JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 18, No. 2, June 2024, pp. 69-79

Detection Algorithm of Tram Track Obstacles Based on Improved-SSD

Yunming Wang*, Yiang Zhou, Xianwu Chu, and Guodu Peng
School of Automation and Electrical Engineering, Dalian Jiaotong University, Dalian, China
wang19871128@126.com, 15524729016@163.com, Chuxw@djtu.edu.cn, 815078315@qq.com

Abstract
The accurate and rapid identification of tram track obstacles is a crucial aspect in improving the safety of urban tram driving. To improve the detection accuracy and detection speed of urban tram track obstacles, the current study proposes an urban tram track obstacle detection algorithm based on Improved-SSD. To this end, for Conv3_3, Conv4_3, and Conv5_3, a bidirectional fusion module is designed to strengthen the feature expression ability of the low-level feature layer and enrich the semantic information. Meanwhile, for Fc7, Conv6_2, Conv7_2, Conv8_2, and Conv9_2, a two-stage deconvolution module is devised to compensate for the lack of detailed information of the high-level feature layer. To improve the detection speed, the convolution split structure is designed to replace all 3×3 convolutions in the backbone network VGG16. Then, to improve the model's ability to match a specific dataset, the k-means algorithm is used to optimize the aspect ratio of the prior bounding box. Finally, the improved algorithm is trained with and tested using a self-made dataset. The experimental results show that, compared to the traditional SSD, the mean average precision of the Improved-SSD algorithm in detecting track obstacles is increased by 1.09%. The detection speed is also increased by 0.9 FPS. Lastly, the prediction box matches the real obstacle box better than that of the traditional SSD.

Category: Computer Graphics / Image Processing

Keywords: Tram; Obstacle detection; Deep learning; SSD; Prior bounding box

I. INTRODUCTION

Many large and medium-sized cities around the world are building new urban public transportation systems where rail transit serves as the foundation supporting conventional modes of public transport. Urban trams, which have the characteristics of large capacity, fast speed, and environmental protection, have emerged as an important part of such rail transit. The tracks of such trams occupy the same road space as other urban vehicles. Urban tram tracks are often occupied by pedestrians, various types of vehicles, animals, and other obstacles. Tram drivers are typically not able to observe obstacles on tracks ahead of them in an accurate and timely manner due to bad weather, blind spots, distractions, and other factors, ultimately affecting passenger safety and even leading to traffic accidents [1]. The rapid developments in artificial intelligence [2] and deep learning technology that have been achieved in recent years have effectively improved the accuracy of target detection. Therefore, there has been substantial interest in the use of deep learning technology in track obstacle detection research for trams.

The existing object detection algorithms [3] based on

Open Access http://dx.doi.org/10.5626/JCSE.2024.18.2.69 http://jcse.kiise.org

Received 22 April 2024; **Accepted** 10 June 2024
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

deep learning can mainly be divided into two categories: the first category is the two-stage object detection algorithm. This algorithm forms a sample candidate box, uses the convolutional neural network (CNN) [4, 5] to classify the sample, and finally corrects the position of the bounding box. Typical two-stage object detection algorithms include regions with CNN (RCNN) [6], Fast RCNN [7], Faster RCNN [8]. These algorithms exhibit high detection accuracy but slow detection speed, so they cannot meet the requirements of real-time detection. Aiming to solve the problem of a large number of learning parameters used by CNN networks, one study [9] used a hybrid expansion CNN instead of the VGG16 network in Faster RCNN to achieve object detection with improved accuracy and efficiency. Another study [10] proposed an underwater object detection model based on improved Faster RCNN, which replaces the VGG16 network with the Res2Net101 network to enhance the expression ability of the receptive fields at each network layer. The online hard example mining (OHEM) algorithm was also introduced to solve the imbalance problem regarding the positive and negative samples of the bounding box.

The second type is a single-stage object detection algorithm based on regression, which mainly comprises YOLO [11-13] and SSD [14]. The advantage of this type of algorithm is its high detection speed. One prior study [15] designed the feature extraction network DB-DarkNet-53 based on the YOLOv3 algorithm. This network achieved an average detection accuracy of 83.5% on the PASCAL VOC dataset along with a detection speed of 35.8 frames per second (FPS), which represent improvements in both the accuracy and speed of object detection. Another study [16] proposed the FFC-SSD model with the aim of improving the accuracy of the SSD algorithm in detecting small-sized targets. This model adopted the method of group clustering to obtain the prior bounding box parameters that are more in line with the target sample size. To enhance the feature extraction capability of small targets, a depooling efficient multi-scale feature fusion (MSFF) module was also designed. The authors of another work [17] combined the idea of the feature pyramid network and the use of a feature layer containing more semantic information in MobileNetV2 SSD for fusion; the results showed that it successfully regenerated the feature pyramid and that the mean average precision (mAP) reached 76.5%. A different study [18] optimized the number ratio and aspect ratio of the SSD algorithm by combining the practical application scenarios of vehicle detection, which had the effects of improving both the bounding box regression speed and the detection accuracy.

Therefore, improving the detection model and optimizing the generation method of prior bounding box is an effective way to improve the accuracy of object detection.

To enhance the accuracy and real-time detection of track obstacles, the current work proposes an urban tram track obstacle detection algorithm based on Improved-SSD. The low-level feature layer is designed to have a bidirectional fusion module to enrich the semantic information of the low-level feature map. The high-level feature layer is designed to have a two-stage deconvolution module to increase the amount of edge information. To improve the detection speed of the model, the convolution split structure is adopted to replace all 3×3 convolution kernels in VGG16. The k-means [19] algorithm is used to optimize the generation method of the aspect ratio of the prior bounding box to improve the matching ability of the model with a specific dataset. The track obstacle dataset is self-made through field shooting and data preprocessing, and the performance of the improved algorithm in detecting track obstacles is verified on this dataset.

