

Toward the recognition of spacecraft feature components: A new benchmark and a new model

Linwei Qiu¹, Liang Tang²(✉), and Rui Zhong¹

1. School of Astronautics, Beihang University, Beijing 100191, China

2. Key Laboratory of Space Intelligent Control Technology, Beijing Institute of Control Engineering, Beijing 100094, China

ABSTRACT

Countries are increasingly interested in spacecraft surveillance and recognition which play an important role in on-orbit maintenance, space docking, and other applications. Traditional detection methods, including radar, have many restrictions, such as excessive costs and energy supply problems. For many on-orbit servicing spacecraft, image recognition is a simple but relatively accurate method for obtaining sufficient position and direction information to offer services. However, to the best of our knowledge, few practical machine-learning models focusing on the recognition of spacecraft feature components have been reported. In addition, it is difficult to find substantial on-orbit images with which to train or evaluate such a model. In this study, we first created a new dataset containing numerous artificial images of on-orbit spacecraft with labeled components. Our base images were derived from 3D Max and STK software. These images include many types of satellites and satellite postures. Considering real-world illumination conditions and imperfect camera observations, we developed a degradation algorithm that enabled us to produce thousands of artificial images of spacecraft. The feature components of the spacecraft in all images were labeled manually. We discovered that direct utilization of the DeepLab V3+ model leads to poor edge recognition. Poorly defined edges provide imprecise position or direction information and degrade the performance of on-orbit services. Thus, the edge information of the target was taken as a supervisory guide, and was used to develop the proposed *Edge Auxiliary Supervision DeepLab Network (EASDN)*. The main idea of EASDN is to provide a new edge auxiliary loss by calculating the L2 loss between the predicted edge masks and ground-truth edge masks during training. Our extensive experiments demonstrate that our network can perform well both on our benchmark and on real on-orbit spacecraft images from the Internet. Furthermore, the device usage and processing time meet the demands of engineering applications.

KEYWORDS

spacecraft
deep learning
feature components
semantic segmentation

Research Article

Received: 8 September 2020

Accepted: 5 May 2021

© The Author(s) 2021

1 Introduction

Spacecraft surveillance and recognition systems have found application in on-orbit maintenance, space docking, and other orbit services. When offering on-orbit services, one of the most significant tasks a satellite has to complete is the recognition of spacecraft feature components. Several countries have developed miscellaneous spacecraft surveillance and recognition systems since the start of the

21st century. For example, to mitigate the growing risk of space debris, the Air Force Research Laboratory (AFRL) implemented a plan named Autonomous Nanosatellite Guardian for Evaluating Local Space (ANGELS) in 2005. Special satellites were placed in geostationary orbit to conduct continuous surveillance and accurate detection of satellite targets in the near-space area. These satellites extract important attributes of the target and communicate with their own surveillance platform for

✉ tl614@sina.com

feedback [1]. In 2014, the U.S. Air Force implemented the Geosynchronous Space Situational Awareness Program (GSSAP), which included four satellites to monitor other satellites in space [2–4]. Canada launched a satellite named NEOSSAT in 2013 to detect asteroids in near-Earth space. It was also designed to track high-orbit satellites and space debris [5]. These attempts have many shortcomings: (1) traditional detection approaches such as radar or infrared observations have cost and energy supply restrictions; (2) they do not have any on-orbit ability to process data acquired by the sensors. In other words, they have to communicate with the ground control center, leading to signal delays and substantial human costs.

Compared with traditional detection methods, image recognition is an easy but relatively effective method for obtaining position and direction information. Tang and Zou proposed a non-cooperative spacecraft recognition approach based on the fusion of local features [6]. Zhi *et al.* and Hu also introduced combined multi-feature metrics towards the recognition of spacecraft feature components [7, 8]. However, these traditional image-processing studies, which depend on elaborate designs, are geared towards certain types of spacecraft. Moreover, they tested their methods on a few images, rather than on a standard dataset. Recently, deep learning technology has achieved great success in the field of artificial intelligence [9]. Convolutional neural networks (CNNs) have yielded outstanding results in the computer vision field and refresh almost all of the records [10]. These methods have rendered the recognition of spacecraft feature components feasible both in algorithms and engineering applications. Zhang *et al.* used CNN and long short-term memory network (LSTM) for space target recognition [11, 12]. However, they attempted to recognize a space target based on radar echo signals. To the best of our knowledge, a specialized CNN-based deep learning model has not yet been proposed for the recognition of spacecraft feature components in optical images. Even more importantly, the lack of substantial on-orbit images makes it difficult to train or evaluate such a model.

This paper introduces a new labeled dataset with thousands of artificial on-orbit spacecraft images. Base images are derived from 3D Max and STK software, and take both the modeling accuracy and costs into consideration. These images include many types of

satellites (such as low earth orbit, medium earth orbit, and geosynchronous earth orbit satellites) and many postures. A degradation algorithm including geometric transformation, noise, and image blur was developed to simulate on-orbit illumination conditions and imperfect camera observation. Thousands of artificial spacecraft images were generated by applying this degradation algorithm to the base images. The feature components, such as panels of the spacecraft in all degraded images, were annotated manually to evaluate the performance of the recognition algorithms.

Our work mainly focuses on semantic segmentation for the recognition of spacecraft feature components, which involves labeling the images with pixel-level semantic information [13]. Traditional image segmentation methods mainly include four classes: (1) *threshold segmentation*, in which an image is divided into several features based on thresholds with several pixel values; (2) *edge detection segmentation*, in which the pixel gray level or discontinuous color of the edge is applied to detect the area and achieve complete image segmentation; (3) *regional growth segmentation*, the essence of which is to connect pixels with similar features, such as grayscale features, shape features (e.g., scale-invariant feature transform (SIFT) [14], histogram of oriented gradient (HOG) [15]), and other features, whereas it is challenging to select features; (4) *graph theory*, which entails mapping an image to a graph, and transforming the segmentation of the image into the partitions of the graph. This includes the Markov random field method based on graph cut [16] and the random walk method [17].

