Sensors and Materials, Vol. 35, No. 2 (2023) 607–621
MYU Tokyo

S & M 3198

Vehicle Detection Algorithm Based on Background Features Assistance in Remote Sensing Image

Yifei Cao,[1*] Yingqi Bai,[2] Ran Pang,[1] Boyu Liu,[3] and Kui Zhang[1]

[1]Beijing Institute of Surveying and Mapping, No. 15, Yangfangdian Road, Haidian District, Beijing 100038, China
[2]SpaceWill Info. Co., Ltd., No.1 1, Changchun Qiao Road, Haidian District, Beijing 100089, China
[3]Beijing University of Civil Engineering and Architecture,
No. 15, Yongyuan Road, Daxing District, Beijing 102616, China

(Received October 28, 2022; accepted January 16, 2023)

Keywords: vehicle detection, background assistance, remote sensing, similar interference

Toward solving the problem of the lack of useful features caused by inconspicuous vehicle features and the interference of similar features around a vehicle in the process of remote sensing image vehicle detection, we propose an algorithm based on the assistance of background features. This algorithm is based on the YOLOv4 model and includes a weight redistribution module based on the feature correlation degree. The model introduces background features around a vehicle so that the model learns to determine the environment in which the vehicle target is located during training. This can effectively reduce the occurrence of missed detection. Moreover, the algorithm increases the sensitivity of the backbone to vehicle features through the weight redistribution of the features inside the anchor box, thus making full use of the feature correlation between the vehicle target and the surrounding background. As a result, the vehicle target is effectively distinguished from the interference target. The experimental results show that the precision rate and success rate of this algorithm on the DLR-3K dataset reach 75.4 and 68.5%, respectively. The model detection rate reached 12.7 frames/s. The proposed algorithm has high performance in executing vehicle detection tasks in the presence of interference due to similar targets.

1. Introduction

With the rapid development of artificial intelligence technology, the construction of an intelligent transportation system and the field of intelligent traffic management have been widely considered by researchers. Vehicle detection is an important prerequisite for data acquisition. Fast and accurate vehicle detection can ensure that the whole system is efficiently supplied with high-quality data. Therefore, it is of great practical importance to study vehicle target detection algorithms.(1)

Currently, the spatial resolution of remote sensing images has reached sub-meter level. High-resolution remote sensing images enable the clear identification of small targets of interest on the ground. The detection of vehicles, the most common form of transportation in remote sensing

*Corresponding author: e-mail: 1023362448@qq.com
https://doi.org/10.18494/SAM4204

ISSN 0914-4935

images, can provide data support for many fields.[2] For example, in the field of intelligent transportation system construction, vehicle detection can be applied to road condition information collection and road management information acquisition, and in the field of intelligent traffic management, vehicle detection can be applied to urban traffic flow statistics and road capacity control.

However, owing to the special characteristics of remote sensing images, vehicle feature detection based on remote sensing image data is still very challenging compared with detection based on ground images.[3] Specifically, although the imaging level of the sensors carried by satellites has been improving, the vehicle features in the captured remote sensing images are similar to some of the background features because the satellite is far from the ground, causing the problem that the algorithm cannot effectively distinguish vehicle targets from background features. Worldview-2 images, one of which is shown in Fig. 1, have reached a spatial resolution of 1.2 m and can effectively detect small targets such as vehicles. However, owing to the long imaging distance, the vehicle edge features are blurred. In addition, the shape and color of the vehicle driving on the tarmac at the top of Fig. 1 are almost the same as those of the building next to it. Therefore, problems such as small vehicle targets and the difficulty of distinguishing vehicles from the surrounding background exist with remote sensing images. These problems lead to the poor detection of vehicles using existing detection algorithms.

In recent years, vehicle detection algorithms based on deep learning have gradually become popular among researchers. Compared with the traditional target detection algorithms, deep-learning-based vehicle detection algorithms have greater robustness and generalization ability and can autonomously perform the feature recognition of vehicle targets without manual feature design, making them less influenced by human factors. At present, the mainstream algorithms are divided into two categories: one-stage detection algorithms, typified by the YOLO[4,5] and SSD[6,7] series, and two-stage detection algorithms, such as the R-CNN[8,9] series. Both categories have advantages and disadvantages in terms of accuracy and detection speed.

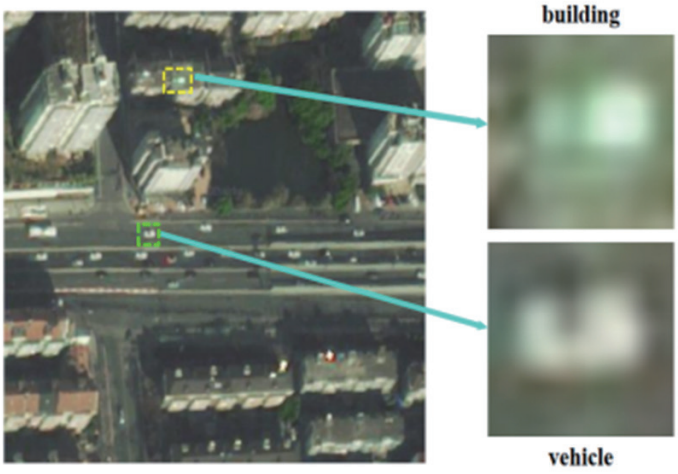


Fig. 1. (Color online) Display of different vehicles in remote sensing images.