II. DETECTION ALGORITHM OF TRAM TRACK OBSTACLE BASED ON IMPROVED-SSD

Urban trams can smoothly run on urban roads that are unobstructed, but there are often track obstacles such as pedestrians, vehicles, and animals, which seriously affect the safety and efficiency of tram driving. To improve the detection accuracy and detection speed of urban tram

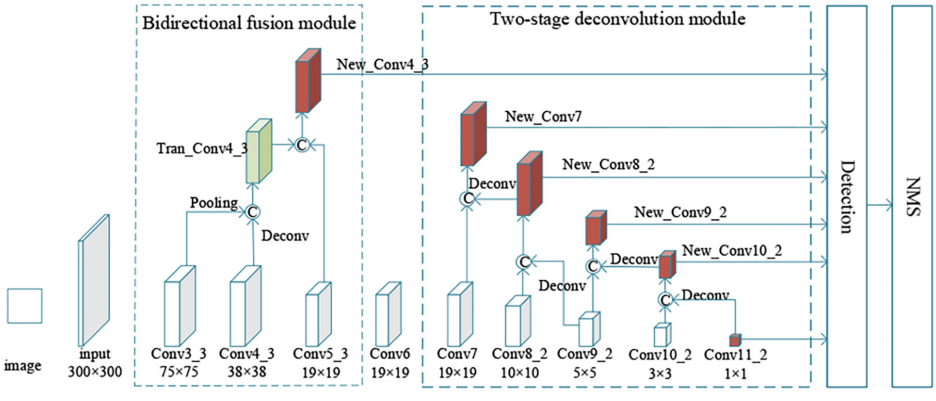


Fig. 1. Network structure of Improved-SSD.

track obstacles, the current study proposes Improved-SSD. The network structure of Improved SSD is shown in Fig. 1. C represents the channel cascade fusion of the feature map.

The Improved-SSD network structure is based on VGG16. To improve its ability to detect small obstacles, Improved-SSD utilizes Conv3_3, Conv4_3, and Conv5_3 design bidirectional fusion modules as replacements for the original Conv4_3. To make up for the excessive loss of detailed information in the high-level feature layer, Improved-SSD uses Fc7, Conv6_2, Conv7_2, Conv8_2, and Conv9_2 to build a two-stage deconvolution module. Improved-SSD also uses a convolutional split structure to replace all 3×3 convolution kernels in VGG16, with the ultimate aim of increasing the detection speed of the improved network.

A. Bidirectional Fusion Module

Traditional SSD uses Conv4_3 to detect the feature layer of small targets, which increases the chance of detection. To strengthen the semantic information of Conv4_3, the system proposed in this paper combines pooling and deconvolution to better extract the edge information and semantic information of the upper and lower feature layers.

Conventional pooling and deconvolution are one-way operations, and feature maps cannot make full use of feature information in other directions. The resolution sizes of the feature maps that are output by Conv3_3, Conv4_3 and Conv5_3 are 75×75, 38×38, and 19×19, respectively, which are sufficiently large to retain most of the semantic information of small targets. Therefore, this paper uses these three feature layers for pooling fusion and deconvolution fusion to generate New_Conv4_3 as the detection prediction layer for small targets. The bidirectional deconvolution module is shown in Fig. 2.

The implementation steps of the bidirectional fusion module are as follows:

- Pooling fusion: Conv3_3 performs 2×2 pooling operations. The obtained result performs 1×1 convolution, then fuses with Conv4_3 according to the cascade of the number of channels to obtain Tran_Conv4_3.
- Deconvolution fusion: Conv5_3 performs deconvolution to obtain De_Conv5_3. De_Conv5_3 and Tran_Conv4_3 are integrated in a cascade according to the number of channels. The result of the fusion performs 1×1 convolution to obtain New_Conv4_3. The size of New_Conv4_3 and the size of the original Conv4_3 are both 38×38×512.

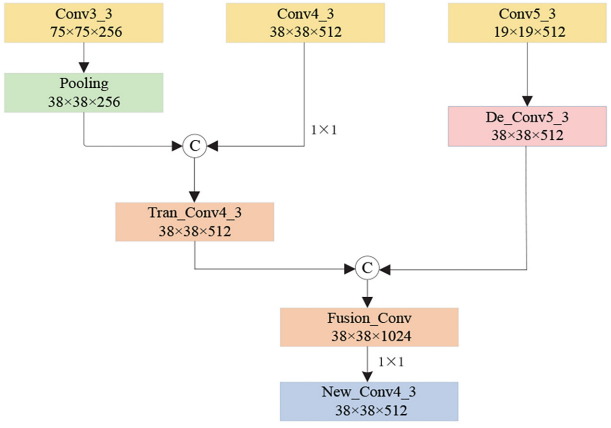


Fig. 2. Bidirectional fusion module.

The bidirectional fusion module contains both positive pooling fusion and reverse deconvolution fusion. This structure enriches the semantic information of the New_Conv4_3 and improves the accuracy of the algorithm in terms of detecting small target obstacles in track.

B. Two-Stage Deconvolution Module

To recover the edge information lost by the high-level feature layer, it is necessary for the high-level feature layers to perform deconvolution operations. However, if the five high-level feature layers perform continuous deconvolution operations, a large amount of noise information will be introduced and the detection accuracy will be reduced. Therefore, a two-stage deconvolution module is designed to divide the deconvolution operation of the high-level feature layer into two stages; this two-stage deconvolution module structure is shown in Fig. 3.