All of these traditional methods have specific disadvantages; hence, it is not easy to use one of these approaches to recognize spacecraft feature components when having to overcome different spacecraft postures, backgrounds, and light conditions. In contrast to these methods, the deep learning method can automatically learn, extract, and accurately represent data features. CNN-based semantic segmentation establishes mapping from pixels to semantics without the need for posterior manual work. FCN [18], SegNet [19], and U-NET [20] are classical models of semantic segmentation using CNNs. E-Net [21], and Link-Net [22] are light networks created specifically for tasks that require low-latency operations, which have fewer parameters and faster speeds. DeepLab V3+ [23], a state-of-the-art model in the current semantic segmentation domain, enables

segmentation precision across multiple datasets. However, our initial work showed that the DeepLab V3+ model has poor edge recognition abilities. The feature components of a spacecraft usually have regular shapes. Poorly defined edges provide imprecise position or direction information and lower the performance of on-orbit services.

In this paper, we propose a border-supervised network based on the DeepLab V3+ model to address this problem. We leveraged the edge information of the target as a supervisory guide by proposing the *Edge Auxiliary Supervision DeepLab Network (EASDN)* algorithm, which calculates the L2 loss between the predicted edge masks and ground-truth edge masks during training. Subsequently, we conducted extensive experiments to demonstrate that our networks perform well both on our benchmark and on real on-orbit spacecraft images from the Internet. Our light version, which is less computationally expensive, can also meet the demands of device usage and time in engineering applications.

2 Spacecraft feature component dataset

Owing to the particularity of spacecraft, it is difficult to obtain real on-orbit images from the Internet, and those in the public domain are almost all simulation images. Thus, we chose to produce base spacecraft images using three-dimensional (3D) animation rendering and production software. We enlarged these base images to render an on-orbit effect by applying a degradation algorithm. In addition, the spacecraft feature components in all of these images were labeled.

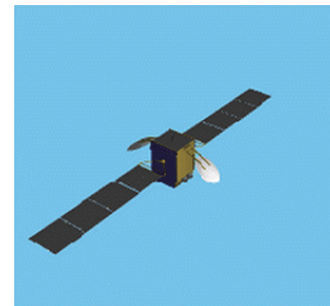
2.1 Base images

Zhang *et al.* established the BUAA-SID 1.0 space target database [24], which is based on 3D models of space targets and utilizes 3D Max software to render and generate a full-view simulation image sequence of a space target. However, the number of images is limited, and the rendering effect does not closely correspond to the real on-orbit environment. Our base on-orbit images of spacecraft stem from 3D Max and STK software, both of which exploit the 3D model of the spacecraft to obtain the simulation image. The difference lies in the image quality and computational cost. It takes a significant amount of time for 3D Max to render a 3D model of the spacecraft using a raytracer, although the quality of the rendering image is high and close to that of real on-orbit

satellites. STK, on the other hand, generates simulation images without distributed rendering. Capturing 3D models directly in a space scene from STK is readily achievable and saves time, but the outputs are not realistic. Figure 1(a) is the rendering result produced with 3D Max, and Fig. 1(b) is the result obtained with STK. The 3D model in Fig. 1(b) is composed of regular objects such as balls, cubes, conical surfaces, and so on. They lack textures, reflections, and shading, which are necessary to simulate realistic images.



(a) 3D Max rendering result



(b) Spacecraft model in STK

Fig. 1 Rendering results for satellite “a2100” using 3D Max and STK software.

2.1.1 3D Max

3D models of spacecraft can be built and rendered from essential 3D objects and components using 3D Max software. Nevertheless, we need to survey the geometric structural parameters of the spacecraft. Thus, it would be more efficient to utilize the models in the STK satellite library and import the files of the STK model (which would need to be converted to the “.lwo” format) into 3D Max software for rendering. In total, 16 types of satellites were rendered by the 3D Max software.

2.1.2 STK

STK software was utilized to simulate the on-orbit

operation of a satellite and record high-definition videos with a resolution of 1280×960 pixels at 60 frames per second. Color images were extracted from the videos every five frames as STK output images. Seven types of satellites were produced using STK software. Each satellite contained more than 200 images.

The 3D Max or STK software produces RGB images, whereas real on-orbit satellite images are usually captured in gray mode. Conversion to grayscale is therefore necessary for all images to create a drawing of a real situation. Because human eyes are most sensitive to green and least sensitive to blue, a reasonable grayscale image can be obtained by calculating the weighted average of RGB channels according to the following formula [26]:

$$G(i, j) = 0.299 \times R(i, j) + 0.578 \times G(i, j) + 0.114 \times B(i, j) \tag{1}$$

The resolution of the base images is scaled to 320×240 pixels for a unifying setting. The image size is scaled by the bicubic interpolation method, which retains the visual effect of the original image as much as possible. This method considers not only the variation in the four neighboring pixels, but also the variation of other surrounding pixels, which has a smoothening effect on the edges of the object [27]. Ultimately, we produced images of 23 satellites, which were used to create 4550 gray images. The spacecraft feature components were labeled on the base images.

2.2 Degradation algorithm

The spacecraft image captured on orbit was found on the Internet, as shown in Fig. 2 [25]. As can be seen from the figure, the optical observation image contains a considerable amount of noise, which is especially obvious in the background of the image. As a result of the effect of noise interference, the color of the background is no



Fig. 2 Photographic image of the Shenzhou-7 on-orbit spacecraft captured by a small satellite [25].

longer black, whereas the gray values of these pixels are degraded.