One-stage target detection algorithms are based on the concept of regression. For these algorithms, it is no longer necessary to extract a proposal region. The images to be detected are directly input into the network, and the target bounding box and category information in the images to be detected are obtained in the output layer. Although the detection accuracy of such algorithms is lower than that of algorithms based on a proposal region, the detection speed basically meets real-time requirements. Etten[10] fused the features of different scales by using a passthrough layer, which improved the sensitivity of their model to small targets. They used different detectors for different scales to cope with extreme scale variations. However, because they only considered the effect of the target scale on the detection performance and ignored the importance of background features, the detection performance was reduced when the target was affected by a complex background. On this basis, Cao *et al.*[7] introduced contextual information into a remote sensing image vehicle detection task by channel superposition to improve the ability of their algorithm to detect vehicles in complex scenes.

Two-stage target detection algorithms are based on the idea of proposal regions. Firstly, a region of interest that may contain targets is slidingly extracted from an image using a filter window to reduce the influence of irrelevant regions on target detection. Then, a convolutional neural network is used to classify and recognize the targets in the extracted regions. Schumann *et al.*[11] argued that vehicle targets have few sample sizes in a dataset, and the number of layers in current feature extraction networks can often reach 100 or more. Thus, these problems led to the unsatisfactory detection of vehicles in remote sensing images. Therefore, the authors performed a sample enhancement of the vehicle targets in the dataset and also proposed a shallow backbone for vehicle detection by adopting the idea of R-CNN. Chen *et al.*[12] applied Faster R-CNN to the vehicle detection task. They changed the number and size of convolutional kernels in the backbone to aggregate the detected vehicle feature of each layer to a similar scale, then they selected the frame of the detection target by using an anchor box with a uniform scale. Although the two-stage target detection algorithms based on proposal regions can achieve high accuracy, their real-time performance is weak and they cannot meet the required detection speed in actual production.

The existing vehicle target detection algorithms for remote sensing images have improved the extraction capability of convolutional neural networks for vehicles by increasing the depth of the backbone and expanding the image contrast. However, these algorithms still have two problems. First, the existing remote sensing image vehicle detection algorithms focus on the expression of vehicle features, ignoring the influence of the background features around the target, and they fail to effectively use the background features around the target. Secondly, the existing algorithms are weak in distinguishing the vehicle from its surrounding background features, making them unable to obtain a clear vehicle outline. To solve these problems, we take the YOLOv4 algorithm as the base model and improve it to propose a one-stage vehicle detection algorithm. This algorithm can effectively use the vehicle features in remote sensing images and the background features around the vehicle.

2. Experimental Results and Analysis

2.1 YOLOv4

The YOLO series of algorithms divide the input image into $s \times s$ grids, and if the center of the target to be detected falls within a grid, this grid is responsible for predicting that target. Each grid predicts B bounding boxes and one item of category information, and the bounding box contains the target position and confidence information. Employing this detection approach, YOLOv4 predicts three bounding boxes for each grid, each containing coordinate information (x, y, w, h) as well as one item of confidence information and C items of conditional category probability information.[13] The YOLOv4 network structure consists of four parts: an input layer, a backbone, a neck, and a head, and the specific network structure diagram is shown in Fig. 2.

The computation can be divided into four steps: first, the input image is preprocessed by the input layer, and the image is adaptively scaled to unify the image size, while the anchor box is adaptively computed by the Mosaic[14] data enhancement method. Second, the preprocessed images are fed into the CSP-Darknet53 backbone containing five cross stage partial (CSP) modules for target feature extraction in the images. Third, the spatial pyramid pooling (SPP)[15] module with path aggregation network (PANet)[16] is used to generate candidate boxes on the feature maps generated after the processing of the fourth and fifth CSP modules at different scales. Finally, the original image is downsampled by 8×, 16×, and 32× by three YOLO heads to generate three feature vectors of different sizes. These vectors can predict image features, generate bounding boxes, and predict target classes and confidence levels in the image.