First, Conv9_2 performs a deconvolution operation to obtain a De_Conv9_2 with the same dimensions and number of channels as Conv8_2. Next, Conv8_2 and De_Conv9_2 are converged in a cascading manner based on the number of channels. The result of the fusion performs 1×1 convolution operation to obtain the new feature layer New_Conv8_2. Then, New_Conv8_2 performs a deconvolution operation to obtain a De_New_Conv8_2 with the same dimensions and number of channels as Conv7_2. Finally, Conv7_2 and De_New_Conv8_2 are fused together by cascading the number of channels. The result of the fusion finally performs 1×1 convolution to obtain New_Conv7_2.

Conv7_2 performs a deconvolution operation to obtain a De_Conv7_2 with the same dimensions and number of channels as Conv6_2. Next, Conv6_2 and De_Conv7_2 are converged in a cascading manner based on the number of channels. The result of the fusion performs 1×1 convolution operation to obtain the new feature layer New_Conv6_2. Then, New_Conv6_2 performs a deconvolution operation to obtain a De_New_Conv6_2 with the same dimensions and number of channels as FC7. Finally, FC7 and De_New_Conv6_2 are fused together by cascading the number of channels. The result

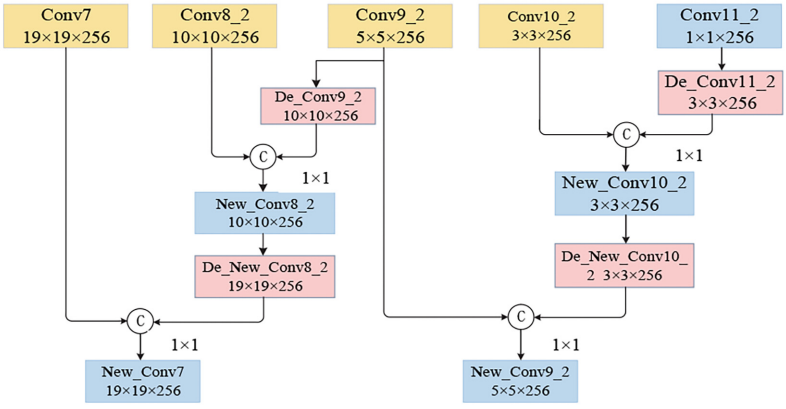


Fig. 3. Two-stage deconvolution process.

of the fusion finally performs 1×1 convolution to obtain New_FC7.

After two-stage deconvolution, New_Fc7, New_Conv6_2, New_Conv7_2, and New_Conv8_2 are newly generated.

C. Convolutional Split Structure

The two-step improvement of the SSD algorithm presented above increases the number of network parameters and reduces the training and detection speeds of the network. This paper adopts a convolutional splitting structure to speed up the detection speed of the network.

The core idea of the convolutional splitting structure is to divide the traditional convolution process into depthwise convolution and pointwise convolution: depthwise convolution is used for each channel whereas pointwise convolution is used for channel transformation. The convolution splitting structure is shown in Fig. 4.

In traditional convolution, assume that the size of the input feature graph F is $S_F \times S_F \times M$ and the size of the standard convolution kernel K is $S_K \times S_K \times M \times N$. S_F is the width and height of the input feature graph F; S_K is the width and height of the standard convolution kernel K; M is the number of channels; and N is the number of convolution kernels. Assuming that the step size is 1, the size of the output feature map C is $S_C \times S_C \times N$, and S_C is

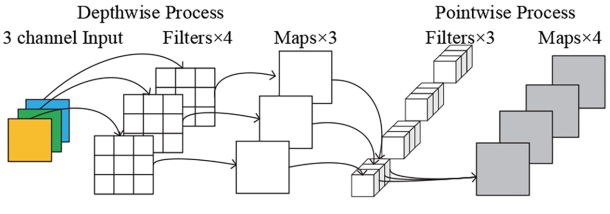


Fig. 4. Schematic diagram of convolution splitting structure.

the width and height of the output feature map C. Therefore, C can be calculated as follows using Eq. (1):

$$C_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \times F_{k+i-1,j+l-1,m} . \quad (1)$$

In Eq. (1), $C_{k,l,n}$ represents the kth row and the l_{th} column of the nth channel of $k = 1, 2, \ldots, S_C$, $j = 1, 2, \ldots, S_C$, $n = 1, 2, \ldots, N$. Meanwhile, $K_{i,j,m,n}$ represents the i_{th} row and j_{th} column of the m_{th} channel in the nth convolution kernel of K, $i = 1, 2, \ldots, S_K$, $j = 1, 2, \ldots, S_K$, $m = 1, 2, \ldots, M$.

The computational amount of traditional convolution is obtained as shown in Eq. (2):

$$Q_1 = S_K \times S_K \times S_M \times S_M \times S_F \times S_F . \quad (2)$$

In the convolutional split structure, the traditional convolution kernel K is first replaced by a deep convolution kernel K^d with a size of $S_K \times S_K \times 1 \times M$ and a convolution kernel with a size of $1 \times 1 \times M \times N$. The size of the feature map C^d obtained by depthwise convolution is $S_G \times S_G \times M$. The method used to calculate C^d is shown in Eq. (3):

$$C^d_{k,l,n} = \sum_{i,j,m} K^d_{i,j,m} \times F_{k+i-1,j+l-1,m} . \quad (3)$$

Then, pointwise convolution is performed on the feature map C^d, and the size of the feature map is $S_G \times S_G \times N$.

