Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 71 (2023) 3761–3774

ADVANCES IN SPACE RESEARCH
(a COSPAR publication)

www.elsevier.com/locate/asr

Intelligent recognition of spacecraft components from photorealistic images based on Unreal Engine 4

Yunpeng Zhao, Rui Zhong[*], Linyan Cui

School of Astronautics, Beihang University, 100083 Beijing, People's Republic of China

Received 27 April 2022; received in revised form 24 August 2022; accepted 19 September 2022
Available online 26 September 2022

Abstract

Autonomous and accurate recognition of satellite components is crucial for space tasks such as rendezvous, docking, maintenance, and refueling. Over the decades, the great advancement of deep learning has made us see the possibility of applying semantic segmentation to spacecraft intelligence recognition. However, the lack of training datasets required for deep learning is an insurmountable difficulty. Based on the needs of space missions and current challenges, this paper builds the space target dataset for satellite component recognition. Based on Unreal Engine 4, we establish a space simulation environment that can generate photorealistic images with earth backgrounds. After collecting and modifying 33 different high-quality satellite models, we import them into the environment and generate 10,000 satellite images with various attitudes and diverse backgrounds. Unlike existing datasets, our dataset, named UESD, has five distinctive components: solar panel, antenna, thruster, instrument, and optical payload. Furthermore, UESD constructs the earth background as much as possible to avoid the shortcomings of relevant simulation images. After building the dataset, we use a series of state-of-the-art semantic segmentation models to test their performances on our dataset. Using ConvNeXt-Base as the backbone, we propose a new decoder module named LUperNet. Our method achieves 84.6%*mIoU* and shows satisfactory accuracy compared with all baselines. More experiments are carried out to test the generalization ability. Results show that even on new satellite targets, a totally different dataset URSO, and real satellite images, our method can still recognize the components and maintain high accuracy. Experiments prove the effectiveness of applying our dataset and method to spacecraft component recognition. Our dataset, satellite models, and codes are available at https://github.com/zhaoyunpeng57/BUAA-UESD33.

Keywords: Semantic segmentation; Space target dataset; Spacecraft intelligence recognition

1. Introduction

In recent years, on-orbit service against major space targets dominated by satellites has become an important development direction of aerospace technology. The recognition technology of satellite components is the critical step. Recognizing components and determining their location is a prerequisite for subsequent rendezvous (Anthea et al., 2021), docking, servicing, debris removal, and other tasks. Here, components refer to the observable local payloads that play an essential role in the operation of the space target on-orbit and have noticeable graphic features and textures in the space target images. Typical components mainly include solar panels, antennas, and equipment carried by satellites. However, most space-based target detection and recognition researches focus on point-target detection at long distances. For satellite component detection, scholars use SURF (Bay et al., 2006) and traditional target detection methods (Cai et al., 2015), which require a

[*] Corresponding author.
E-mail addresses: zy2015128@buaa.edu.cn (Y. Zhao), zhongruia@163.com (R. Zhong), cuily@buaa.edu.cn (L. Cui).

https://doi.org/10.1016/j.asr.2022.09.045
0273-1177/© 2022 COSPAR. Published by Elsevier B.V. All rights reserved.

large number of parameters and are not satisfactory in the accuracy and efficiency.

In the last decade, artificial intelligence algorithms represented by deep learning have thrived. The emergence of deep convolution neural networks (Karen and Andrew, 2015; Kai et al., 2016; Gao et al., 2017) promotes the rapid development of many Computer Vision tasks, including object tracking (Hyeonseob and Bohyung, 2016; Bo et al., 2019; Qiang et al., 2019), visual correspondence (David et al., 2017; Shuai et al., 2019; Ju et al., 2020), and semantic segmentation (Hyeonseob et al., 2015; Liang et al., 2018). It is worthwhile and promising to apply deep learning technology to spacecraft component recognition. However, existing studies either use object detection (Chen et al., 2019) or instance segmentation (Chen et al., 2021), which only focuses on solar panels and antennas, so it is difficult to obtain satellite information comprehensively. Therefore, this paper first proposes using semantic segmentation to recognize five distinctive components, including instrument and thruster, that have not been studied before.