Therefore, the real on-orbit optical image of the spacecraft needs to be processed by adding simulated illumination (light or sunshine), but the influence of noise, distortion, and geometric transformation must be considered. The process followed by the degradation algorithm is illustrated in Fig. 3.

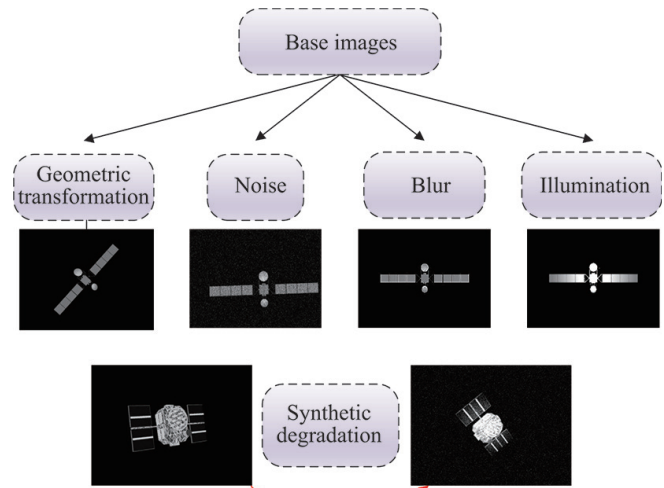


Fig. 3 Our degradation algorithm.

The synthetic degradation shown in Fig. 3 refers to the process of applying several of the aforementioned degradation operations to a base image, to obtain the artificial spacecraft images under different conditions. It is worth noting that the operations of noise, blur, and illumination are solely conducted on the base image and its mask does not need to be changed. On the other hand, for the geometric transformation operation, it is necessary to process the base image and its corresponding labeled image simultaneously. At the same time, the dataset of spacecraft feature components can be enlarged using this degradation algorithm, because one base image can yield many (we set this number to 7 for diversity) artificial on-orbit images by randomly applying different operations in different operating sequences.

2.2.1 Illumination

The illumination added here is stronger at the center of the lobe than that around the lobe. This means that the main lobe is the brightest locally. The relevant expression is as Eq. (2):

$$f(x, y) = k \left[1 - \frac{\sqrt{(x - x_0)^2 + (y - y_0)^2}}{r} \right] \tag{2}$$

where k is the brightness of the center point, (x_0, y_0) are the coordinates of the center point, and r is the radius. When k is sufficiently large, the main lobe of illumination tends to be overexposed, thus losing its original texture. If k is too small, the effect of the illumination operation is not conspicuous. In our dataset, the value of k is 128. The radius is equal to a quarter of the minimum width and height of the input image.

In our dataset, the background in the base image is almost entirely black, and only the spacecraft has gray values. Therefore, it is appropriate to set the center point as the centroid of the base image. The image centroid calculation method was as Eq. (3):

$$\begin{cases} f(x, y) = \begin{cases} 1, & \text{target} \\ 0, & \text{background} \end{cases} \\ x_0 = \frac{\sum xf(x, y)}{\sum f(x, y)} \\ y_0 = \frac{\sum yf(x, y)}{\sum f(x, y)} \end{cases} \quad (3)$$

where (x_0, y_0) is the centroid coordinate, and $f(x, y)$ is the gray value of pixel (x, y) .

The effect of illumination is depicted in Fig. 4.

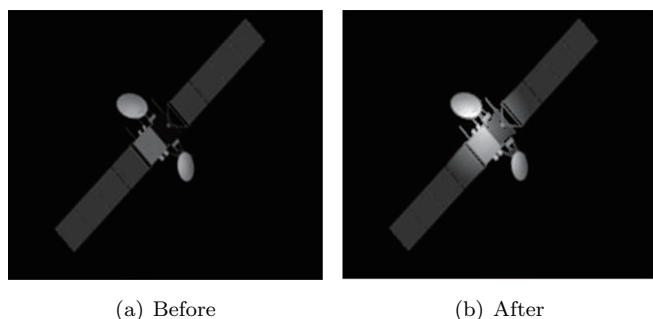


Fig. 4 Effect of illumination.

2.2.2 Noise

To ensure that the images in the dataset more closely resemble those of the real spacecraft in orbit, it is necessary to add suitable noise interference to the images with a black background. According to the probability distribution of the noise, the actual image noise can be divided into Gaussian, Rayleigh, gamma, exponential, and uniform noise. Gaussian noise (i.e., normal noise) is often employed in practice because it is mathematically easy to use in space and frequency domains. The probability density function of the Gaussian random

variable z is given by the following formula [28]:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-(z - \mu)^2 / (2\sigma^2)) \quad (4)$$

where z is the gray value, μ represents the average value of z , and σ is the standard deviation of z . The performance of the noise is shown in Fig. 5. In our dataset, we exploited the Gaussian-distributed additive noise. The mean of the random distribution was 0, and its variance was set to 0.005.

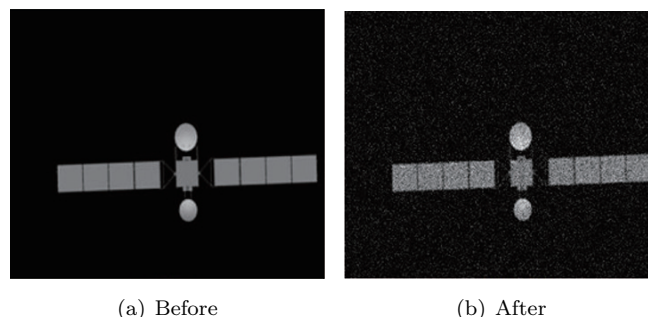


Fig. 5 Effect of noise.

