall is the proportion of all parts of the test set that are correctly identified as positive samples to all parts that are indeed positive samples. AP is the probability that the positive sample identified by the model accounts for all samples, and it can be used to measure the accuracy of the model’s detection of a class of objects. mAP is the average of AP values of all categories. The dataset used in this study has four categories. For the same test video, the number of frames processed per second (i.e., FPS) is the detection speed evaluation standard under the same training model in the incoming object detection network.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Mean Average Precision} = \frac{\sum \text{Accuracy}}{N(\text{class})},$$

where TP is correctly identified as a positive sample. The TN is correctly identified as a negative sample. The FP is negative
samples incorrectly classified as positive samples. The FN is the positive sample incorrectly classified as a negative sample.

4.4. Experimental Analysis

4.4.1. MYOLO-Lite Series Algorithm Analysis. In this study, experiments were carried out according to the MYOLO-lite series target detection model constructed, and training was conducted on the self-made target dataset in front of the train. After 100 batches of training, 90% of 2605 images in the training set were selected for training after data enhancement, a total of 2345 images were selected for training, and 10% of 245 images were selected for verification. Loss function uses the loss function of YOLO V4. The training weights and loss functions of each batch of training during the training were saved, and the curve of loss value was extracted after 100 batches of training (Figure 9). Figure 9 shows that the loss value changes considerably in the first 20 batches. The loss value gradually stabilizes after reaching the 80th batch.

Figure 10 shows P-R (precision-recall) curves of MYOLO-lite target detection models for four types of targets: track, signal, tunnel, and train. The accuracy and recall rate of the P-R curve are plotted on the Y-axis and X-axis, where the area enclosed by the curve and the coordinate axis is the AP value of the class.

mAP and classification results of all kinds of target prediction of MYOLO-lite series algorithms on the test set are shown in Figure 11.

4.4.2. Comparative Analysis of Various Algorithm Types. In this study, the Faster R-CNN and SSD were compared to verify the performance of the improved detection algorithm. YOLO V3, YOLO V4, and the lightweight YOLO V4-tiny algorithm were also analyzed. The training weight with the lowest loss zvalue was used as the training result, and the test set was tested by loading the weights to obtain the AP values of the four objects in each detection model. The experimental results are shown in Table 3.

Table 2: Number of objects in the training and test sets.

<table>
<thead>
<tr>
<th>Detected object name</th>
<th>Number of objects in the training set</th>
<th>Number of objects in the test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track</td>
<td>3957</td>
<td>1694</td>
</tr>
<tr>
<td>Signal</td>
<td>1184</td>
<td>506</td>
</tr>
<tr>
<td>Tunnel</td>
<td>817</td>
<td>346</td>
</tr>
<tr>
<td>Train</td>
<td>348</td>
<td>153</td>
</tr>
</tbody>
</table>

Figure 6: Pretraining model method.

Figure 7: Original image.
Table 4 shows that as a typical algorithm for the two-stage class, Faster R-CNN obtains a high accuracy rate. However, its training takes approximately four times as long as the training for a one-stage class. The FPS value is only more than one-tenth of the one-stage class. Processing a picture requires 0.8 s, which cannot meet the real-time requirements in practical applications. In the algorithm of the one-stage class, SSD has the fastest detection speed. However, its low-level features have few convolutional layers, insufficient feature extraction, poor detection effect, and low mAP value. YOLO series algorithm can effectively balance the relationship between speed and accuracy. Lightweight network YOLO V4-tiny uses SPDarkNet-53-tiny as the backbone network, reducing the number of network layers and simplifying the strengthening of feature extraction network structure. Compared with YOLO V4, YOLO V4-tiny exhibits detection speed with a good improvement, but its detection accuracy decreases.

Compared with the YOLO V4 algorithm, the MYOLO-lite series algorithm designed in this study exhibits approximately 3% improvement in mAP, and the AP values of track, signal, tunnel, and train are improved to a certain

![Figure 8: Image after data enhancement. (a) Magnification and brighten, (b) magnification and mirror, (c) magnification and darken, (d) image reduction and translation, (e) image reduction and translation, and (f) mirror, reduction, and translation.](image)

![Figure 9: Loss function graph.](image)
extent, verifying the feasibility of improvement. The training time of MYOLO-lite series algorithms is less than that of YOLO V4. MYOLO-lite series algorithms are more than three times faster than YOLO V4. The use of MobileNet V3 as the backbone network achieves the best results, with the highest detection speed and detection accuracy. The mAP value reaches 95.74%, the FPS is 42.04, and picture detection takes only 0.024 s.

Figure 12 compares the effects detected by the algorithms to show the detection effect of the model intuitively.

Figure 12 shows that Faster R-CNN, SSD, YOLO V3, and YOLO V4 have a certain degree of missed detection. The MYOLO-lite series network has basically no missed detection, its size and position of detection boxes are more standard, and its detection effect is the best.