An unavoidable disadvantage of deep learning is the need for training datasets. Due to the difficulty in obtaining a large number of authentic space target images, the lack of public datasets has seriously hindered the progress of space intelligence recognition. Scholars build datasets by collecting authentic images and generating simulation images. The number of authentic images is small, but it can keep the details of satellites and their components to the greatest extent. On the contrary, the number of simulation images is large, but the disadvantage is that it is easily affected by the accuracy of satellite models. At the same time, the simulation images are difficult to contain the earth's background. Therefore, how to build a large number of satellite images with high-quality modeling of the satellite models and the earth's background has become the focus of research.

Motivated by this, this paper establishes a space simulation environment based on Unreal Engine 4 (UE4). The environment builds the earth, sun, and star background, which can simulate the satellites running in earth orbit and generate photorealistic images. In order to obtain more abundant satellite images, we construct a 3D model dataset containing 33 high-quality satellites models. Importing the model dataset into the environment can obtain satellite images with multiple attitudes and backgrounds.

Fig. 1 shows the comparison of the existing STK dataset, a real image from NASA, and two images of the simulation environment based on UE4 established in this paper. Our method approximates the real satellite images as much as possible and avoids the shortcomings of the existing space target datasets that do not have earth backgrounds. Furthermore, unlike some studies that focus on identifying the whole satellite or a few kinds of components, the space target dataset established in this paper includes five distinctive components: solar panel, antenna, instrument, thruster, and optical payload. We carry out experiments to obtain the baseline results of different semantic segmentation models on our dataset. Based on the best backbone, we propose a new method named LUperNet. Experiments are carried out to test the effectiveness and generalization of our method. According to the results, we analyze the advantages and disadvantages of the existing methods and look forward to future research.

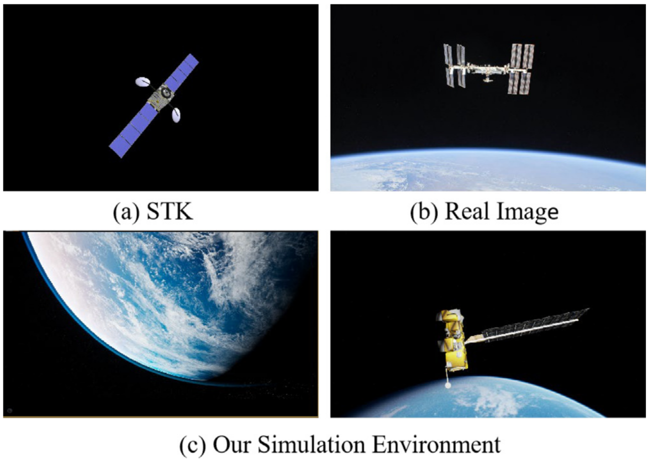
(a) STK
(b) Real Image
(c) Our Simulation Environment

Fig. 1. The comparison of STK dataset, real image, and our simulation environment based on UE4.

The main contributions of this paper can be summarized as follows.

1) The dataset proposed in this paper is the first public UE4-based dataset used for spacecraft intelligence recognition with five distinctive components: solar panel, antenna, instrument, thruster, and optical payload. In order to obtain photorealistic images, we establish a space simulation environment that constructs the near-earth space to a high degree based on UE4. The environment models the earth, sun, stars, land, ocean, atmosphere, aurora, and other elements on earth. Besides the environment, we establish a model dataset containing 33 different kinds of satellites by collecting NASA's satellite geometric models and modifying the textures. After importing the satellite geometric models into the space simulation environment, we build a space target dataset named Unreal Engine Satellites Dataset (UESD) with different attitudes and backgrounds.

2) Based on ConvNeXt-B, we carefully design a new decoder module to use each feature in the encoding process fully. We design two experiments: an entire dataset (five components) for cooperative targets with known equipment distribution to test our dataset and method; a partial dataset (two components) for non-cooperative targets to test the generalization ability of our method.

Cooperative targets: To benchmark our dataset, we first use various state-of-the-art semantic segmentation models to test their performances and compare them with our method. We also verify the effectiveness of our method on ADE20K dataset. Our method recognizes five distinctive components with high accuracy and performs better than baselines. The above experiments prove that our dataset and method can be used for spacecraft intelligence recognition. For cooperative targets, autonomous and accurate recognizing components help the space missions, such as rendezvous and docking.