2.2.3 Blur

After the noise operation, the image texture too obviously consists of particles, and the image is still far from the actual situation. Therefore, Gaussian blur was added to ensure the image is closer to the real situation. The blur radius of the Gaussian blur was 0.05.

2.2.4 Geometric transformation

Geometric transformation processing is also required to expand the dataset to include additional feasible images and to counteract overfitting that occurs during training. This processing does not change the pixel value of the image, and it only maps the position coordinates of one image to another. Our geometric transformations include translation, rotation, scaling, and random combinations of these three essential operators, which are illustrated in Fig. 6.

Eventually, synthetic degradation, which includes these operations, was carried out, as shown in Fig. 7. As a result, our final *Spacecraft Feature Component Dataset (SFCD)* consists of 27,240 on-orbit images.

3 Our models

In Section 2, we described the process we followed to construct the new SFCD, which consists of images of on-orbit spacecraft and their corresponding labels. This

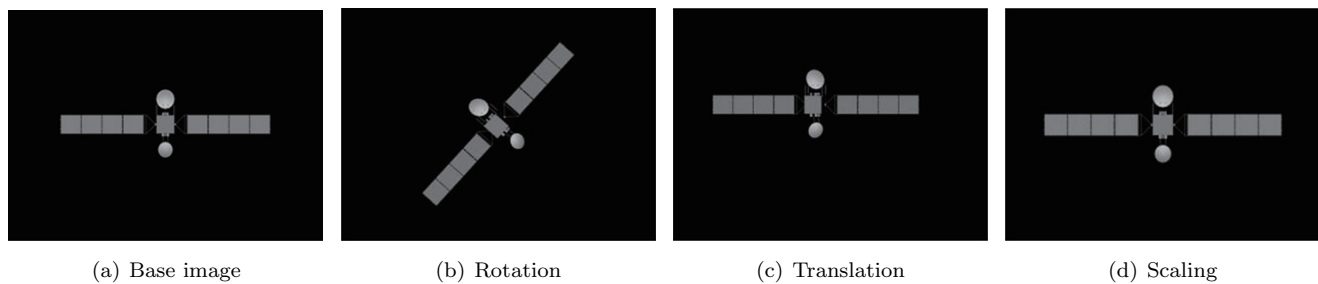


Fig. 6 Geometric transformation.

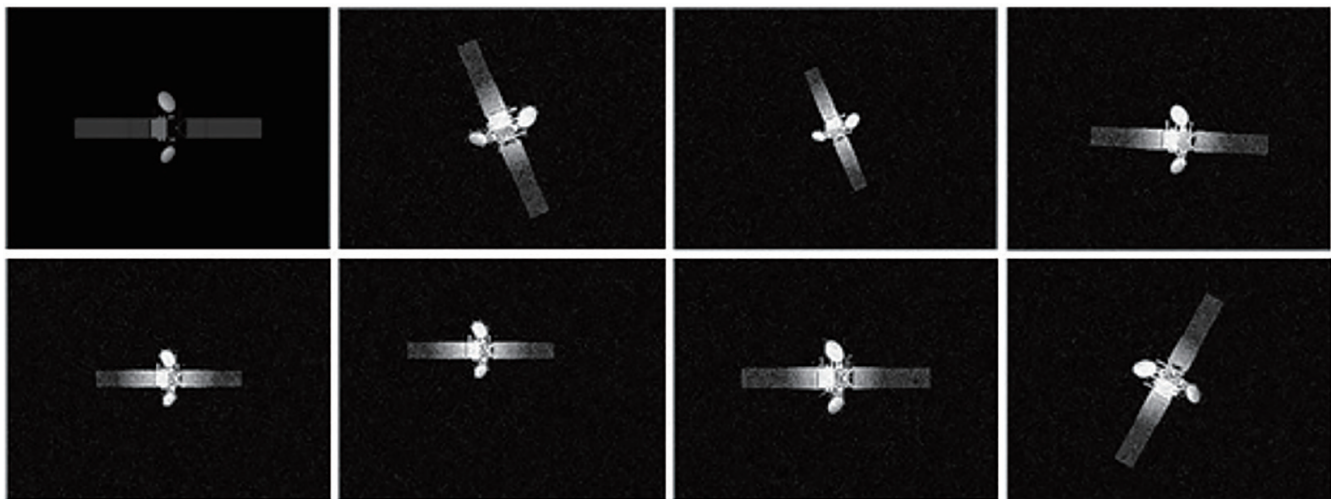


Fig. 7 Synthetic degradation.

section is concerned with the problem of building an effective model that recognizes the feature components of an on-orbit spacecraft. The degradation algorithm (especially the operations responsible for adding noise, blur, and illumination) increases the difficulty of semantic segmentation, inevitably in terms of object details and boundary regions. To address this problem, we developed the EASDN based on the DeepLab V3+ model, which we modified to take into consideration the degraded image boundary information.

3.1 DeepLab V3+

DeepLab V3+ [23], which forms the basis of our intelligent spacecraft feature component recognition network, offers the advantages of portability and high precision. It is the latest version of the DeepLab algorithm series [29–31]. To fuse multi-scale information, DeepLab V3+ introduces the encoder–decoder architecture commonly used in semantic segmentation, and balances precision and cost in terms of time via atrous convolution by controlling

the resolution of the encoded features. The DeepLab V3+ design is structured such that the entire previous model of DeepLab V3 [31] is employed as an encoder, followed by a simple but effective decoder. Atrous spatial pyramid pooling (ASPP) is the core of the DeepLab algorithm series. Multiple effective receptive fields can be realized via convolution and pooling operations with different dilation rates to obtain multiresolution features and discover multi-scale context information. DeepLab V3+ also utilizes depthwise separable convolution to reduce the computational overhead. DeepLabV3+ has advanced semantic segmentation capabilities and achieves segmentation accuracy on multiple benchmarks.