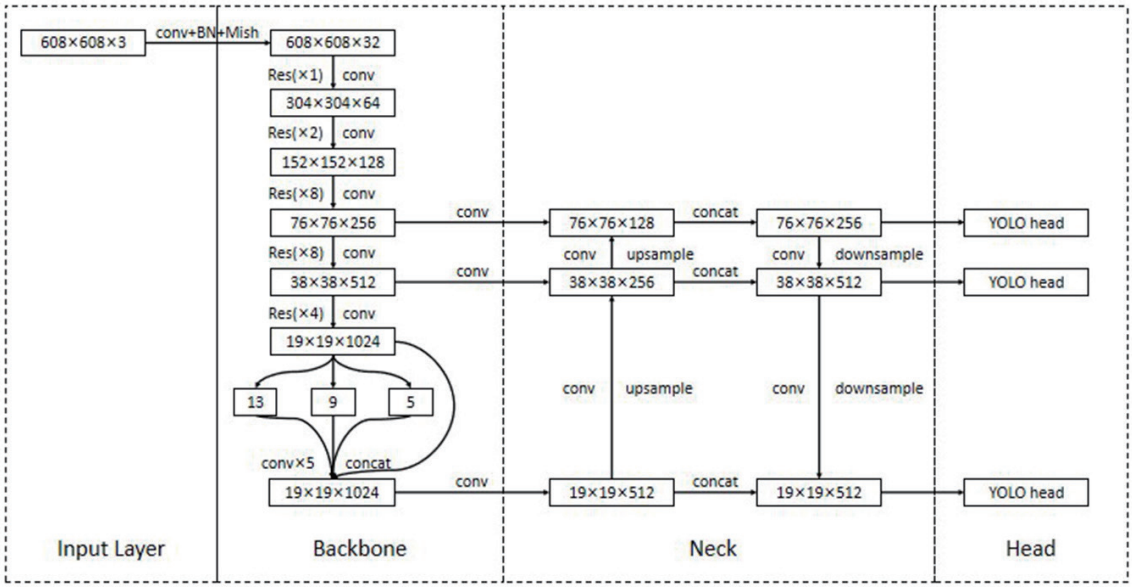


Fig. 2. YOLOv4 structure diagram. Res(×1), Res(×2), Res(×8), Res(×8), and Res(×4) from top to bottom represent the first five CSP modules.

2.2 CSP-Darknet53

CSP-Darknet53 adds the CSP module[17] to each residual block of Darknet53. The basic principle is to divide a feature map into two parts. One part is input to the residual network to continue the downsampling operation, and the other part is recovered from the scale by upsampling to concatenate with the previous layer of the feature map. Compared with the feature map obtained by direct downsampling, the feature map obtained by the CSP method can effectively extract the image features while retaining stronger location information. This further increases the usability of the features and enhances the learning ability of the network.

CSP-Darknet53 uses the Mish function as the activation function:

$$Mish = x \times \tanh\left(\ln\left(1+e^{x}\right)\right), \tag{1}$$

where x is the eigenvalue of the current sampling point and tanh is the hyperbolic tangent function. It can be seen that when x tends to infinity, this function can take any value, avoiding the disappearance of the gradient during the training process.

2.3 SPP and PANet

To enable images of arbitrary size to be input into the neural network, YOLOv4 employs the SPP network. We use four different scales of maximum pooling for feature enhancement and to increase the receptive field, where the pooling kernel sizes are 13 × 13, 9 × 9, 5 × 5, and 1 × 1. The perceptual field is calculated as

$$N = \frac{(w-f+2p)}{s}+1, \tag{2}$$

where w is the input image size, f is the convolution kernel size, p is the padding size, and s is the step size.

In addition, YOLOv4 uses PANet instead of feature pyramid networks (FPN).[18] PANet uses the bottom-up downsampling method to conduct multichannel feature fusion and uses the low-level strong position information to increase the positioning accuracy of feature extraction. This part uses the Leaky Rectified Linear Unit (ReLU) activation function

$$f(x)\begin{cases} x, & \text{if } x > 0 \\ \lambda x, & \text{if } x \le 0 \end{cases}, \tag{3}$$

where λ is the parameter of the Leaky ReLU and is in the range of 0 to 1. From the formula, we can see that the part of the Leaky ReLU that is less than 0 is non-zero, solving the problem that some parameters cannot be updated.

2.4 Loss function

In addition to the innovative structure of its algorithm, YOLOv4 also improves the loss function of the algorithm by adopting *CIoU_Loss* [Eq. (4)], which takes into account the distance from the center point of the bounding box, and calculates the bounding box aspect ratio.

$$CIoU\_Loss = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{4}$$

Here, *IoU* is the intersection ratio between the anchor box and the bounding box. ρ is the Euclidean distance between the two centroids. b and b^{gt} are the center points of the anchor box and bounding box, respectively. c is the diagonal distance of the smallest outer rectangle that can contain both the anchor box and the bounding box. αv is the influence factor, which indicates the consistency between the aspect ratios of the anchor box and the bounding box. α is the weight function, and v and α are calculated as

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2, \tag{5}$$

$$\alpha = \frac{v}{(1 - IoU) + v}. \tag{6}$$

3. Our Work

In remote sensing images, the appearances of vehicles are diverse and complex. In many cases, it is difficult to effectively detect vehicles by relying only on the features of the vehicles themselves. Sometimes, other targets with similar appearances to vehicles may even be mistaken as vehicles. For example, vehicle detection in remote sensing images using only the shape of the vehicles themselves may be difficult. Because a large number of vehicles and buildings are rectangular in remote sensing images, the shapes of the roofs of buildings are very similar to those of vehicles in the process of feature extraction. This will mislead the detection network and cause it to incorrectly identify the roofs of buildings as vehicles. In this case, making full use of the feature correlation between the vehicle and the surrounding background can minimize the problem of missed and false vehicle detection caused by unclear vehicle features. Therefore, in this paper, we propose a weight redistribution module based on the feature correlation degree to reduce the false detection rate of the network for non-vehicle objects. The flow of the proposed method is shown in Fig. 3.