Non-cooperative targets: It is difficult to obtain the equipment distribution diagrams of non-cooperative targets in actual application scenarios, so annotating all components in advance is hard. Therefore, we conduct a partial dataset containing two significant components: solar panel and antenna. These two components usually can be seen at a longer distance than other components and have similar shapes. We adopt a new training method by dividing the dataset into two parts, 23 satellites to train and ten satellites to test. In addition, we test the generalization ability of the network on a public dataset URSO and real satellite images. Testing on different datasets simulates the recognition of new targets in real situations. Experiments prove the satisfactory generalization ability of our method. Even on new satellites, a completely different dataset, and real satellite images, our method can still recognize the components, and the accuracy maintains a high level.

The remainder of this paper is organized as follows. In Section 2, we introduce the related work of the space target dataset and semantic segmentation. Section 3 explains the constructing method and the composition of our dataset. Our proposed method is illustrated in Section 4. Section 5 is the experimental part. We first use a series of advanced semantic segmentation models to test our dataset and compare them with our method. Then, we carry out experiments to test new satellites, the URSO dataset, and real images. In addition, after analyzing the pros and cons of the existing method, we introduce our future work. Finally, our conclusion is summarized in Section 6.

2. Related work

2.1. Space target dataset

The lack of relevant space target datasets is a severe challenge to applying object detection and semantic segmentation algorithms to satellites. In the past few years, scholars have tried different methods to build datasets, which can be divided into collecting authentic satellite images and generating simulation images. Shuai (2019) used the YOLO algorithm to identify non-cooperative satellites based on 300 authentic satellite images. Hoang et al. (2021) created a space target dataset containing 3117 authentic satellite photos, which can be used for object detection and instance segmentation. The dataset included solar panels, antennas, and satellite bodies. The simulation images usually came from different software. Zhang et al. (2010) and Hao and Yong (2017) constructed datasets based on STK. Li and Zhang (2020) used 3ds max software to establish satellite models and build the training dataset. Mate et al. (2020) created a publicly available spacecraft dataset named SPEED, containing most synthetic images and a few authentic photos. The SPEED was initially used for attitude determination, but Xu et al. (2020) annotated the local features of the dataset to identify antennas and sensors.

The deficiency of the above space target datasets is that the number of authentic images is small, and the types of satellites are few, making it challenging to meet the needs of large-scale deep learning datasets. The number of simulation images is large, but they usually do not have the rich texture details of authentic images. In addition, existing simulation datasets often only focus on the whole satellite or a few components, such as solar panels and antennas. What is worse, because the above simulation images only include satellites and do not include the earth background, the model trained on the above datasets is challenging to transfer to the actual space environment due to the change in lighting environment and the influence of alternating interference between stars and earth background.

The dataset URSO (Pedro and Yang, 2019) used for spacecraft attitude estimation is the most similar to actual space target recognition requirements. The state-of-the-art game engine (e.g., UE4) has been widely used in autonomous driving (Alexey et al., 2017; Shital et al., 2017) and robotics (Pablo et al., 2018). Some works (Roland et al., 2018) have demonstrated the feasibility of using image simulation tools to build datasets and have been applied in pose estimation (Sumant and Simone, 2019) and asteroid landing (Steve et al., 2013). Based on UE4, URSO makes relatively rich and accurate modeling of the earth's background and can generate photorealistic images. Nevertheless, it only includes two satellites: Dragon spacecraft and Soyuz.

Therefore, this paper will adopt the construction method of the URSO dataset. We establish a near-earth space simulation environment with relatively realistic earth, sun, and star background. The establishment of high-quality earth background avoids the problem of a simple background of existing simulation images. We also build a large number of satellite models to make up for the disadvantage of only two satellites in URSO. In order to obtain the satellite information as comprehensively as possible, in addition to the antenna and solar panel, our dataset also includes the components that other datasets do not have, such as propulsion system and optical payload.