DeepLab V3+ extracts deep features with an Xception structure and performs feature fusion. Other common feature extraction networks, such as ResNet [32], DRN [33], and MobileNet [34], can also be applied as the backbone of DeepLab V3+. According to our experiments and the ablation study of Chen *et al.* [23], the model based on Xception is the best. Although the MobileNet model

is not as good as the model based on Xception, it has fewer parameters and is computationally less complex. Hence, MobileNet can be employed as a backbone that meets the requirements of a light model. Based on these factors, Xception and MobileNet were selected as the backbone for feature extraction.

3.2 Edge Auxiliary Supervision DeepLab Network

To simulate the on-orbit environment, the degradation algorithm is applied to the base images. Unfortunately, it causes the information about the target space to be lost in the image, especially in the boundary area of the spacecraft, thus reducing the precision of semantic segmentation. Inspired by previous work [35, 36], we developed the EASDN to which we added a supervision guide for the target, enabling the neural network to predict the edge of the target more precisely and improve the segmentation accuracy simultaneously. The structure of the model is illustrated in Fig. 8.

As shown in Fig. 8, the predict image P is obtained by the model. The predicted edge map P_E is the product of the Edge Extract Module with input P . The true edge map G_E is produced in the same manner as the ground truth G . Then, the true edge map G_E is designed to pass through a smooth module to obtain the smoothing true edge map G_s . Finally, the edge auxiliary loss is calculated between P_E and G_s .

3.3 Edge Extract and Smooth Module

The function of the Edge Extract Module is to extract

the edge of the object in an image by using an edge detection filter. The inputs are the predictive and real masks. Edge detection filters can generally be described as having a 3×3 convolution. The Sobel filter [37] and Laplace filter [28] are well-known edge detection filters. Based on our experiments, the Laplace filter is isotropic and delivers superior performance; therefore, we chose it as our edge detection filter. Its basic formula is as Eq. (5):

$$y(u, v) = x(u, v) + c(\nabla^2 x(u, v)) \tag{5}$$

where $x(u, v)$ is the input image, $y(u, v)$ is the output image, and c is a coefficient.

Smooth Module is prepared for the true edge map G_E , which facilitates training and alleviates the overfitting of our model. Here, we leverage the Gaussian filter. Alternatively, the median filter or mean filter would also be acceptable because their effects are almost the same in this specific problem according to our experiments.

3.4 Loss function

The total loss L_{Total} consists of the original pixel-wise cross entropy loss L_{CE} and the new Edge Auxiliary Loss L_{Edge} :

$$L_{Total} = L_{CE} + \alpha L_{Edge} \tag{6}$$

where α is a weight parameter that balances the weights of L_{Total} and L_{Edge} and L_{CE} is a common loss function for semantic segmentation [38]. The definition is

$$L_{CE} = -y \log(y') - (1 - y) \log(1 - y') \tag{7}$$

where y' is the predictive value and y is the true label value.

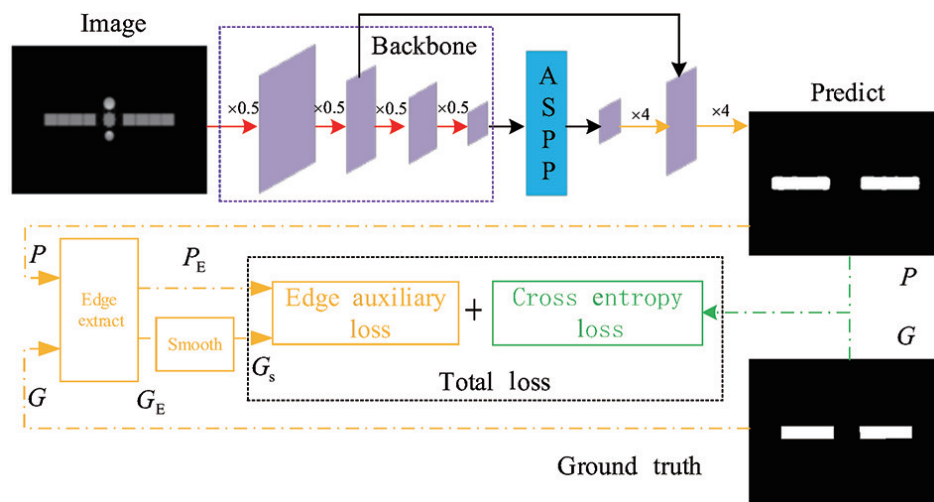


Fig. 8 Edge Auxiliary Supervision DeepLab Network.

In addition, the formula for L_{Edge} is as Eq. (8):

$$L_{\text{Edge}} = \frac{1}{2m} \sum_x |y - y'|^2 \quad (8)$$

where x is a sample and m is the sample number of a batch. L_{Edge} is the mean square error between P_E and G_s .

4 Experiments

4.1 Experimental setup

4.1.1 Training setting

The model was tested on a Linux Ubuntu 16.04 system based on the TensorFlow framework. Three NVIDIA GeForce GTX 1080Ti 11 GB GPUs, Intel Core i7 CPU, and 16 GB of memory were used for the training process. The training settings are listed in Table 1.

Table 1 Training settings

Parameter	Training setting
Learning rate scheduler	Poly
Training steps	15,000
Output stride	8
Input size (pixels)	320×240
Initial learning rate	0.0001
Training steps	15,000
Learning rate decay step	2,000
Learning rate decay factor	0.1

The learning rate scheduler is a method for adjusting the learning rate, which has a certain influence on the final training effect of the model. The output stride is a specific setting that measures the size of the feature map output by the encoder. It represents the ratio between the input size and size of the feature map. When the output stride is 16, which represents the size of the feature map from the encoder and is 1/16 of the input size, the corresponding atrous convolution rate is [12, 24, 36], and when the output stride is 8, the rate is [6, 12, 18], which enables the segmentation precision to be improved. However, in this setting, the calculation cost and training time increased. We also applied pre-trained models on ImageNet for training [39].