In the training phase, the parameters related to the bounding box in the dataset are first obtained. Then, the bounding box is mapped onto the 76 × 76 × 256 feature map obtained by the third CSP module to obtain the vehicle feature map Z_{car} containing m bounding boxes. Because

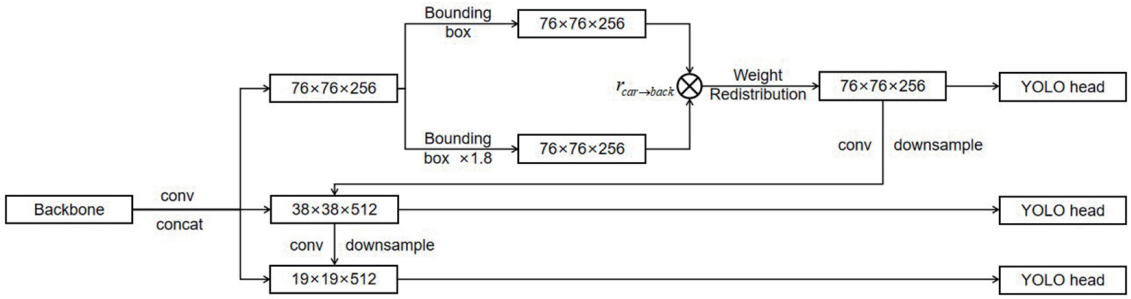


Fig. 3. Schematic diagram of the weight redistribution module based on feature correlation degree.

YOLOv4 maps the feature map from the backbone to the neck after the calculation of the third CSP module, the bounding box is mapped to the third CSP module to provide the neck with a higher-quality feature map. Then, the bounding box is expanded by a factor of 1.8 to obtain the corresponding parameters. The expanded bounding box includes the vehicle itself as well as some background features around the vehicle. Finally, the expanded box is mapped to the feature map at the same location to obtain a feature map Z_{back} containing m vehicles and the surrounding background. After obtaining Z_{car} and Z_{back}, the feature correlation degree between each vehicle and the surrounding background features $r_{car\to back}$ is calculated by

$$r_{car\to back} = relu\left(W_s \cdot e^s_{car\to back}\right) \cdot \tanh\left[W_v \cdot conv\left(Z_{car}, Z_{back}\right)\right], \tag{7}$$

where $r_{car\to back}$ is a weight scalar. $relu(\cdot)$ is a linear rectification function. W_s is a spatial location relationship weight vector whose value is obtained by training iterations. $\tanh(\cdot)$ is the hyperbolic tangent function. W_v is a visual feature relationship weight vector whose values are also obtained by training iterations. $conv(Z_{car}, Z_{back})$ indicates that the vehicle feature map Z_{car} is used as the convolution kernel to conduct mutual convolution between the vehicle and the surrounding background feature map Z_{back}. Cross convolution can add vehicle information to the background area around the vehicle and highlight the vehicle feature expression. Therefore, $conv(Z_{car}, Z_{back})$ can reflect the relationship between the vehicle and the background in a feature map. $e^s_{car\to back}$, representing the spatial location relationship between the bounding boxes and the expanded bounding boxes, can be expressed as

$$e^s_{car\to back} = \left(w, h, s, w', h', s', \frac{x-x'}{w'}, \frac{y-y'}{h'}, \frac{(x-x')^2}{w'^2}, \frac{(y-y')^2}{h'^2}, \log\frac{x-x'}{w'}, \log\frac{y-y'}{h'} \right), \tag{8}$$

where x and y represent the coordinates of the upper-left corner of the bounding box. w and h represent the width and height of the bounding box, respectively. s represents the area of the bounding box. x' and y' represent the coordinates of the upper-left corner of the expanded bounding box. w' and h' represent the width and height of the expanded bounding box, respectively. s' represents the area of the expanded bounding box.

In the verification phase, the weight redistribution module based on the feature correlation degree no longer takes the relevant parameters of the bounding boxes as input values; instead, it takes the parameters of the anchor box generated using the Mosaic mechanism as the input. The average value of $r_{car\rightarrow back}$ is counted and used as the basis for judging whether the candidate target is a vehicle. The weights are reassigned to the targets within the range of the anchor box by using the following equation to obtain the output feature map F:

$$F = Z \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{(r'_{car\rightarrow back} - \overline{r}_{car\rightarrow back})^2}{2}}, \tag{9}$$

where $r'_{car\rightarrow back}$ is the feature correlation degree between the target and the surrounding background features within the anchor box, $\overline{r}_{car\rightarrow back}$ is the average value of $r_{car\rightarrow back}$, and Z is the 76 × 76 × 256 feature map obtained by the third CSP module.