2.2. Semantic segmentation

The goal of semantic segmentation is to assign each pixel of an image into predefined object classes. Since the emergence of fully convolutional networks (Jonathan et al., 2014), semantic segmentation has made significant progress in autonomous driving, medical image segmentation, and other fields. Models such as Deeplab (Liang et al., 2014), Unet (Olaf et al., 2015), Danet (Jun et al., 2019), Sfnet (Xiang et al., 2020), and OCRnet (Yu et al., 2020) have appeared successively. The details of these models are different, but they all follow the basic framework of encoder-decoder. In addition to the advanced architecture, excellent ideas such as pyramid pooling module (PPM) (Heng et al., 2017), atrous spatial pyramid pooling (ASPP) (Liang et al., 2017), depth-wise convolution (Wen et al., 2016), and attention mechanism (Jie et al., 2018) also improve the performance of CNN.

In the years when CNN demonstrates its ability in the Computer Vision (CV) fields, Transformer (Ashish et al., 2017), an encoder-decoder architecture with self-attention modules, also shines brightly in natural language processing (NLP). When it comes to 2021, Transformer meets CV and arouses magical echoes. Vision Transformer (ViT) (Alexey et al., 2021) is the first work to leverage Transformer to image classification. Since then, pioneering architectures like SETR (Si et al., 2021), Swin (Ze et al., 2021), and Twins (Xiang et al., 2021) have demonstrated comparable or better performance than CNN in semantic segmentation. The self-attention mechanism in Transformer can capture the long-range dependence and adaptability, but the shortcoming is that it converts images into 1D sequences, which neglects the 2D structure of images. The huge computational consumption is also a severe problem for high-resolution images. An important question is: *Why does Transformer perform better than CNN?* To answer this question, ConvNeXt (Zhuang et al., 2022) starts with a standard ResNet-50 and gradually modernizes the architecture to Transformer. ConvNeXt demonstrates the ability to outperform Transformer and points the way for CNN. After that, VAN (Meng et al., 2202) proves that the combination of depth-wise convolution and point-wise convolution can utilize local information and capture long-distance dependencies, which means that convolution can have the same effect as Transformer. Novel studies further explain the connection between local attention and dynamic depth-wise convolution (Qi et al., 2022). Xiao et al. (2022) even reveal that large convolutional kernels may be better than small kernels because large kernels have much larger effective receptive fields.

Scholars have made such outstanding achievements in the backbone or encoder module, but the decoder may still have room for improvement. In addition, most of the above methods are tested on public datasets, and the recognition effect for our dataset may decline slightly. Therefore, we need to obtain the benchmark and modify the networks according to the characteristic of our dataset. Based on the conclusion that depth-wise Conv and large convolution kernel have outstanding performance, this paper uses ConvNeXt-B as the backbone for its excellent performance on public datasets and modifies the decoder to utilize the features of each stage of the encoding process. Our method will be explained in detail in Section 4.

3. Dataset

Models: The first step in building the space target dataset is to obtain the satellite models. STK has many space models, but it is not easy to export to the simulation environment due to their particular format. Because the satellite's detailed structure and sizes are hard to obtain, it is also difficult to build a lot of high-precision models using CAD software like 3dsmax. Therefore, we choose the third way, using 3D models provided by NASA's website. Comparing NASA satellite models with STK and real satellite images, NASA models have higher authenticity and better details. We select 33 different satellite models from NASA. However, the solar panels of some original models are only decorated with blue images without any texture. Therefore, we refer to the satellite images to modify these textures. Fig. 2 shows four satellite models in Blender software. We could find that the textures of the modified solar panels are relatively complicated. In addition, we change the gloss, metallicity, and other properties to make them look more realistic. We also collect equipment distribution diagrams of each satellite to identify the components. The NASA satellite models and diagrams used in this paper are also shown on Github.

Simulation Environment: This paper establishes the near-earth space simulation environment based on Unreal Engine 4. Our environment includes the construction of the earth, sun, star background, cloud, ocean, and land on the earth's surface. We model the earth as a sphere and simulate the flow of clouds or day and night changes by applying different texture maps and setting their functions. Functions here include adjustments to the proportion of day and night, land light intensity, cloud density, height, velocity, and other parameters. Adjusting the parameters can get different earth backgrounds. We use directional light to simulate the sunlight. The establishment of the star is also using star textures. Our environment is shown in Fig. 1. The texture maps of the earth's day map, night map, stars, and clouds can be found here.