The calculation amount of the convolutional split structure is the sum of depthwise convolution and pointwise convolution. The calculation method is shown in Eq. (4):

$$Q_2 = S_K \times S_K \times M \times S_F \times S_F + M \times N \times S_F \times S_F . \quad (4)$$

The process used to calculate the ratio of the convolution splitting structure and the computational quantity of traditional convolution is as shown in Eq. (5):

$$\frac{Q_2}{Q_1}=\frac{S_K\times S_K\times M\times S_F\times S_F+M\times N\times S_F\times S_F}{S_K\times S_K\times M\times N\times S_F\times S_F}=\frac{1}{N}+\frac{1}{S_K^2} \quad (5)$$

The value of N in deep neural networks is typically large, and the size $S_K\times S_K$ of the convolution kernel is typically 3×3. Therefore, the ratio of the convolution split structure to the computational amount of traditional convolution is close to 1/9. The convolutional splitting structure greatly reduces the number of parameters of the model.

D. Prior Bounding Box Optimization

At present, there is no publicly available standard dataset for track obstacle detection, so a dataset has been made for this purpose in the current paper. To this end, this paper collected 2,000 pictures containing common obstacles of the following categories: animal, person, car, and motorcycle. Then, noise was added and mirrored for each image to increase the number of images in the dataset. This self-made dataset ultimately contained 4,000 images with 4,328 obstacle targets. In this paper, the area ratio of the target bounding box to the image is less than 0.03, as the focus is on small target obstacles. The self-made dataset contains 1,541 small target obstacles and 2,787 large target obstacles. The animal category consists entirely of small obstacles, while car category consists entirely of large obstacles. The person and motorcycle categories have both small and large obstacles.

When moving between different datasets, there can be large redundancy or deviation in the aspect ratio of the real frame and the prior bounding box, which affects the detection accuracy of the SSD algorithm.

In this paper, the image size of the self-made dataset is scaled to 300×300, and the width and height distributions of the four track obstacles are analyzed. The aspect ratio distributions of the four obstacles are shown in Fig. 5. The golden dots indicate the widths and heights occupied by the obstacle in a 300×300 size picture. Dots of the other colors represent prior bounding boxes of different sizes, and prior bounding boxes on the same line have the same aspect ratio.

In Fig. 5, it can be seen that the distribution of the original set prior bounding box and the target real box of the self-made dataset are very different. The generation method of the original prior bounding box aspect ratio will have a large impact on the detection effect of the SSD algorithm.

The setting of the prior bounding box aspect ratio should change according to the change in the real box in the dataset. The more the prior bounding box matches the real box, the smaller the impact of redundant background noise on accuracy. At the same time, the smaller the difference between the prior bounding box and the real box, the easier the positional regression. Therefore, the prior bounding box settings should be changed according to the changes in the ground truth in different datasets.

To reduce the deviation between the real box and the prior bounding box of the self-made dataset, this paper uses the k-means algorithm to cluster and analyze the aspect ratio of all targets in the self-made dataset. The specific steps followed in this process are as follows:

This paper sets the abscissa of all real boxes in the data to long and the ordinate to wide to obtain the following two-dimensional clustering sample: $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$, $x^{(i)}\in R^n$.

1) Three initial cluster centers are randomly generated at: $\{u_1, u_2, u_3\}$.

2) Eq. (6) is used to calculate which cluster category each training sample belongs to:

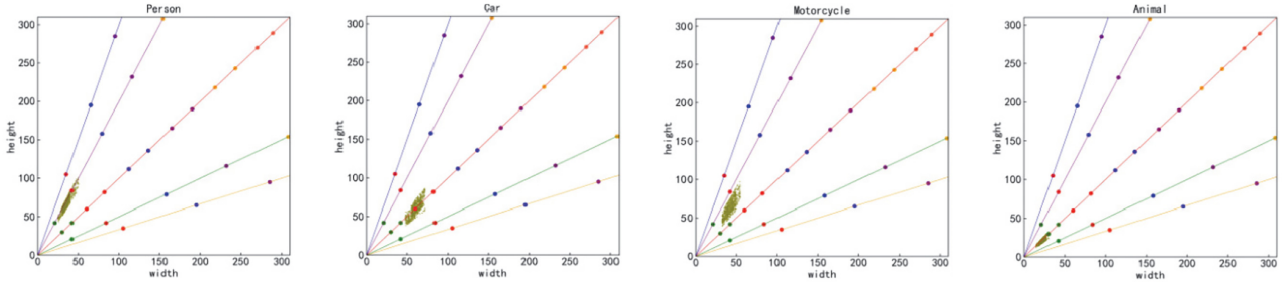
$$c^{(i)}=\min_j \left\|x^{(i)}-u_j\right\|^2. \quad (6)$$

In Eq. (6), j = 1, 2, 3.

3) The cluster centers for each class are recalculated as follows:

$$u_j=\frac{\sum_{i=1}^{m}\{C^{(i)}=j\}x^{(i)}}{\sum_{i=1}^{m}\{C^{(i)}=j\}}. \quad (7)$$

4) Steps 3 and 4 are repeated until reaching convergence to obtain three cluster centers. The widths and heights of



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

Fig. 5. Width and height distributions of four orbital obstacle targets: (a) person, (b) car, (c) motorcycle, and (d) animal.