The weight redistribution module based on the feature correlation degree can determine the environment according to the background features around the candidate target. If the environment matches the category of the vehicle's environment learned in the training phase, then the features inside the anchor box are assigned high weights; otherwise, low weights are assigned. Through the weight redistribution of the features inside the anchor box, the sensitivity of CSP-Darknet53 to the vehicle features can be strengthened, enabling the head to more clearly discern the category to which the features belong, which has a positive impact on the detection results.

4. Experimental Results and Analysis

4.1 Experimental data and environment

The dataset used in this study is the DLR-3K[19] dataset commonly used in the field of vehicle detection by aerial remote sensing, in which all images are collected through an aerial platform. This dataset is selected because it can cover various vehicle statuses in most remote sensing images, as shown in Fig. 4. The DLR-3K dataset contains 20 remote sensing images of size 5616 × 3744, and 14235 vehicle targets are manually labeled by a directional bounding box. The spatial resolution is 0.13 m, which is eight times that of current commercial remote sensing images. Therefore, we adopt the Gaussian kernel to convolve the original image to reduce the image size eightfold and generate an image with spatial resolution equivalent to 1.04 m. Finally, the dataset is randomly divided into a training set and a test set in the ratio of 7:3. The proposed algorithm is developed on the basis of the PyTorch framework and programming in Python. A desktop computer with an Intel Core I7-10700K CPU and an NVIDIA RTX3070 8 GB GPU is used to train and test the proposed algorithm.

Fig. 4. (Color online) DLR-3K dataset.

4.2 Test results and analysis

To visualize the detection performance of the proposed algorithm, the detection results of the algorithm with the DLR-3K dataset are qualitatively analyzed to verify the effectiveness of the proposed algorithm in remote sensing image vehicle detection. Some of the detection results for vehicle targets in remote sensing images are shown in Fig. 5.

From the first group of detection results in Fig. 5, a large number of interfering objects that are similar to vehicles can be seen in the residential areas around the highway, including air conditioners on roofs and white guard booths. The shape and color of these interfering objects in the images are very similar to those of vehicles. Nevertheless, the detection results show that the proposed algorithm can effectively distinguish vehicles from interfering objects. No false detection occurs in the detection results. However, the effectiveness of detecting vehicles is lower in areas covered by shadows.

In the second group of detection results, the distribution of vehicles on the highway is more intense, and some vehicles are obscured by the shadows of buildings. The detection results show that the proposed algorithm can better detect vehicles being driven at a high speed on the highway. The effectiveness of detection is again reduced when the shadows of buildings obscure the vehicles.

From the third group of detection results, it can be seen that the range of detection results is generally larger for vehicles moving at a high speed. This is because the vehicle speed is too fast when it is photographed by satellite, resulting in the unclear edge of the vehicle on the remote sensing image. This algorithm also detects these following shadows as vehicles, resulting in a wider detection range.

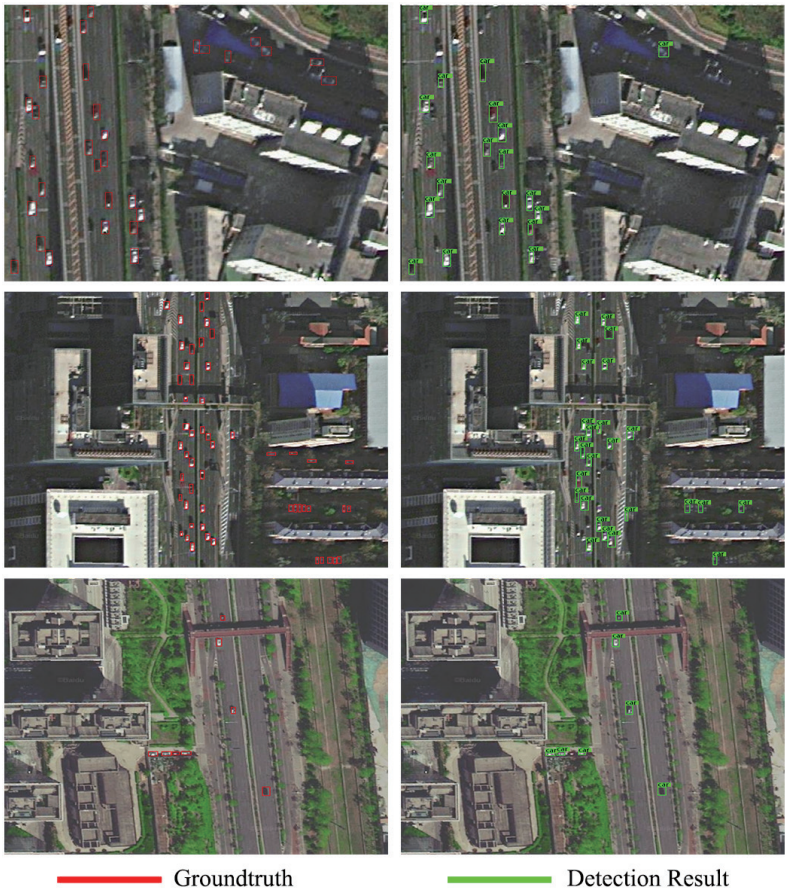


Fig. 5. (Color online) Vehicle detection results.

