Hindawi
International Journal of Aerospace Engineering
Volume 2020, Article ID 8816187, 10 pages
https://doi.org/10.1155/2020/8816187

Research Article

R-CNN-Based Satellite Components Detection in Optical Images

Yulang Chen,[1] Jingmin Gao,[1] and Kebei Zhang[2]

[1]*School of Automation, Beijing Information Science & Technology University, Beijing 100192, China*
[2]*Beijing Institute of Control Engineering, Beijing 100190, China*

Correspondence should be addressed to Jingmin Gao; gaojm_biti@163.com

Received 12 March 2020; Revised 24 July 2020; Accepted 17 September 2020; Published 5 October 2020

Academic Editor: Jeremy Straub

The accurate detection of satellite components based on optical images can provide data support for aerospace missions such as pointing and tracking between satellites. However, the traditional target detection method is inefficient when performing calculations and has a low detection precision, especially when the attitude of the satellite and illumination conditions change considerably. To enable the precise detection of satellite components, we analyse the imaging characteristics of a satellite in space and propose a method to detect the satellite components. This approach is based on a regional-based convolutional neural network (R-CNN), and it can enable the accurate detection of various satellite components by using optical images. First, on the basis of the Mask R-CNN, we combine the DenseNet, ResNet, and FPN to construct a new feature extraction structure and obtain the R-CNN based satellite-component-detection model (RSD). The feature maps are extracted and concatenated at a deeper multiscale level, and the feature propagation between each layer is enhanced by providing a dense connection. Next, an information-rich satellite dataset is constructed, which is composed of images of various kinds of satellites from various perspectives and orbital positions. The detection model is trained and optimized on the constructed dataset to obtain the satellite component detection model. Finally, the proposed RSD model and original Mask R-CNN are tested on the same established test set. The experimental results show that the proposed detection model has higher precision, recall rate, and $F1$ score. Therefore, the proposed approach can effectively detect satellite components, based on optical images.

1. Introduction

With the rapid development of space technology, accomplishing many space tasks, such as autonomous rendezvous and docking in space and space target capture, requires a satellite to accurately identify the main body or components of the target satellite to obtain the target position and attitude information [1–5]. Detecting the components of the target satellite belongs to the field of target detection, whose goal is to accurately detect the location and type of satellite components, such as solar wings, antenna, and docking devices. Accomplishing this goal is a key problem in the field of computer vision, and it can be solved by considering the similarity of the object features such as background, texture, and shape. However, the task remains challenging due to the differences between the target individuals [6–8].

The methods to detect satellite components, which were developed before the development of deep learning, can be divided into image matching and traditional target detection methods. Mingdong et al. [9] used the image matching algorithms to detect space objects. Zhi et al. [10] first preprocessed and segmented the images and later extracted the features by using Surf. Finally, the fractal clustering model of the satellite components was used to perform the component classification. Cai et al. [4, 11, 12] adopted the traditional target detection method to detect the triangle bracket of a solar wing and proposed different improvements in the feature extraction stage. In the traditional target detection approach, each step is optimised independently, and the global optimisation of the whole method cannot be performed. Furthermore, the computational efficiency of this approach is low [6].

After the successful application of the deep convolution neural network (DCNN) in image classification, the target detection task entered a period of rapid development [13–15]. When a DCNN is used for target detection

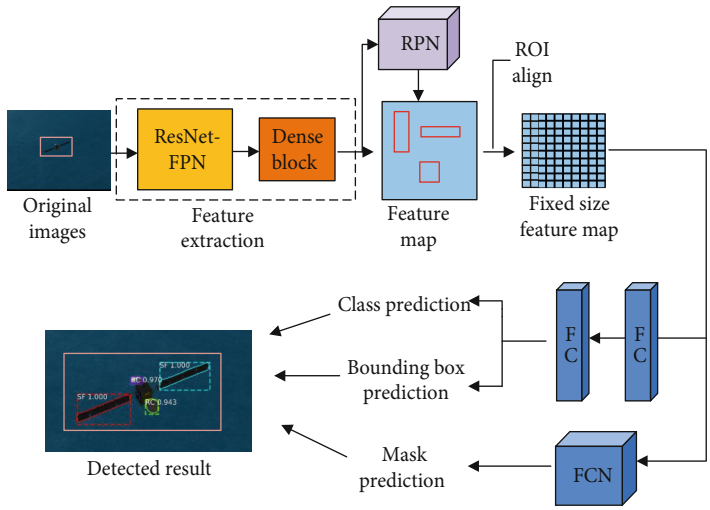


Figure 1: Network structure of RSD.

and target recognition, it exhibits a high robustness to the interference of the environment and satisfactory generalisation ability, and it can realise target detection with a high accuracy [6–8, 14, 15]. At present, the detection model is mainly divided into a single-stage model and a two-stage model, such as YOLO, SSD, and Mask R-CNN [16–21]. Zeng and Xia [22] proposed a space target recognition method based on the DCNN and [23] proposed a kind of target feature point extraction method based on deep learning to realise the detection and location determination of a docking node.