Table 1. Aspect ratios of cluster centers

Width	Height	Aspect ratio
38	22	1.727
66	41	1.610
83	54	1.537

Table 2. Improved aspect ratio setting of the SSD algorithm's prior bounding box

Prediction layer	Size	Number
Conv4_3	21{1:2, 1:1, 2:1}	38×38×4
Fc7	45{1:2, 1:1.62, 1:1, 1.62:1, 2:1}	19×19×6
Conv6_2	99{1:2, 1:1.62, 1:1, 1.62:1, 2:1}	10×10×6
Conv7_2	153{1:2, 1:1.62, 1:1, 1.62:1, 2:1}	5×5×6
Conv8_2	207{1:2, 1:1, 2:1}	3×3×4
Conv9_2	261{1:2, 1:1, 2:1}	1×1×4

the three cluster centers are listed in Table 1.

In Table 1, it can be seen that the aspect ratios of the three cluster centers are 1.727, 1.610, and 1.537, respectively; the mean of the three aspect ratios is 1.62. The original 1:3 and 3:1 prior bounding boxes match the real target box the least. Therefore, the aspect ratio of 1:1.62 is used to replace the aspect ratio of 1:3. Meanwhile, the aspect ratio of 3:1 is replaced by the aspect ratio of 1.62:1.

The prior bounding box aspect ratio settings of each feature layer of the SSD algorithm are improved as presented in Table 2.

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Environment and Evaluation Indicators

This experiment is trained and tested on the deep learning framework of TensorFlow in Python 3.8.3.

During the process of training the network, the learning rate is typically used to control the update speed of network parameters. If the learning rate is set too small, the training efficiency will be reduced. However, if the learning rate is set too large, it will cause the parameters to hover around the optimal value, as a result of which the network will not reach the optimal convergence state. In this paper, the polynomial decay function is introduced to adjust the learning rate, and the formula used to calculate the learning rate is as shown in Eq. (8):

$$lr = lr_{base} \times \left(\frac{1-iter}{iter_{\max}}\right)^{power}. \quad (8)$$

In Eq. (8), lr_{base} represents the initial learning rate, which is set to 5×10^{-4}, $iter$ represents the number of iterations, $iter_{\max}$ represents the maximum number of iterations, and power represents the power exponent of the function. Finally, lr_{end} is set to 5×10^{-7}.

The evaluation indicators used in this experiment were average precision (AP) and FPS. AP is an evaluation index of the precision and recall of a certain category in the data sample. FPS represents the number of pictures processed per second.

Precision refers to the number of correctly detected items as a percentage of the total detected quantity, and the calculation process is shown in Eq. (9):

$$P = \frac{TP}{TP+FP}. \quad (9)$$

Recall refers to the percentage of the number of correctly detected items to all labeled quantities, and the formula for calculating recall is shown in Eq. (10):

$$R = \frac{TP}{TP+FN}. \quad (10)$$

In Eq. (10), TP represents the number of positive classes that are predicted as positive classes. FP represents the number of negative classes that are predicted as positive classes. FN represents the number of positive classes that are predicted as negative classes.

The detection results are used to plot the P-R curve with accuracy as the vertical axis and recall as the horizontal axis. The area of the P-R curve is AP, while the formula used to calculate AP is shown in Eq. (11):

$$AP = \int_0^1 P(R)dR. \quad (11)$$

For multi-category detection tasks, mAP is used to measure the detection performance of the model for each category. The formula used to calculate mAP is shown in Eq. (12):

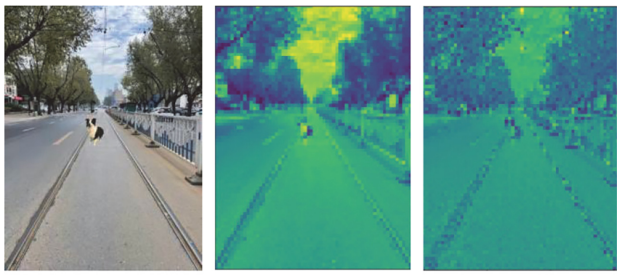
$$mAP = \frac{1}{n}\sum_{i=1}^{n} \mathrm{AP}_i. \quad (12)$$

B. Experimental Results and Analysis

To evaluate the performance of the Improved-SSD algorithm proposed in this paper in detecting tram track obstacles, the following experiments are carried out.

The visual feature plots of the SSD algorithm and the Improved-SSD algorithm in the Conv4_3 output are shown in Fig. 6.

In Fig. 6, the features output by the SSD algorithm in the Conv4_3 are blurred, and the features of the animal are gradually abstracted. The results show that there is a



(a) (b) (c)

Fig. 6. Characteristic diagram of Conv4_3: (a) original image, (b) SSD, and (c) improved-SSD.

(a) (b)

Fig. 7. Visualization results before and after prior box optimization: (a) SSD and (b) improved-SSD.

significant loss of spatial information after multi-layer convolution. The Improved-SSD algorithm has clearer profile features in the Conv4_3 output. This shows that the bidirectional fusion module can enrich the semantic information and spatial information of the low-level feature map, which is conducive to improving the detection accuracy of small obstacles.

After prior box optimization, the results of the SSD algorithm and Improved-SSD algorithm in detecting a person on the track are shown in Fig. 7.

In Fig. 7, the confidence level of the SSD algorithm for a person is 79%, while that of the Improved-SSD is 81%. The aspect ratio of the prediction box is 1:3 in the SSD model and 1:1.62 in the Improved-SSD model. The results show that, by optimizing the prior bounding box, the difference between the prior bounding box and the target is reduced, the background noise is reduced, and the confidence is improved.