In summary, the vehicle detection algorithm based on the assistance of background features in remote sensing images can effectively detect densely distributed vehicles in remote sensing images. The proposed algorithm can also achieve better detection results for vehicles with a similar shape and color to the interfering objects.

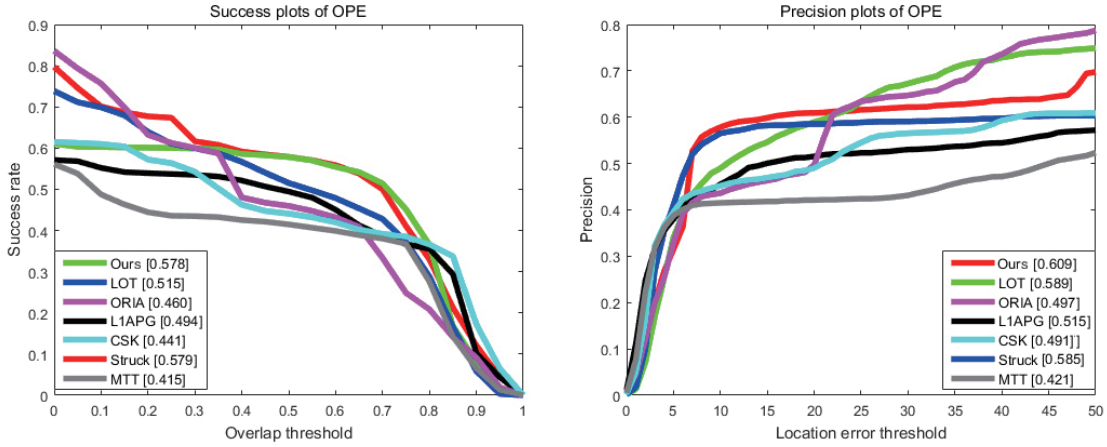
4.3 Comparison and analysis of algorithm performance

To verify the performance of the proposed algorithm, it is compared with classical remote sensing image vehicle detection algorithms with the same amount of training and the same training dataset. The algorithms adopted for comparison are SSD512, Faster R-CNN,[20] FCOS,[21] YOLT, YOLOv3,[22] and the unimproved YOLOv4 algorithm. Table 1 shows the experimental results.

As shown in Fig. 6, the precision rate and success rate of the proposed algorithm are significantly better than those of the other algorithms. According to the experimental results in Table 1, the precision rate of this algorithm reaches 75.4% and the success rate reaches 68.5%. Compared with YOLOv4, the detection speed is reduced by 1.9 f/s because of the inclusion of the

Table 1
Comparison of performance with other detection algorithms.

Algorithm	Precision (%)	Success (%)	F1 score (%)	FPS (f/s)
SSD512	40.3	36.8	39.2	9.6
Faster R-CNN	42.8	35.2	40.5	7.3
FCOS	46.9	24.9	37.3	10.1
YOLT	52.3	40.2	47.7	8.8
YOLOv3	48.9	39.1	45.7	13.4
YOLOv4	63.3	59.7	63.4	14.6
Proposed algorithm	75.4	68.5	73.8	12.7



(a) (b)

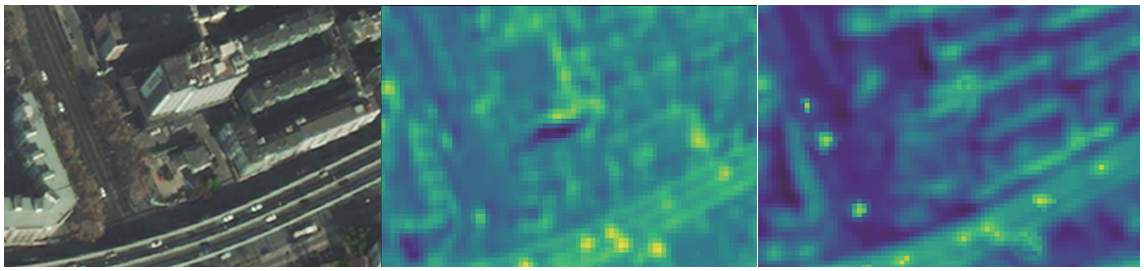
Fig. 6. (Color online) Precision and success rate plots of the different algorithms on the DLR-3K dataset: (a) precision rate plots and (b) success rate plots.

weight redistribution module based on the feature correlation degree during the feature extraction process. However, because of this, the precision rate and success rate of the proposed algorithm are improved by 12.1 and 8.8% compared with YOLOv4, and the F1 score of the proposed algorithm reaches 73.8%. It can be concluded that the proposed algorithm has a high-precision vehicle detection capability, but there are still some limitations in the detection speed.

4.4 Evaluation of effectiveness of weight redistribution module based on feature correlation degree

To verify the effectiveness of the weight redistribution module based on the feature correlation degree, the changes in the vehicle detection performance of YOLOv4 before and after adding the weight redistribution module are next compared.