The abovementioned detection methods based on the CNN have two shortcomings: (1) The methods can only detect the classification and location of target satellite, but cannot accurately detect the position and edge contour of multiple components of the target satellite at the pixel level. (2) In terms of dataset construction, the abovementioned studies did not systematically sample the satellite; therefore, the information of the dataset was not sufficient. To address these two problems, this work makes the following contributions. (1) We improved the feature extraction structure of the Mask R-CNN and proposed an R-CNN based method (RSD) to detect satellite components. This detection model can realise the detection of multiple components of the satellite in pixel level, and the object can be segmented into pixel instances to achieve a higher detection precision. (2) For the dataset construction, we first establish a satellite image dataset, which contains images of 92 kinds of satellites from multiple perspectives and multiple orbital positions. The samples in the dataset can fully represent the appearance characteristics of each satellite and reduce the difference between the dataset and real scene images, which can provide effective sample support for the training of the model.

The remaining paper is organised as follows. Section 2 provides a general overview of the RSD and introduces the construction methods of the satellite dataset. The experiment details and test results of the RSD are presented in Section 3, and Section 3 analyses and discusses the test results. Finally, the conclusions are presented in Section 4.

2. Detection of Satellite Components

During the movement of a satellite, the screening and brightness of the components change constantly, which is not conducive to realise accurate detection. In addition, the number of samples in the established dataset is small. Considering these factors, to achieve a higher precision for the detection of satellite components, this paper proposes an R-CNN-based model to detect satellite components. Our RSD model is an improved version of the Mask R-CNN [21]. This paper combines the network architecture of DenseNet and ResNet with the idea of the FPN [25] and applies it to the backbone of our improved Mask R-CNN. The prediction heads consist of three branches, which are used for classification prediction, regression box prediction, and generation mask.

Figure 1 shows the overall process of the RSD algorithm.

The steps in the RSD to detect the satellite components are as follows:

Step 1. Input the image to be detected.

Step 2. The initial feature extraction of the image is performed using the ResNet-FPN. Further feature extraction is later performed by using the dense block. The feature maps of each scale are upsampled and concatenated, and the concatenated feature maps are input into the dense block for further feature extraction. Finally, the system outputs the corresponding feature maps.

Step 3. Input the feature map into the RPN network structure and generate several filtered accurate ROIs through the proposal layer.

Step 4. These ROIs were processed using the ROI align to match the pixels in the original image with those in the feature map and extract the corresponding target features in the shared feature map.

Step 5. These ROIs are input into the FC and FCN for target classification and instance segmentation, respectively. Finally, the classification results, regression box, and segmentation mask are generated. The classification results and position information of the satellite components can be obtained.

2.1. Feature Extraction Structure. In the DCNN, as the depth increases, the feature propagation between each layer degrades, resulting in the loss of information in the transmission process. In addition, after the feature map is upsampled and multiscale concatenation is performed, the semantic information may be obscured, and a large number of parameters may be introduced. To overcome these problems, in this study, the idea of DenseNet was applied to the ResNet-FPN, and the features extracted from the ResNet-FPN were further processed by using a densely connected convolutional structure.

Figure 2 shows the feature extraction structure of the RSD, which is mainly composed of two parts, namely, the ResNet-FPN and densely connected convolution block (DB).

2.1.1. ResNet-FPN. Deepening the neural network can improve the generalisation performance of the model [27]; however, increasing the number of layers may lead to problems such as gradient disappearance or gradient explosion, which makes it difficult to train the deep neural network. The ResNet structure can effectively solve the above problems [14, 26]. In the task of satellite component detection, multiscale detection is extremely critical, especially for small objects, such as small parts of the satellite. However, at such large distances, the antenna and other components account for only 1/2000 of the total area of the image, and thus they are often difficult to detect. Therefore, we adopt the FPN structure and ResNet-50 as the backbone. This structure can fuse the features of all the levels; thus, this structure has both a strong semantic information and strong spatial information, which can improve the precision and speed of detection of small objects at multiple scales.

As shown in Figure 2, the ResNet-FPN consists of three parts: the bottom-up connection, top-down connection, and horizontal connection. The bottom-up connection pertains to the process of feature extraction with ResNet as the backbone, the top-down connection pertains to the process of upsampling from the top layer, and the transverse connection pertains to the fusion of the upsampling feature map and the feature map of the same size generated from the bottom-up process.