The number of training times for the experiment is set to 30,000. Table 3 lists the performance pairs for ablation verification.

As can be seen in Table 4, the accuracy of detecting small obstacles in SSD+bidirectional fusion is 0.48% higher than that in the SSD algorithm. The results show that the bidirectional fusion module improves the detection

Table 3. Ablation experimental results

Ablation experiments	AP (%)				mAP (%)	FPS
	Person	Car	Motorcycle	Animal		
SSD	75.22	86.95	79.58	65.98	76.93	25.5
SSD+bidirectional fusion	75.51	87.01	79.82	66.46	77.20	20.1
SSD+two-stage deconvolution	75.40	87.36	79.81	66.03	77.15	18.3
SSD+convolutional split	75.07	86.84	79.33	65.84	76.77	36.4
SSD+optimized prior bounding box	75.16	86.92	79.46	65.82	76.84	26.2
Improved-SSD	76.99	87.54	80.88	67.39	78.20	26.4

FPS=frames per second.

Table 4. Performance of different detection algorithms on the overall dataset

Model	AP (%)				mAP (%)	FPS
	Person	Car	Motorcycle	Animal		
Faster RCNN	76.53	87.74	80.36	66.65	77.82	12.5
YOLOv3	76.27	88.16	80.23	65.06	77.43	32.6
SSD	75.22	86.95	79.56	65.98	77.03	25.5
FSSD	75.69	86.96	79.87	66.76	77.32	22.1
Improved-SSD	76.99	87.54	88.98	67.39	78.20	26.4

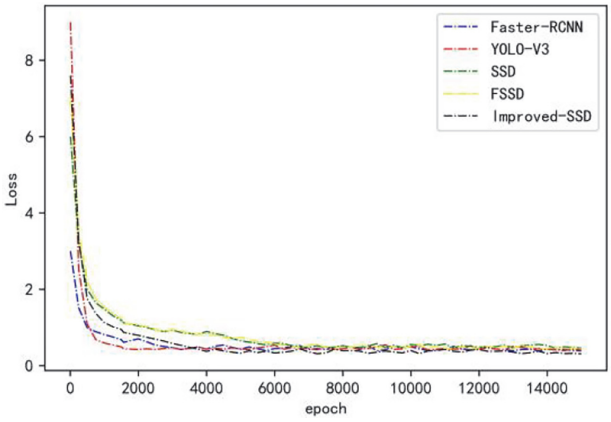


Fig. 8. Loss curves of different algorithms.

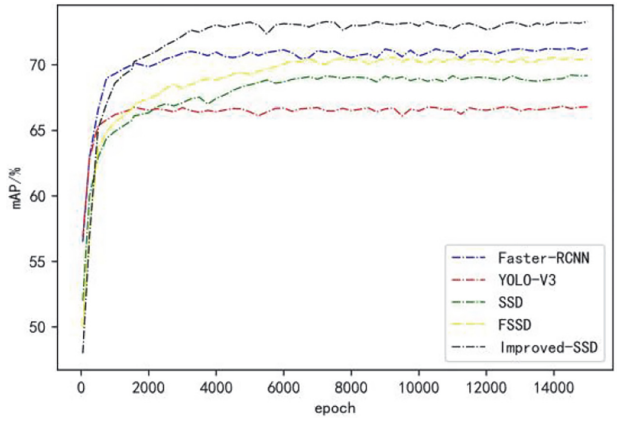


Fig. 9. Comparison of detection accuracy for small obstacles.

accuracy of small obstacles. The accuracy of detecting a car in SSD+two-stage deconvolution is increased by 0.41% when compared with that of the SSD algorithm. The results show that the two-stage deconvolution module improves the accuracy of detecting large target obstacles. However, the complexity of these two modules reduces the model detection speed. SSD+optimized prior bounding box achieves greatly improved detection accuracy with various types of obstacles. The speed of obstacle detection in SSD+convolutional split is increased by 10.9 FPS when compared with the SSD algorithm. The results show that the convolutional split structure can reduce the number of parameters of the model and improve the detection speed of the model. Compared to SSD and the single module improvement algorithm, Improved-SSD not only achieves a substantial improvement in the detection accuracy of track obstacles but also maintains a high detection speed.

This paper uses five algorithms: Faster RCNN, YOLOv3, SSD, FSSD (feature fusion single shot multibox detector), and Improved-SSD to train the self-made dataset. Among them, FSSD is an improved SSD algorithm that achieves improved accuracy in small target detection. The loss function curves are shown in Fig. 8.

In the initial training stage, the loss value of the Improved-SSD algorithm is greater than that of the SSD algorithm. This is attributed to the fact that the Improved-SSD algorithm adds bidirectional deconvolution modules and two-stage deconvolution modules, which increases the complexity of the network. In the process of the first 2,500 iterations, the loss of the Improved-SSD algorithm decreases rapidly. After 5,000 iterations, the loss of the Improved-SSD algorithm tends to converge, its convergence speed is faster than that of the SSD and FSSD algorithms, and the convergence value is the lowest. According to the above analysis, the modules added by the Improved-SSD algorithm improve the extraction ability of feature information, which is more conducive to later training.

To verify the accuracy of the proposed Improved-SSD algorithm in detecting small obstacles in orbit, five algorithms were tested using the animal dataset. Rectified intersection over union (IoU) is set to 0.6 and the test results are shown in Fig. 9.