4.4.3. Ablation Experiment. Comparison was made between YOLO V4, lightweight YOLO V4, MM3-YOLO-lite, and MM3-Yolo-lite using Soft-NMS, and focal loss was introduced and combined with the CIou_Loss regression loss function. The AP values of four types of targets, mAP values, and FPS values in each model are obtained through experimental analysis. The experimental results are shown in Table 5.

Table 5 shows that compared with YOLO V4, the detection speed of the lightweight YOLO V4 is greatly improved, but the detection accuracy is somewhat decreased. The mAP value of MV3-YOLO V4-lite is 3.3% higher than that of YOLO V4, and the AP values of tunnel of MV3-YOLO V4-lite are significantly improved. The MV3-YOLO-lite of Soft-NMS + CIou_Loss replaces traditional nonmaximally suppressed NMS with Soft-NMS, and focal
loss was introduced and combined with the CIoU_Loss regression loss function. Compared with the MV3-YOLO-lite algorithm, the MV3-YOLO-lite of Soft-NMS + CIoU_Loss algorithm obtains a 0.32% improvement in mAP. For signals and trains, the optimized algorithm can effectively optimize the interference situation of being blocked and the problem of small sample size. Moreover, the AP value is improved to a certain extent, and the detection speed is approximately 42 FPS. In general, it has little impact on the detection speed.
Figure 12: Continued.
NMS + CIOU_Loss algorithm in this study can effectively improve the detection performance of the object detection algorithm and still meet the requirements of real-time detection.

mAP of the MV3-YOLO-lite of Soft-NMS + CIOU_Loss algorithm on the test set and the prediction results of various targets are shown in Figure 13.

Figure 12: Comparison of the effects of object detection algorithms. (a) Original image. (b) Faster R-CNN. (c) SSD. (d) YOLO V3. (e) YOLO V4. (f) MV1-YOLO-lite. (g) MV2-YOLO-lite. (h) MV3-YOLO-lite.

Table 5: Ablation results.

<table>
<thead>
<tr>
<th>Object detection algorithm</th>
<th>Track (%)</th>
<th>Signal (%)</th>
<th>Tunnel (%)</th>
<th>Train (%)</th>
<th>mAP (%)</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLO V4</td>
<td>97.99</td>
<td>89.78</td>
<td>89.03</td>
<td>92.77</td>
<td>92.40</td>
<td>12.8</td>
</tr>
<tr>
<td>YOLO V4-tiny</td>
<td>97.58</td>
<td>86.34</td>
<td>87.29</td>
<td>90.16</td>
<td>90.34</td>
<td>21.37</td>
</tr>
<tr>
<td>MV3-YOLO-lite (NMS)</td>
<td>99.81</td>
<td>91.59</td>
<td>96.26</td>
<td>95.30</td>
<td>95.74</td>
<td>42.04</td>
</tr>
<tr>
<td>Soft-NMS MV3-YOLO-lite</td>
<td>99.70</td>
<td>92.79</td>
<td>96.04</td>
<td>95.71</td>
<td>96.06</td>
<td>41.73</td>
</tr>
</tbody>
</table>

mAP = 96.06%

Figure 13: The prediction results of the MV3-YOLO-lite of Soft-NMS model. (a) mAP values and (b) the prediction results.
Images were used to test the model and the effect of the algorithm. Part of the detection effect is shown in Figure 14.

5. Conclusions

A self-made visual dataset of a forward-moving train is introduced, and an improved MYOLO-lite algorithm is proposed. The previous object price test model does not meet the real-time requirements in terms of detection speed. Thus, an improved MYOLO-lite model based on YOLO V4 is proposed. Redesigning the a priori box with K-means clustering ideas enhances scale adaptability. Then, the proposed MYOLO-lite method is optimized to detect the obstruction and coincidence of large objects, such as occluded tracks in front of the train. The Soft-NMS is suppressed by slightly improving the flexibility. The inhibition of the prediction box to the adjacent prediction box is reduced. The generation process of the detection box is optimized, and the cross-entropy is redesigned by integrating focal loss and CIOU_Loss algorithm for the problems of

![Figure 14: Image comparison of ablation experiments. (a) Original image. (b) YOLO V4. (c) MV3-YOLO-lite (NMS). (d) Soft-NMS MV3-YOLO-lite.](image-url)
positive and negative samples and difficult sample imbalance in the dataset loss function. Therefore, the samples are balanced by adjusting the weight parameters, thereby increasing the positioning accuracy of the object.

Data Availability

The data used to support this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This research was sponsored by the National Natural Science Foundation of China (62103074) and scientific research project of Liaoning Provincial Department of Education (JDL2020002).

References

[20] Yue Sun, Design of Train Positioning System for Non-electrified Section of Railway Special Line Based on Video, Beijing Jiaotong University, Beijing, China, 2021.