The image is input into the ResNet-FPN, assuming that the size of the input image is 512×512, and the number of channels is 64. After extracting the structural features of the ResNet-FPN, the feature maps M2, M3, M4, M5, and M6 are output. The sizes of these maps are 64×64, 32×32, 16×16, 8×8, and 4×4, respectively, and the number of channels is 256. Next, M2, M3, …, and M6 are input into the subsequent densely connected convolution block to further extract the features.

2.1.2. Densely Connected CNN. A deep neural network can autonomously learn the characteristics of data through a large number of sample data. However, when the amount of sample data is limited, the trained model usually has an inferior generalisation ability, and the traditional CNN also has problems such as gradient disappearance, large number of parameters, and parameter redundancy [14, 24]. To solve these problems, we combine the idea of the dense connection with the feature extraction structure of the ResNet-FPN, deep feature extraction is performed for the satellite components, which enhances the feature propagation between the layers and alleviates the obscurity of the semantic caused by the sampling up of the feature maps and multiscale feature map fusion.

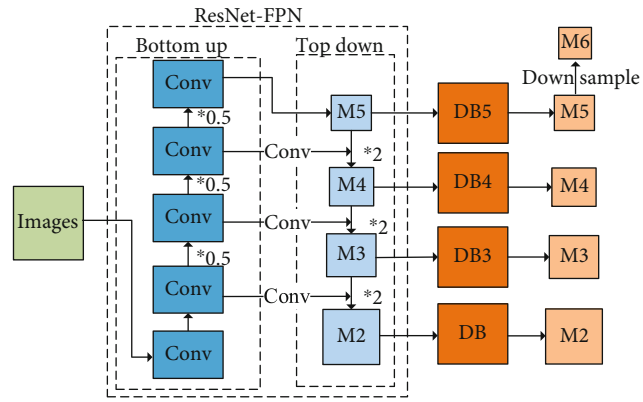


Figure 2: Feature extraction structure of the RSD.

Figure 3 shows the densely connected convolution block (dense block) in the feature extraction structure used in this paper, which consists of five layers. The first layer includes only the convolution layer, and the other layers all contain a batch normalisation layer (BN), modified linear activation layer (ReLU), and convolutional layer (CONV).

In the dense block, the input of each layer is related not only to the output of the previous layer but also to the output of all the previous layers, which serves as the input. This structure can make full use of all the feature information included in the previous layer, considerably reduce the connection distance between the front and back layers, and effectively solve the problem of gradient disappearance with the deepening of the network [14].

The generation formula for the feature graph of layer i is as follows:

$$X_i = H_i([X_0, X_1, X_2, \cdots, X_{i-1}]). \quad (1)$$

Here, $[X_0, X_1, X_2, \cdots, X_{i-1}]$ represents the concatenation of the feature graph generated in layer $0, 1, \cdots, i-1$ as the dimension of the channel. H_i is a composite function corresponding to the batch normalisation (BN), modified linear element (ReLU), and convolution (Conv). Assuming that the number of feature graphs transmitted by each nonlinear transformation H is K, and the number of feature graphs at *layer 0* is K_0, the number of input feature graphs at *layer i* is $K_0 + (i\text{-}1) * K$; K is also known as the growth rate.

The specific structure of the Dense Block is described in Table 1. The convolution layer DB_Conv1 does not contain the BN and ReLU layers. This layer is set as such to reduce

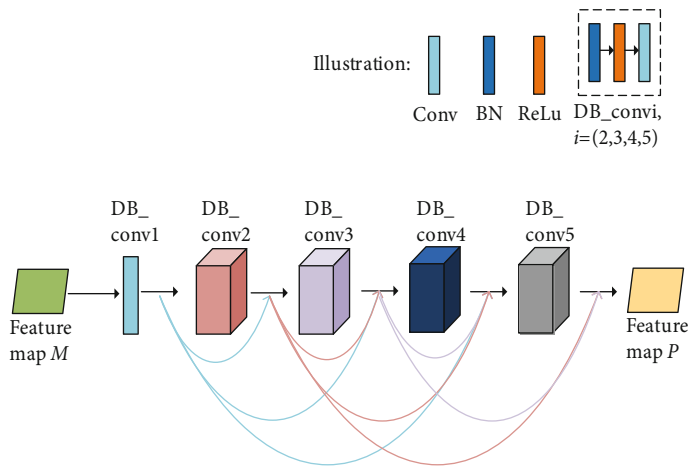


Figure 3: Dense block of the improved Mask R-CNN.

Table 1: Dense block architectures.