Figure 7(a) shows the original image, from which it can be seen that there are many buildings near the road, and on top of the buildings, there are many interference targets similar to vehicles. As can be seen in Fig. 7(b), the feature maps output by the original YOLOv4 not only have high response values at the vehicle locations, but also have equally high response values at the



(a) (b) (c)

Fig. 7. (Color online) Comparison of feature extraction ability, where the closer the pixel is to yellow, the greater the vehicle feature extraction ability: (a) original image, (b) original YOLOv4 output feature map, and (c) feature map output by our algorithm.

locations of the interference targets. This makes it difficult to distinguish the interference targets and the vehicles, which causes a large number of interference targets to be mistakenly detected as vehicles. As can be seen in Fig. 7(c), after adding the weight redistribution module based on the feature correlation degree, the feature map output by YOLOv4 clearly distinguishes between vehicles and interference targets. In Fig. 7(c), the response values of non-vehicle regions are significantly reduced, and the difference between the vehicles and the background can be clearly seen. This is due to the fact that the backbone used in YOLOv4 directly judges whether the feature categories inside the anchor box belong to vehicles, and it does not judge the environment where the anchor box is located after feature extraction from remote sensing images. In contrast, the weight redistribution module based on the feature correlation degree can calculate the correlation degree between the features inside the anchor box and the background features around the anchor box. Through the a priori knowledge accumulated in the training phase, the environment where the anchor box is located can be judged. Then the weights of the anchor box categories can be redistributed.

In summary, the weight redistribution module based on the feature correlation degree can improve the differentiation between vehicle features and other similar features in the feature map. The detector can subsequently detect the vehicle targets in the remote sensing images more accurately.

4.5 Ablation experiments

To determine the influence of the range of background information around the vehicle on the vehicle detection capability of the proposed algorithm, we conduct ablation experiments on the expanded range of the bounding box. Expansion rates of 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0 are used in the experiments. The experimental environment and hyperparameter settings are the same in all the experiments.

As shown in Table 2, the detection speed of the proposed algorithm without introducing background features (expansion rate of 1.0) is 14.6 f/s. With increasing expansion rate, the detection speed of the algorithm gradually decreases, which is due to the large amount of

Table 2
Detection performance of proposed algorithm for different expansion rates.

Expansion rate	Precision (%)	Success (%)	F1 score (%)	FPS (f/s)
1.0	63.3	59.7	63.4	14.6
1.2	61.4	56.3	60.5	14.4
1.4	63.6	58.9	58.3	14.0
1.6	67.5	64.2	70.7	13.4
1.8	75.4	68.5	73.8	12.7
2.0	72.8	69.2	72.1	11.8

computational resources consumed in background feature extraction. Compared with the expansion rate of 1.2, the precision rate and success rate for the expansion rate of 1.0 are 1.9% and 3.4% higher, respectively, indicating that the introduction of a small amount of background features around the vehicle negatively affects the detection performance. When the expansion rate is further increased from 1.2, the precision rate and success rate gradually improve. When the expansion rate is 2.0, the success rate reaches the highest value of 69.2%, but the precision rate and detection speed are lower than those for an expansion rate of 1.8. For the expansion rate of 1.8, the proposed algorithm achieves a precision rate of 75.4%, a success rate of 68.5%, an F1 score of 73.8%, and a detection speed of 12.7 f/s, giving the best overall result in this experiment. In summary, the introduction of a moderate amount of background features around vehicles when performing vehicle detection can help the backbone distinguish vehicle features from other similar interference features and then improve the backbone to extract vehicle features.

5. Conclusions

We have proposed an algorithm for detecting vehicles in remote sensing images. We introduced a weight redistribution module based on the feature correlation degree in the backbone to improve the extraction capability of YOLOv4 for vehicle features. The network can reduce the misclassification rate for non-vehicle objects by calculating the feature correlation degree between the vehicle target and the surrounding background, thus improving the vehicle detection precision rate and success rate. The experimental results on DLR-3K, an aerial remote sensing vehicle detection dataset, show that the average accuracy of this algorithm exceeded that of YOLOv4, with improved recall and success rates. The detection speed of the proposed algorithm is also greater than that of YOLOv3. Moreover, the detection accuracy of vehicles is higher in the case of similar target interference. However, the proposed algorithm requires further improvement. In particular, the number of parameters in the algorithm should be reduced to improve its detection speed.

Acknowledgments

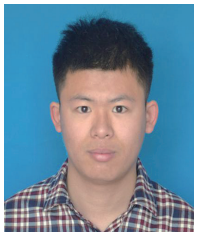
This work was supported by the National Science and Technology Major Project (grant no. 65-Y50G01-9001-22/23) and Beijing Key Laboratory of Urban Spatial Information Engineering, No. 2020204.