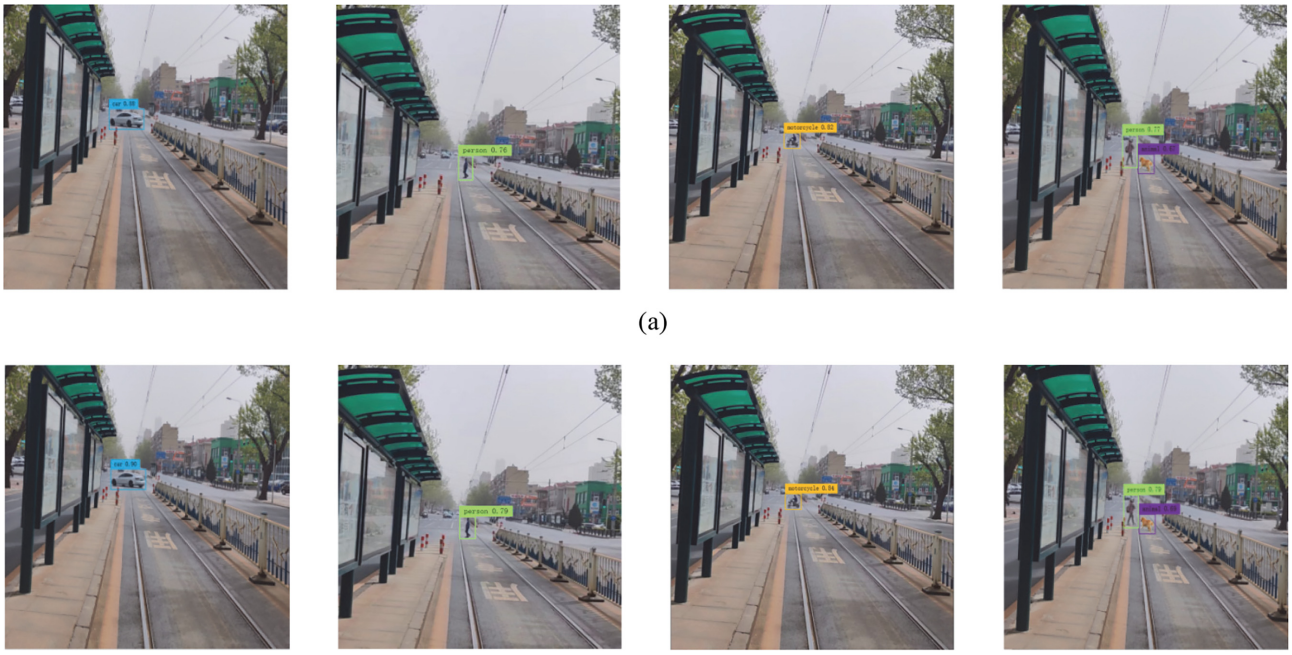
In the initial stages of training, the Improved-SSD algorithm detects small obstacles with slightly less accuracy than other algorithms. This is due to the increase in complexity as the network improves. In about 5,000 iterations, the Improved-SSD algorithm detects small obstacles with significantly better accuracy than other algorithms. The Improved-SSD algorithm is designed with a bidirectional fusion module, which increases the ability of Conv4_3 to extract the features of small target obstacles. Optimizing the generation method of the aspect ratio of the prior frame also improves the accuracy of the Improved-SSD algorithm in detecting small obstacles.

Table 4 lists the AP and FPS obtained by the five algorithms on the self-made dataset for various types of orbital obstacles.

Compared to the Faster RCNN, YOLOv3, SSD, and FSSD algorithms, the mAP of the Improved-SSD algorithm is improved by 0.38%, 0.77%, 1.17%, and 0.88%, respectively. Each type of obstacle also has high detection accuracy. In terms of detection speed, the FPS of the Improved-SSD algorithm reaches 26.4 FPS, which is faster than the SSD and FSSD algorithm.

The SSD and Improved-SSD algorithms were used to detect some obstacles in the self-made dataset, and the visualization results are shown in Fig. 10.

As can be seen in Fig. 10, the Improved-SSD algorithm is superior to the SSD regardless of the confidence in obstacles and the matching between the prediction box and the real box. This is because the improved-SSD enriches the semantic information of the low-level feature layer, enhances the edge information of the high-level feature layer, and improves the confidence level of detecting obstacles. According to the clustering results,



(a)

(b)

Fig. 10. Visual comparison chart of track obstacle detection: (a) SDD and (b) Improved-SSD.

the Improved-SSD optimizes the generation method of the aspect ratio of the prior bounding box such that the prediction box matches the real box better.

IV. CONCLUSION

To improve the detection accuracy and efficiency of urban tram track obstacles, this paper proposes an urban tram track obstacle detection algorithm based on Improved-SSD. To enhance the semantic information of the low-level feature layer, a two-way fusion module is designed. A two-stage deconvolution module is designed to enrich the feature information of the high-level feature layer. The convolutional split structure is adopted to improve the detection speed of obstacles. Moreover, to improve the matching degree between the prior bounding box and the obstacle target, the k-means algorithm is used to optimize the aspect ratio of the prior frame. Under the self-made dataset, the mAP of the Improved-SSD algorithm is 89.87%, and the detection speed is 21.4 FPS. The improved algorithm effectively improves the detection accuracy and speed of track obstacles, and it provides technical support facilitating the safe driving of trams.

CONFLICT OF INTEREST

The authors have declared that no competing interests exist.

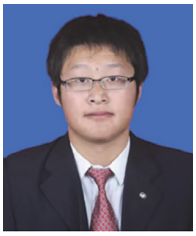
AKNOWLEDGMENTS

The authors would like to thank the entire project team for their innovation and help. This research was funded by the Basic Research Project of Liaoning Provincial Department of Education (LJKMZ20220857), the Science and Technology Program of Liaoning Province (2021-BS-219), and the Dalian Science and Technology Innovation Fund Project (2021JJ13SN82).