Layer	Input	Kernel	Stride	Output size
DB_Conv1	128 × 128, 256	3 × 3	1	128 × 128, 128
DB_Conv2	128 × 128, 128	3 × 3	1	128 × 128, 32
Concat_1	128 × 128, 128 128 × 128, 32			128 × 128, 160
DB_Conv3	128 × 128, 160	3 × 3		128 × 128, 32
Concat_2	128 × 128, 128 (128 × 128, 32) × 2			128 × 128, 192
DB_Conv4	128 × 128, 192	3 × 3	1	128 × 128, 32
Concat_3	128 × 128, 128 (128 × 128, 32) × 3			128 × 128, 224
DB_Conv5	128 × 128, 224	3 × 3	1	128 × 128, 32
Concat_4	128 × 128, 128 (128 × 128, 32) × 4			128 × 128, 256

the number of channels to avoid the subsequent feature extraction process, which incurs a large computation cost and several parameters. Assuming that the size of the input feature graph M is 128 × 128 and the number of channels is 256, the first convolution layer is used to reduce the number of channels to 128, and the features are later extracted and fused through the following four layers. Finally, the output feature graph P is output with the number of channels being 256, and this graph is input to the subsequent RPN and prediction head to realise the target detection.

2.2. Dataset Construction. Because of the limited number of satellite images available for the real scene and uneven distribution of the visual angle, the model cannot satisfy the needs of model training and learning and exhibits an inferior performance. Therefore, in this paper, the images of satellite under various perspectives and orbital positions were collected using the software System Tool Kits (STK), which is an analytical software developed by American Analytical Graphics in the aerospace domain, and these images served as the basis of the dataset.

The overall process of constructing the dataset of the satellite components is shown in Figure 4. First, the images of the satellite are collected to establish a dataset containing rich information of the satellite. Second, the components of the satellite in the image are labelled. As an example, the antenna and solar wing were taken (denoted as components I and II, respectively) as the target components to be detected. The constructed dataset contained 1288 samples. The dataset was randomly divided into a training set and test set in proportion.

By using the reasonable multiangle and multiorbit position sampling strategy, a large number of satellite images can be collected, thereby providing more systematic materials for the establishment of the subsequent datasets. As shown in Figure 5, to make the perspective distribution of the image in the dataset more uniform and reasonable, this paper sampled the appearance of the satellite from the following 14 perspectives.

Using the above method, the appearance of the satellite can be sampled uniformly from 14 perspectives. Under the sampling of this uniform perspective, the satellite will have component occlusion and overlap, which fully simulates the possible situation of the real scene.

To ensure that the satellite data set contains more effective information and to better overcome the differences of the simulated images and real scene images, in the image collection, we take the position of the satellite as one of the factors to be considered and adjust the relative position of the satellite and the sun. As shown in Figure 6, the satellite is sampled at two orbital positions to induce changes in the light intensity and imaging effect to better simulate the real scene.

Figure 7 is the comparison between the simulated image (a) and the real image (b) of the ISS (International Space Station). In terms of the outline, shape, and texture of components, taking the solar wing as an example, the solar wing of the two images is almost consistent. In terms of color, the color of each component in the two images is very similar, except for the difference in brightness. Taking orbit positions into consideration can make the information of the dataset more reasonable and effective and considerably

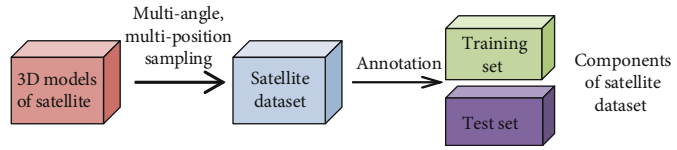


Figure 4: Overall process of dataset construction for the target components of satellites.

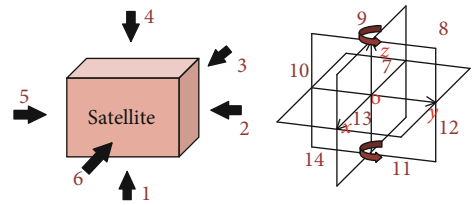


Figure 5: 14 sampling perspectives. (a) 6 sampling perspectives are 1, 2, …, and 6. (b) 8 sampling perspectives are 7, 8, …, and 14. The 3d coordinate was taken as the datum, and origin o was the centre of the satellite. The 3d coordinate was divided into eight octagons. The octagon determined by the positive half axis of the x, y, and z axes was the first octagon. The 11th to 14th octant was below the xoy plane, and the octant was marked counterclockwise.

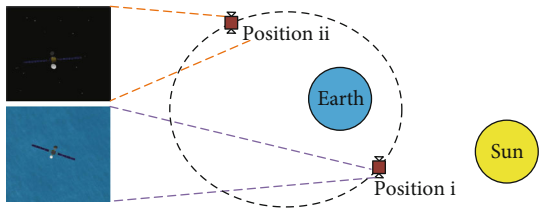


Figure 6: Multiposition of orbit sampling. When the satellite is in position I, the satellite's image is clear and the contrast is high. However, when the satellite is in position II, the satellite's image is not clear and the contrast is low.

reduce the differences between the simulation images and real images, which provides a concrete foundation for the model training, Figure 8 shows several samples in the final built dataset.