4.1.2 Dataset

We utilized the aforementioned Spacecraft Feature Component Dataset for training and testing. The training, validation, and test sets were divided randomly. There were 1603 images in the test set.

4.2 Evaluation metrics

To measure the function and contribution of the segmentation system, its performance needs to be evaluated. We would need to employ standard and acknowledged metrics to ensure fairness. Several aspects of the system need to be tested to assess its effectiveness, including the execution time, memory occupation, and accuracy [40].

Many criteria are available for measuring the accuracy of algorithms in image segmentation. These standards are usually variations of their pixel precision and intersection over union (IoU). The mean intersection over union (mIoU) is the most common metric because of its simplicity and representativeness, and has been employed by most researchers to report their results [18, 19, 21, 22]. Therefore, mIoU was considered as a criterion for accuracy in this study.

mIoU is the ratio of the intersection and union of two sets. For semantic segmentation, the two sets are the ground truth and the predicted values. This ratio can be viewed as the ratio between the intersection (i.e., true positive) and the sum of false negative, false positive, and true positive. It is computed by calculating the intersection over union for each class, and then taking the average [18]:

$$\text{mIoU} = \frac{1}{k+1} \sum_{i=0}^k \frac{p_{ii}}{\sum_{j=0}^k p_{ij} + \sum_{j=0}^k p_{ji} - p_{ii}} \quad (9)$$

where there are $k+1$ classes ($k=1$ in this study), p_{ij} denotes the number of pixels that are predicted to be in class j with label i . Therefore, p_{ii} is the true positive. The denominator of Eq. (9) represents the sum of the false negatives, false positives, and true positives for one class.

4.3 Quantitative evaluation

We only evaluate the models based on what we have trained because no similar datasets or models have been proposed before. Table 2 shows the segmentation accuracy of DeepLab V3+ with different backbone networks and our proposed EASDN model. In addition to displaying the segmentation accuracy of the model, Table 3 provides the training and testing speeds of the models, as well as the model size. Considering the different requirements of the model accuracy and speed on different platforms and hardware, our models satisfy the demands of engineering applications.

Table 2 mIoU of different models

Model (backbone)	mIoU (training set)	mIoU (validation set)	mIoU (test set)
DeepLab V3+ (MobileNet)	88.12%	88.08%	88.08%
DeepLab V3+ (Xception)	93.89%	93.61%	93.05%
EASDN (Xception)	94.46%	94.83%	94.62%

Table 3 Other performance measures of different models

Model (backbone)	Training speed (s/step)	Testing speed (f/s)	Model size (MB)
DeepLab V3+ (MobileNet)	0.045	60	9
DeepLab V3+ (Xception)	0.34	14	157
EASDN (Xception)	0.35	14	158

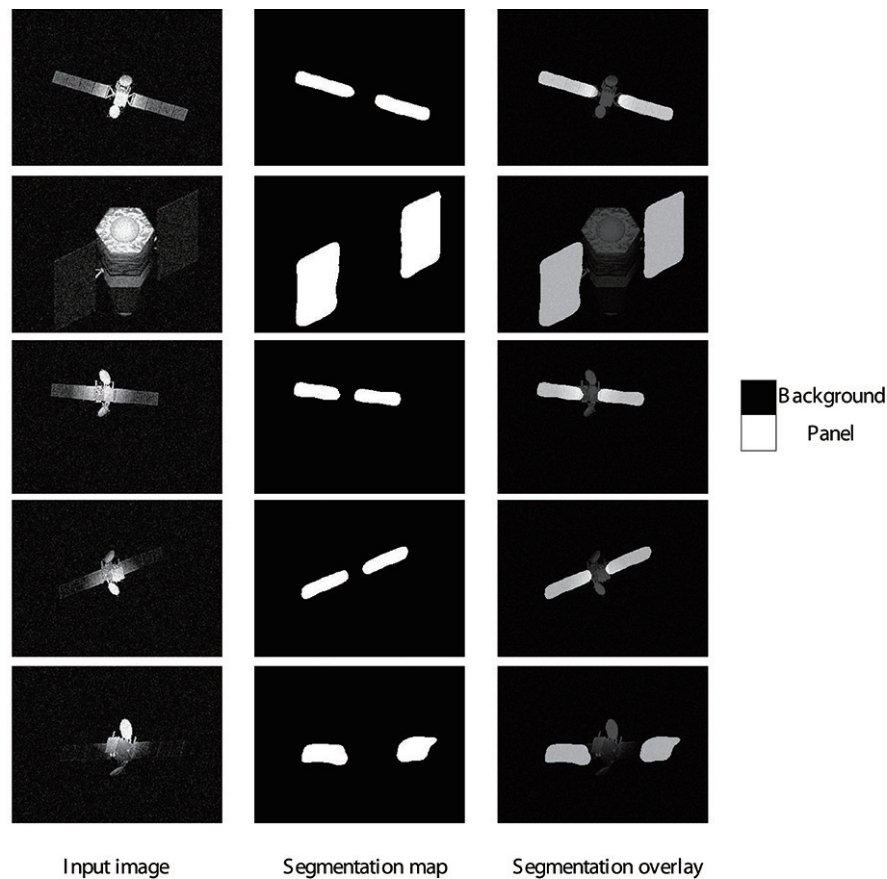


Fig. 9 Recognition results of EASDN.