REFERENCES

1. T. Shen, Y. Qian, L. Xie, T. Yuan, X. Zeng, and X. Zhang, "Obstacle detection algorithm of fully automatic train considering reflection intensity," *Journal of Tongji University (Natural Science Edition)*, vol. 50, no. 1, pp. 6-12, 2022.
2. H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain intelligence: go beyond artificial intelligence," *Mobile Networks and Applications*, vol. 23, pp. 368-375, 2018. https://doi.org/10.1007/s11036-017-0932-8
3. X. Wu, X. Song, S. Gao, and C. Chen, "A review of object detection algorithms based on deep learning," *Transducer and Microsystem Technologies*, vol. 40, no. 2, pp. 4-7, 2021. https://doi.org/10.13873/J.1000-9787(2021)02-0004-04
4. N. Milosevic and M. Rackovic, "Classification based on missing features in deep convolutional neural networks," *Neural Network World*, vol. 29, no. 4, pp. 221-234, 2019. https://doi.org/10.14311/nnw.2019.29.0015
5. M. Mirkhan and M. R. Meybodi, "Restricted convolutional neural networks," *Neural Processing Letters*, vol. 50, pp.

1705-1733, 2019. https://doi.org/10.1007/s11063-018-9954-x
6. 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*, Columbus, OH, USA, 2014, pp. 580-587. https://doi.org/10.1109/CVPR.2014.81
7. R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1440-1448. https://doi.org/10.1109/ICCV.2015.169
8. 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. https://doi.org/10.1109/TPAMI.2016.2577031
9. H. Pan, H. Zhang, X. Lei, F. Xin, and Z. Wang, "Hybrid dilated faster RCNN for object detection," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 1, pp. 1229-1239, 2022. https://doi.org/10.3233/JIFS-212740
10. H. Wang and N. Xiao, "Underwater object detection method based on improved Faster RCNN," *Applied Sciences*, vol. 13, no. 4, article no. 2746, 2023. https://doi.org/10.3390/app13042746
11. 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, NV, USA, 2016, pp. 779-788. https://doi.org/10.1109/CVPR.2016.91
12. J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017, pp. 6517-6525. https://doi.org/10.1109/CVPR.2017.690
13. R. Tu, Z. Zhu, Y. Bai, G. Jiang, and Q. Zhang, "Improved YOLO v3 network-based object detection for blind zones of heavy trucks," *Journal of Electronic Imaging*, vol. 29, no. 5, article no. 053002, 2020. https://doi.org/10.1117/1.JEI.29.5.053002
14. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Computer Vision–ECCV 2016.* Cham, Switzerland: Springer, 2016. pp. 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
15. X. Zhang, X. Dong, Q. Wei, and K. Zhou, "Real-time object detection algorithm based on improved YOLOv3," *Journal of Electronic Imaging*, vol. 28, no. 5, article no. 053022, 2019. https://doi.org/10.1117/1.JEI.28.5.053022
16. J. Xue, J. Zhu, J. Zhang, X. Li, S. Dou, Milin, Z. Li, X. Yuan, and C. Li, "Object detection in optical remote sensing images based on FFC-SSD model," *Acta Optica Sinica*, vol. 42, no. 12, pp. 138-148, 2022. https://doi.org/10.3788/AOS202242.1210002
17. T. Wu, X. Wang, Y. Cai, Y. Jing, and C. Chen, "Lightweight SSD object detection method based on feature fusion," *Chinese Journal of Liquid Crystals and Displays*, vol. 36, no. 10, pp. 1437-1444, 2021. https://doi.org/10.37188/CJLCD.2021-0007
18. W. Chen, Y. Qiao, and Y. Li, "Inception-SSD: an improved single shot detector for vehicle detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 5047-5053, 2022. https://doi.org/10.1007/s12652-020-02085-w
19. N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using K-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science*, vol. 54, pp. 764-771, 2015. https://doi.org/10.1016/j.procs.2015.06.090



Yunming Wang https://orcid.org/0000-0003-1289-047X

Yunming Wang received Ph.D. degree in control science and engineering from the Nanjing University of Technology, Nanjing, China, in 2017. He is currently an associate professor with the College of automation and electrical engineering, Dalian Jiaotong University, Dalian, China. His research interests include network science, algorithms on networks, network dynamics, and machine learning.

Yiang Zhou https://orcid.org/0009-0007-2991-1499

Yiang Zhou is studying for a master's degree in Traffic Information Engineering and Control from Dalian Jiaotong University, China. His main research interest is in deep learning, mainly focusing on the research of time series prediction using recurrent neural network.

Xianwu Chu https://orcid.org/0009-0005-9186-0472

Xianwu Chu received his Bachelor's degree from Liaoning Technical University from September 1996 to July 2000. He continued his study at the same institution, earning his Master's degree from September 2000 to March 2003. Professor Chu began his teaching career at the School of Electrical and Information Engineering, Dalian Jiaotong University, in July 2003, where he has been a dedicated educator ever since. Since August 2021, he has been a faculty member at the College of Automation and Electrical Engineering, Dalian Jiaotong University. His research interests are in rail transit information and control technology.

Guodu Peng https://orcid.org/0000-0002-0122-102X

Guodu Peng is studying for a master's degree in Traffic Information Engineering and Control from Dalian Jiaotong University, China. His main research interest is in deep learning, mainly focusing on the research of time series prediction using recurrent neural network.