2.3. Loss Function. The loss function L of our model is calculated using formula (2), which, similar to that for the Mask R-CNN, consists of two parts [20]: the loss function L_{RPN} for training the RPN, and the loss function $L_{\text{Mul-Branch}}$ for training the multitask branches:

$$L = L_{\text{RPN}} + L_{\text{Mul-Branch}}. \tag{2}$$

L_{RPN} is calculated using formula (2), which includes the loss function of the anchor category L_{cls}. The loss function of the regression box L_{reg} is the softmax cross-entropy loss, and L_{reg} is the smooth $L1$ loss.

$$L_{\text{RPN}} = \frac{1}{N_{\text{cls}}}\sum_{\text{i}} L_{\text{cls}}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{reg}}}\sum_{i} p_i^* L_{\text{reg}}(t_i, t_i^*). \tag{3}$$

Here, p_i represents the classification probability of anchor i, p_i^* represents the ground-truth label probability of anchor i, t_i represents the difference between the predicted regression box and ground-truth label box, and t_i^* represents the difference between the ground-truth label box and positive anchor.

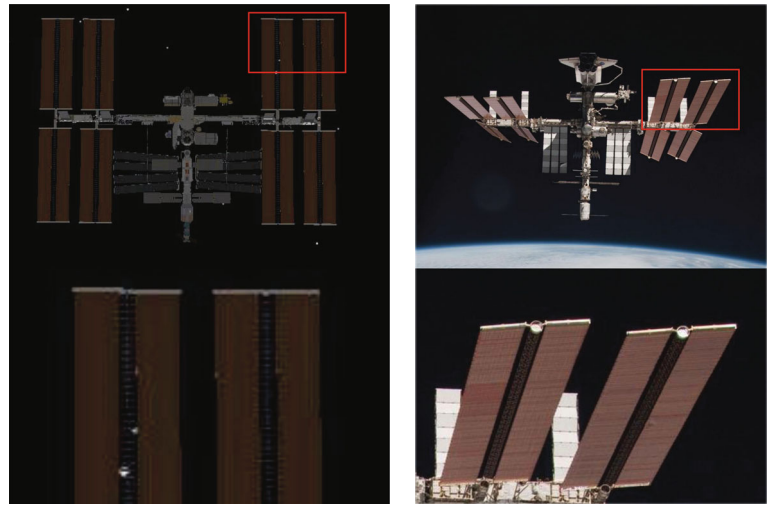
$L_{\text{Mul-Branch}}$ is calculated using formula (3)

$$L_{\text{Mul-Branch}} = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}. \tag{4}$$

Here, the loss function of classification L_{cls} is the cross-entropy loss; the loss function of the regression box L_{box} is the smooth $L1$ loss. The loss function of mask L_{mask} is the average binary cross-entropy loss.

2.4. Model Training. All parts of the RSD are jointly trained. The main steps of the training are as follows: (1) First, the established dataset of the satellite components is divided into a training set and test set, with proportions of 80% and 20% (numbers of 1033 and 255), respectively; (2) the super parameters of the model are set; (3) the weights in the RSD are initialised; (4) the training samples and labels are fed into the model, and the loss function is calculated and back propagation is performed; (5) the model is trained until the value of the loss function L_{train} remains stable, and the learning rate is adjusted to continue training for a period of time. Training is repeated until L_{train} does not exhibit a significant decline.

The experimental parameters of the RSD model and Mask R-CNN are set as follows: The SGD optimiser is adopted in the optimisation method, the momentum is set as 0.9, the weight decay is set as 0.0001, and the nonmaximum suppression (NMS) threshold is set as 0.7. This paper sets the size of the anchor to 4×4, 12×12, 16×16, 32×32, and 56×56, and the ratio of the anchor is set as 1 : 1, 2 : 1, and 1 : 2 to better fit the ground truth of the target. Setting the anchor to have multiple sizes can help better detect the components of multiple sizes.



(a) (b)

Figure 7: Comparison between simulated image and real image.

Figure 8: Several samples in the built dataset.

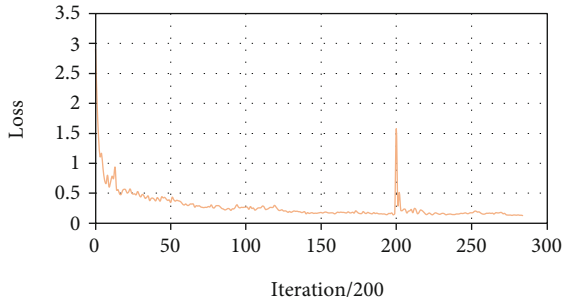


Figure 9: Change curve of the loss function during training.