The results in Table 2 indicate that the DeepLab V3+ model based on Xception has higher segmentation accuracy. In addition, EASDN further improves the segmentation accuracy, with an increase of 1.57% mIoU on the test set. At the same time, it can be seen from Table 3 that although the accuracy of the DeepLab V3+ model based on MobileNet is not competitive with that of the other networks, the training and testing speeds are significantly improved, and the model requires little memory. Moreover, our EASDN model did not reduce

the speed while improving the segmentation accuracy of the algorithm.

4.4 Qualitative results

The qualitative results of our models can be seen in Fig. 9. The figure depicts the segmentation results of the test images in the presence of noise, blur, and sunlight in a simulated on-orbit environment. The visualization results indicate that the recognition and segmentation results of the spacecraft panels by our proposed model can be

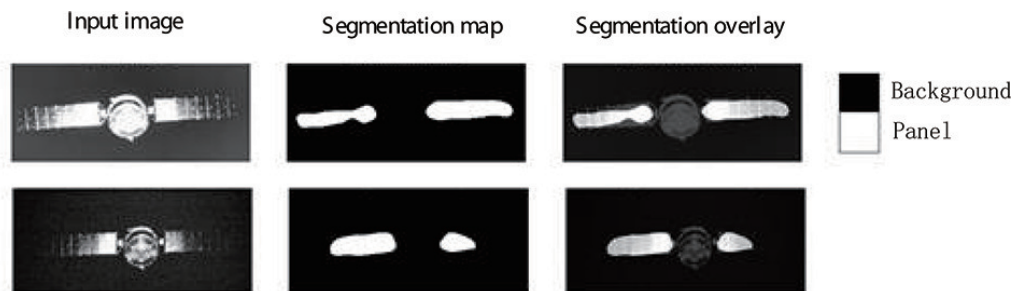


Fig. 10 Results obtained for real on-orbit spacecraft images.

considered to be acceptable.

Figure 10 shows the results of our test of real on-orbit spacecraft images from the Internet. Even under poor conditions, the spacecraft panels can be correctly identified and recognized, demonstrating the robustness of the model. In other words, it proves that our dataset is close to the real environment, and our degradation algorithm is also suitable.

5 Conclusions

A variety of internal degradation factors (noise, blur, and distortion from an on-orbit camera) and external disturbances (sunlight, space dust, and the plume from the space environment) make it challenging to recognize spacecraft feature components. Moreover, the lack of sufficient on-orbit spacecraft images has hindered research. We attempted to address this issue and constructed the SFCD with authentic degradations. The EASDN model was proposed to perform efficient and effective recognition of images in our dataset and real on-orbit images from the Internet. Our extensive experiments validated that the models trained on our SFCD benchmark have a beneficial generalization capability to real on-orbit conditions. In the future, we aim to enlarge the SFCD benchmark by considering a satellite against the Earth (or other planets) as the background and plan to add labels for additional feature components (such as aerials, cameras, and engine nozzles).

Acknowledgements

The authors acknowledge support from the National Natural Science Foundation of China (No. 11772023) and Science and Technology on Space Intelligent Control Laboratory (No. KGJZDSYS-2018-14).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

- [1] Francis, G., Collins, E., Chuy, O., Sharma, A. Sampling-based trajectory generation for autonomous spacecraft rendezvous and docking. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference, **2013**: AIAA 2013–4549.
- [2] Xu, Y. L., Zhou, H. J., Dai, H. Y. Design of GEO helix tourist orbit based on perturbation compensation. *AIP Conference Proceedings*, **2017**, 1839(1): 020084.
- [3] Skinner, M. A., Kelecyc, T. M., Gregory, S. A., Toth, J. P., Liang, D., Yamanaka, D., Kent, S., Tjoelker, R., Margineantu, D., Allison, A. L., *et al.* Commercial space situational awareness—An investigation of ground-based SSA concepts to support commercial geo satellite operators. In: Proceedings of Advanced Maui Optical and Space Surveillance Technologies Conference, **2013**.
- [4] Harris, K., McGruvey, M., Chang, H. Y., Ryle, M. Application for RSO automated proximity analysis and IMAGING (ARAPAJMA): Development of a nanosat-based space situational awareness mission. Technical report. Space Development and Test Directorate, Albuquerque, United States, **2013**.
- [5] Isbruckner, V., Stauder, J., Laurin, D., Hollinger, A. Stray light control for asteroid detection at low solar elongation for the NEOSat micro-satellite telescope. In: Proceeding of SPIE 8442, Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave, **2012**: 84424J.
- [6] Tang, K., Zou, B. Non-cooperative objects recognition based on fusing local feature with integer structure. In: Proceedings of the 1st Chinese Aerospace Safety Symposium, **2015**: 3–7. (in Chinese)
- [7] Zhi, X., Hou, Q., Zhang, W., Sun, X. Optical identification method of space typical targets based