References

1 C. Gómez, J. C. White, and M. A. Wulder: ISPRS J. Photogramm. Remote Sens. **116** (2016) 55. https://doi.org/10.1016/j.isprsjprs.2016.03.008.
2 J. Luo, H. Fang, F. Shao, Y. Zhong, and X. Hua: Defence Technol. **17** (2021) 1542. https://doi.org/10.1016/j.dt.2020.10.006
3 Y. Cui, B. Hou, Q. Wu, B. Ren, S. Wang, and L. Jiao: IEEE Trans. Geosci. Remote Sens. **60** (2022) 1. https://doi.org/10.1109/TGRS.2021.3096809
4 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2016) 779–788. https://doi.org/10.48550/arXiv.1506.02640
5 J. Redmon and A. Farhadi: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2017) 7263–7271. https://doi.org/10.48550/arXiv.1612.08242
6 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg: Proc. European Conf. Computer Vision (ECCV, 2016) 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
7 G. Cao, X. Xie, W. Yang, Q. Liao, G. Shu, and J. Wu: Proc. SPIE 10615, 9th Int. Conf. Graphic and Image Processing (ICGIP, 2017) 1–8. https://doi.org/10.1117/12.2304811
8 R. Girshick, J. Donahue, T. Darrell, and J. Malik: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2013) 580–587. https://doi.org/10.48550/arXiv.1311.2524
9 R. Girshick: Proc. IEEE Int. Conf. Computer Vision (IEEE, 2015) 1440–1448. https://doi.org/10.48550/arXiv.1504.08083
10 A. Van Etten: Comput. Vision Pattern Recognit. (2018) 1. https://doi.org/10.48550/arXiv.1805.09512
11 A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer: Proc. 14th IEEE Int. Conf. Advanced Video and Signal Based Surveillance (IEEE, 2017) 1–6. https://doi.org/10.1109/AVSS.2017.8078558
12 C. Chen, M. Liu, O. Tuzel, and J. Xiao: Proc. 13th Asian Conf. Computer Vision (ACCV, 2016) 214–230. https://doi.org/10.10079/978-3-3154198_14
13 A. Bochkovskiy, C. Wang, and H. M. Liao: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2020) 1–17. https://arxiv.org/pdf/2004.10934.pdf
14 S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe: Proc. IEEE Int. Conf. Computer Vision (IEEE, 2019) 6022–6031. https://doi.org/10.1109/ICCV.2019.00612
15 K. He, X. Zhang, S. Ren, and J. Sun: IEEE Trans. Pattern Anal. Mach. Intell. **37** (2014) 1904. https://doi.org/10.1109/TPAMI.2015.2389824
16 S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2018) 8759–8768. https://doi.org/10.1109/CVPR.2018.00913
17 C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh: Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (IEEE, 2020) 1571–1580. https://doi.org/10.1109/CVPRW50498.2020.00203
18 T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie: Proc. IEEE Conf. Computer Vision and Pattern Recognition (IEEE, 2017) 936–944. https://doi.org/10.1109/CVPR.2017.106
19 K. Liu and G. Mattyus: IEEE Geosci. Remote Sens. Lett. **12** (2015) 1938. https://doi.org/10.1109/LGRS.2015.2439517
20 S. Ren, K. He, R. Girshick, and J. Sun: Proc. 28th Conf. Neural Information Processing Systems (NIPS, 2015) 91–99. https://doi.org/10.48550/arXiv.1506.01497
21 Z. Tian, C. Shen, H. Chen, and T. He: Proc. IEEE Int. Conf. Computer Vision (IEEE, 2019) 9626–9635. https://doi.org/10.48550/arXiv.1904.01355
22 J. Redmon and A. Farhadi: Comput. Vision and Pattern Recognit. (2018) 1. https://doi.org/10.48550/arXiv.1804.02767

About the Authors

Yifei Cao received his B.E. degree in surveying and mapping engineering from Beijing University of Civil Engineering and Architecture, Beijing, China, in 2019 and his master's degree from Beijing University of Civil Engineering and Architecture in 2022. Since 2022, he has been an assistant engineer at Beijing Institute of Surveying and Mapping, China. His current research interests include target detection and tracking in remote sensing. (1023362448@qq.com)



Yingqi Bai received his master's degree in surveying and mapping engineering from Beijing University of Civil Engineering and Architecture, Beijing, China, in 2022. He is currently working at SpaceWill Info Co., Ltd. His research interests include object detection tracking in remote sensing. (byq15624984098@126.com)

Ran Pang received her bachelor's degree from Beijing University of Civil Engineering and Architecture, China, in 2019 and her master's degree from Beijing University of Civil Engineering and Architecture in 2022. Since 2022, she has been an assistant engineer at Beijing Institute of Surveying and Mapping, China. Her research interests are in remote sensing, image processing, and geographic information system maintenance. (510723874@qq.com)

Boyu Liu is studying for a bachelor's degree at Beijing University of Civil Engineering and Architecture and is expecting to graduate in 2025. Her major is navigation engineering. In this paper, her main work was data collection and processing. (liuboyu@stu.becua.edu.cn)

Kui Zhang received his M.S. degree from the School of Information Engineering, China University of Geosciences (Beijing) in 2021. He is currently an assistant engineer at Beijing Institute of Surveying and Mapping, China, where his work focuses on the comprehension of very high spatial resolution images.