The initialisation of the model weight is performed in two phases: The first phase involves the initialisation of the ResNet-FPN and the prediction head, and the second phase involves the initialisation of the dense block. The first phase uses the pretrained model on the MS COCO dataset for the weight initialisation, and the second part performs the uniform distribution initialisation with a lower and upper boundary of -0.05 and 0.05, respectively. As shown in Figure 9, the initial learning rate is 0.001, and L_{train} is approximately 0.15 after 40 K iterations. The learning rate is adjusted to 0.001/10, and the number of iterations is 17 K.

Table 2: Class and number of target components.

Class	Number
Component *I*	518
Component II	197

At this instant, L_{train} is approximately 0.14, L_{train} remains nearly constant, and the training is stopped. The completed training requires approximately 13 h, and the optimised R-CNN based satellite component detection model is obtained.

3. Experimental Results and Discussion

All the experiments (training and testing of the model) in this paper were conducted on the same server under the deep learning development framework of TensorFlow and Keras, with the PC configuration as follows: Inter Xeon e5-2620 v4 2.10 GHz ∗32 CPU and RTX 2020Ti GPU.

3.1. Evaluation Index. In this paper, the precision, recall, and F1 score were used to evaluate the model performance. Precision describes the ability of a classification model to return only relevant objects. The recall describes the ability of a

Table 3: Confusion matrix.

		Prediction class		
		Component *I*	Component II	Background
Ground truth	Component *I*	513	0	5
	Component II	3	169	25
	Background	25	19	/

Table 4: Evaluation of the algorithm performance.

Method	Precision for component *I*	Recall for component *I*	Precision for component II	Recall for component II	Overall precision	Overall recall
RSD	0.95	0.99	0.9	0.86	0.93	0.93
Mask R-CNN	0.91	0.98	0.89	0.78	0.9	0.88

classification model to identify all the relevant targets. The $F1$ score is the harmonic average of the precision and recall rate, and a higher value corresponds to a better detection performance of the model.

The precision is calculated using formula (5):

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{5}$$

where TP represents the number of samples that are positive and tested positive in the test set, while FP represents the number of samples that are negative but tested positive.

The recall is calculated using formula (6):

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{6}$$

where FN represents the number of samples that are positive but tested negative.

The $F1$ score is calculated using formula (7):

$$F1 = \frac{2P \cdot R}{P + R}, \tag{7}$$

where P and R denote the precision and recall, respectively.

3.2. Test and Results. The RSD and the original Mask R-CNN model were used to detect 255 test samples (20% of the dataset). The number of components in the test set is shown in Table 2.

The confusion matrix of the detection results obtained using the proposed method is shown in Table 3.

The precision and recall rate can be obtained using the confusion matrix. As shown in Table 4, the overall precision and recall rate of the improved Mask R-CNN are higher than those of the Mask R-CNN network. The overall precision, recall, and F1 score of the proposed method are all 0.93, respectively. Compared with the Mask R-CNN, the proposed model exhibits a precision improved by 3% and F1 score improved by 4%.

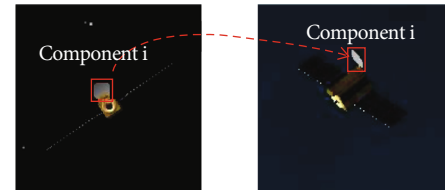
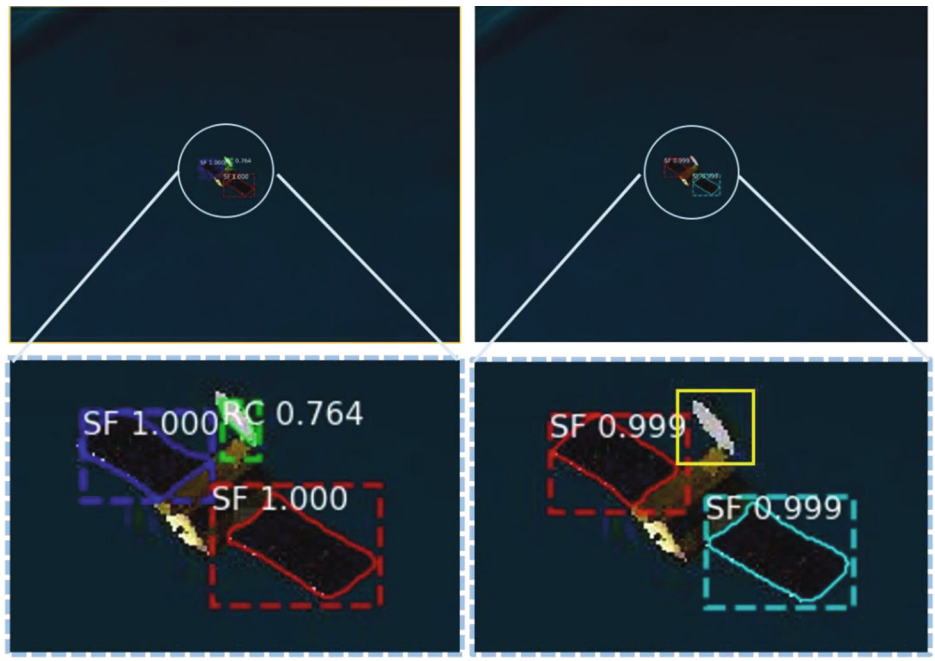