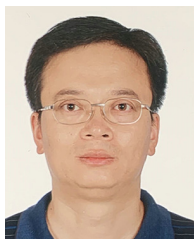
- on combined multi-feature metrics. *Journal of Harbin Institute of Technology*, **2016**, 48(10): 44–50. (in Chinese)
- [8] Hu, J. Research on optical identification method of typical satellite local components. Master Thesis. Harbin Institute of Technology, **2017**. (in Chinese)
- [9] Goodfellow, I., Bengio, Y., Courville, A. *Deep Learning*. Cambridge: The MIT Press, **2016**.
- [10] Krizhevsky, A., Sutskever, I., Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, **2012**, 6(6): 1097–1105.
- [11] Zhang, Y., Wu, Z., Wei, S., Zhang, Y. Spatial target recognition based on CNN and LSTM. In: Proceedings of the 12th Chinese Conference on Signal and Intelligent Information Processing and Applications, **2018**: 197–200. (in Chinese)
- [12] Hochreiter, S., Schmidhuber, J. Long short-term memory. *Neural Computation*, **1997**, 9(8): 1735–1780.
- [13] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. **2017**: arXiv: 1704.06857[cs.CV]. Available at <https://arxiv.org/abs/1704.06857>.
- [14] Li, L., Guo, B., Shao K. Geometrically robust image watermarking using scale-invariant feature transform and Zernike moments. *Chinese Optics Letters*, **2007**, 5(6): 332–335.
- [15] Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2005**, 1: 886–893.
- [16] Boykov, Y., Jolly, M. P. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: Proceedings of the 8th IEEE International Conference on Computer Vision, **2001**, 1: 105–112.
- [17] Liu, Z. Research of image segmentation algorithm based on graph theory. Ph.D. Dissertation. Lanzhou University of Technology, **2018**. (in Chinese)
- [18] Long, J., Shelhamer, E., Darrell, T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2015**: 3431–3440.
- [19] Badrinarayanan, V., Kendall, A., Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2017**, 39(12): 2481–2495.
- [20] Ronneberger, O., Fischer, P., Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. (Lecture Notes in Computer Science, volume 9351). Navab, N., Hornegger, J., Wells, W., Frangi, A. Eds. Springer Cham, **2015**.
- [21] Paszke, A., Chaurasia, A., Kim, S., Cukurciello, E. ENet: A deep neural network architecture for real-time semantic segmentation. **2016**: arXiv: 1606.02147[cs.CV]. Available at <https://arxiv.org/abs/1606.02147>.
- [22] Chaurasia, A., Cukurciello, E. Linknet: Exploiting encoder representations for efficient semantic segmentation. In: Proceedings of the 2017 IEEE Visual Communications and Image Processing, **2017**: 1–4.
- [23] Chen, L. C., Zhu, Y. K., Papandreou, G., Schroff, F., Adam, H. In: *Computer Vision—ECCV 2018*. Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. Eds. Springer, Cham, **2018**.
- [24] Zhang, H., Liu, Z., Jiang, Z., An, M., Zhao, D. BUAA-SID1.0 space object image dataset. *Space Recovery and Remote Sensing*, **2010**, 31(4): 65–71. (in Chinese)
- [25] Information on http://newspaper.dahe.cn/hnsb/html/2008-09/28/content_109711.htm (cited 6 Sep 2020).
- [26] Poynton, C. A. *A Technical Introduction to Digital Video*. John Wiley & Sons, **1996**.
- [27] Keys, R. Cubic convolution interpolation for digital image processing. In: Proceedings of the IEEE Transactions on Acoustics, Speech, and Signal Processing, **1981**, 29(6): 1153–1160.
- [28] Gonzalez, R. C., Woods, R. E., Eddins, S. L. *Digital Image Processing Using MATLAB*. Gatesmark Publishing, **2004**.
- [29] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. **2014**: arXiv: 1412.7062[cs.CV]. Available at <https://arxiv.org/abs/1412.7062>.
- [30] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L. DeepLab: Semantic image segmentation with deep convolutional nets and fully connected CRFs. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **2017**, 40(4): 834–848.
- [31] Chen, L. C., Papandreou, G., Schroff, F., Adam, H. Rethinking atrous convolution for semantic image segmentation. **2017**: arXiv: 1706.05587[cs.CV]. Available at <https://arxiv.org/abs/1706.05587>.
- [32] He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2016**: 770–778.
- [33] Yu, F., Koltun, V., Funkhouser, T. Dilated residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2017**: 472–480.

- [34] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2018**: 4510–4520.
- [35] Van Vliet, L. J., Young, I. T., Beckers, G. L. An edge detection model based on non-linear laplace filtering. *Machine Intelligence and Pattern Recognition*, **1988**, 7: 63–73.
- [36] Cui, L., Zhang, Y. Accurate semantic segmentation in turbulence media. *IEEE Access*, **2019**, 7: 166749–166761.
- [37] Gao, W., Zhang, X., Yang, L., Liu, H. An improved sobel edge detection. In: Proceedings of the 3rd International Conference on Computer Science and Information Technology, **2010**, 5: 67–71.
- [38] Jadon, S. A survey of loss functions for semantic segmentation. **2020**: arXiv: 2006.14822[*eess.IV*]. Available at <https://arxiv.org/abs/2006.14822>.
- [39] Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F. ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, **2009**: 248–255.
- [40] Luc, P., Couprie, C., Chintala, S., Verbeek, J. Semantic segmentation using adversarial networks. **2016**: arXiv: 1611.08408[*cs.CV*]. Available at <https://arxiv.org/abs/1611.08408>.



Linwei Qiu received his B.S. degree in detection, guidance, and control techniques from Beihang University in 2018. He is currently pursuing his M.S. degree in control science and engineering at the School of Astronautics, Beihang University. His research interests include AI applications and intelligent control.

E-mail: qiulinwei@buaa.edu.cn.



Liang Tang received his Ph.D. degree in spacecraft design from Beihang University in 2005. He is currently employed as a researcher at the Key Laboratory of Space Intelligent Control Technology. He is also a doctoral supervisor, model deputy chief designer, and senior director researcher in the complex spacecraft attitude control

field. He has directed and completed multiple major national projects and experiments in space engineering. E-mail: tl614@sina.com.



Rui Zhong received his Ph.D. degree in spacecraft design from Beihang University in 2011. He was a postdoctoral fellow in the Department of Earth and Space Science and Engineering at York University from 2011 to 2013. In 2013, he accepted a position as assistant professor at the School of Astronautics at Beihang University, Beijing, China, where he is currently an associate professor. His research interests include tethered satellite systems, spatial multi-flexible body dynamics, spacecraft dynamics and control, and aerospace applications of machine learning. E-mail: zhongruia@163.com.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.