Figure 10: Deformation of component II. In attitude I, component I is round and the bounding rectangle is a square; however, in attitude II, component I appears as an oval.

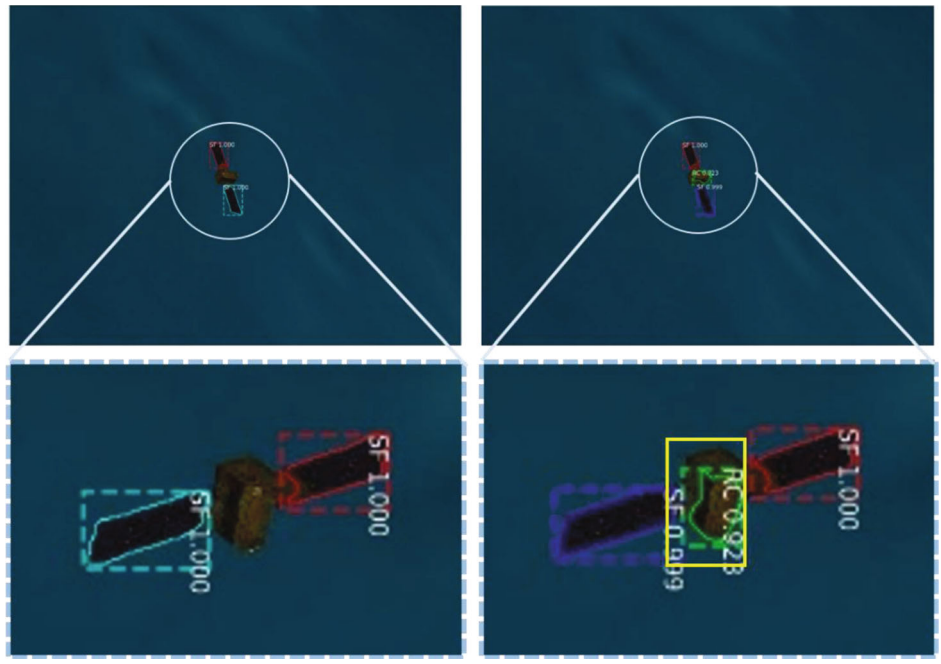
3.3. Discussion. The detection results of the Mask R-CNN for the satellite components and background involved several errors. In contrast, the feature extraction structure of the proposed method could integrate the features of various levels, ensuring that they have a strong semantic information and strong spatial information simultaneously to provide an effective feature map for the subsequent detection and better distinguish the target components and background. The close-up graph of the test results was considered to discuss the improvement in the precision and recall.

Compared with the component I and the main body of the satellite, the area of the component II is small. In addition, because the satellite is in a state of constant motion in a variety of attitudes, the shape of the components changes. Figure 10 shows the imaging of the same satellite in different attitudes. Under ideal conditions, this component I is circular, and in most cases, it is elliptical. The deformation of the target caused by the change in the attitude directly affects the recall. Figure 11(a) shows the detection result pertaining to attitude II, as shown in Figure 10, obtained using the proposed method and Mask R-CNN.

Figure 11 shows the comparisons on 2 samples, where it can be found that the proposed RSD can accurately detect all target components. Mask R-CNN existed a miss-detection of the target component and incorrectly detected the background as one of the targets. The RSD is relatively better than the Mask R-CNN in terms of the precision and recall rate; it can not only identify more target components but also reduce the false identification and classification of the components and background. Under the condition that



(a)



(b)

Figure 11: Comparisons between the proposed one and Mask R-CNN. (a, b) are the detection results of the two samples, where the left side of each sample is the detection result of RSD, and the right side is the detection result of Mask R-CNN. The bottom row is a close-up of the test results, with the yellow box representing the incorrectly detected part.

the component does not undergo severe deformation, the RSD can detect the target component well and classify it correctly.

4. Conclusion

This paper proposes a satellite component detection method based on the region-based convolutional network and establishes a satellite dataset. The results of the performed contrast experiment proved that the proposed RSD model exhibited a better performance than that of the Mask R-CNN. The specific contributions were as follows: (1) The satellite dataset constructed in this paper contained abundant satellite information. A total of 92 kinds of satellites were sampled uniformly from 14 angles and 2 orbital positions to ensure that the dataset could fully simulate the imaging of a satellite in

a variety of attitude and illumination brightness conditions. (2) In this paper, the Dense Net and ResNet-FPN were combined to improve the feature extraction structure. The image was first extracted through the ResNet-FPN and later deeply extracted through the dense block to enhance the feature transmission between each layer. The experiments indicated that the proposed model exhibited a better performance than that of the Mask R-CNN. However, the performance of our RSD in detecting severely deformed components is still not good, and further research is still needed in the future work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request. And in the future, the data used to support the findings of this study will be published online.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors thank STK developers for providing the data used in the paper. This research was carried out by the Beijing Information Science & Technology University, Beijing, China, under the following project, Stable Support Project of State Administration of Science, Technology and Industry for National Defence (HTKJ2019KL502008), Beijing Municipal Education Commissions capacity building for science and technology innovation services-basic research business fees (scientific research) (Project No.1 PXM2018-014224-000032, Project No.2 PXM2019-014224-000026).

References

[1] R. Volpe and C. Circi, "Optical-aided, autonomous and optimal space rendezvous with a non-cooperative target," *Acta Astronautica*, vol. 157, pp. 528–540, 2019.

[2] H. Zhang, C. Zhang, Z. Jiang, Y. Yao, and G. Meng, "Vision-Based Satellite Recognition and Pose Estimation Using Gaussian Process Regression," *International Journal of Aerospace Engineering*, vol. 2019, 20 pages, 2019.

[3] L. Liu, G. Zhao, and Y. Bo, "Point cloud based relative pose estimation of a satellite in close range," *Sensors*, vol. 16, no. 6, p. 824, 2016.

[4] F. Zhang, P. F. Huang, L. Chen, and J. Cai, "Line-based simultaneous detection and tracking of triangles," *Journal of Aerospace Engineering*, vol. 31, no. 3, article 04018013, 2018.

[5] X. Zhang, Z. Jiang, H. Zhang, and Q. Wei, "Vision-Based Pose Estimation for Textureless Space Objects by Contour Points Matching," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2342–2355, 2018.

[6] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[7] L. Liu, W. Ouyang, X. Wang et al., "Deep learning for generic object detection: a survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.

[8] M. Yazdi and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: a survey," *Computer Science Review*, vol. 28, pp. 157–177, 2018.

[9] M. Yang, J. Wang, J. Jianjun, L. Zhang, and J. Qiang, "Research on technologies of space area targets high-precision tracking based on SWAD algorithm," *Infrared and Laser Engineering*, vol. 45, no. 2, pp. 0228002–0228002, 2016.

[10] X. Zhi, Q. Hou, W. Zhang, and X. Sun, "Optical identification method of space typical targets based on combined multi-feature metrics," *Journal of Harbin Institute of Technology*, vol. 10, p. 6, 2016.

[11] J. Cai, P. Huang, L. Chen, and B. Zhang, "A fast detection method of arbitrary triangles for Tethered Space Robot," in *In 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 120–125, Zhuhai, China, December 2015.

[12] L. Chen, P. Huang, J. Cai, Z. Meng, and Z. Liu, "A non-cooperative target grasping position prediction model for tethered space robot," *Aerospace Science and Technology*, vol. 58, pp. 571–581, 2016.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[14] M. Z. Alom, T. M. Taha, C. Yakopcic et al., "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, 2019.

[15] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.

[16] 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*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[17] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Computer Vision – ECCV 2016*, pp. 21–37, Springer, Cham, 2016.

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

[19] R. Girshick, "Fast r-cnn," in *In Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, Santiago, Chile, December 2015.

[20] 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.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *In Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, Venice, Italy, October 2017.

[22] H. Zeng and Y. Xia, "Space target recognition based on deep learning," in *In 2017 20th International Conference on Information Fusion (Fusion)*, pp. 1–5, Xi'an, China, July 2017.

[23] I. S. Fomin, A. V. Bakhshiev, and D. A. Gromoshinskii, "Study of using deep learning nets for mark detection in space docking control images," *Procedia Computer Science*, vol. 103, pp. 59–66, 2017.

[24] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[27] H. T. Cheng, L. Koc, J. Harmsen et al., "Wide & Deep Learning for Recommender Systems," in *In Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10, New York, NY, USA, September 2016.