Satellite images: After importing the models into the space simulation environment, we obtain images by rotating satellites, adjusting their positions, and changing the function parameters of the earth. The final dataset, named UESD, consists of 10,000 images with various types of satellites, different attitudes, and diverse backgrounds. The satellite names and the corresponding number of images are shown in Fig. 3. The number of images of each satellite is not fixed, generally following the principle that the more types of components, the greater the number of images.

Referring to the satellite equipment distribution diagrams, we choose five distinctive components: Solar panel, Antenna, Instrument, Thruster, and Optical payload. Solar panel and Antenna are easy to understand. Instrument means the equipment or the payload on the satellite, such as magnetometer boom (Cassini), gammy ray and neutron detector (Dawn), and star trackers (Cloudsat). Since the loads carried by each satellite are not the same, the shapes and sizes are also different. The instrument here shall include the visible satellite equipment as much as possible. Thruster includes propulsion system and engine. Optical

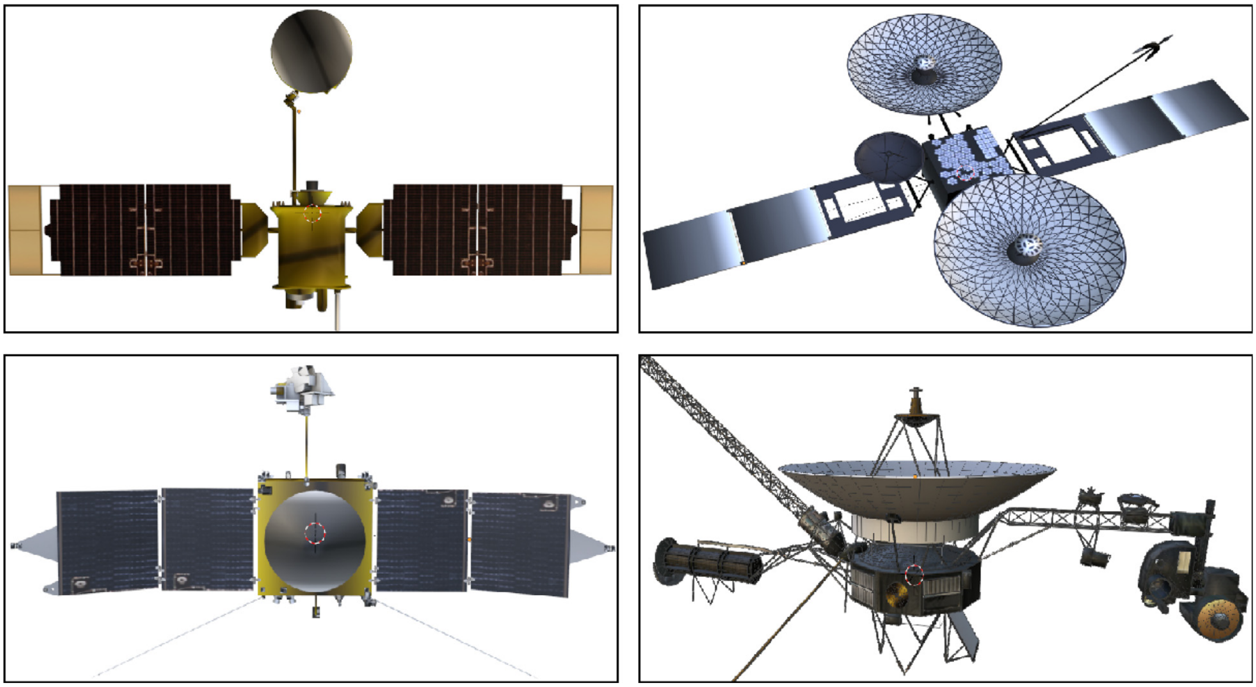

Fig. 2. Satellite models (Mars-global-surveyor, TDRS-h, Maven, Voyager-1, respectively).

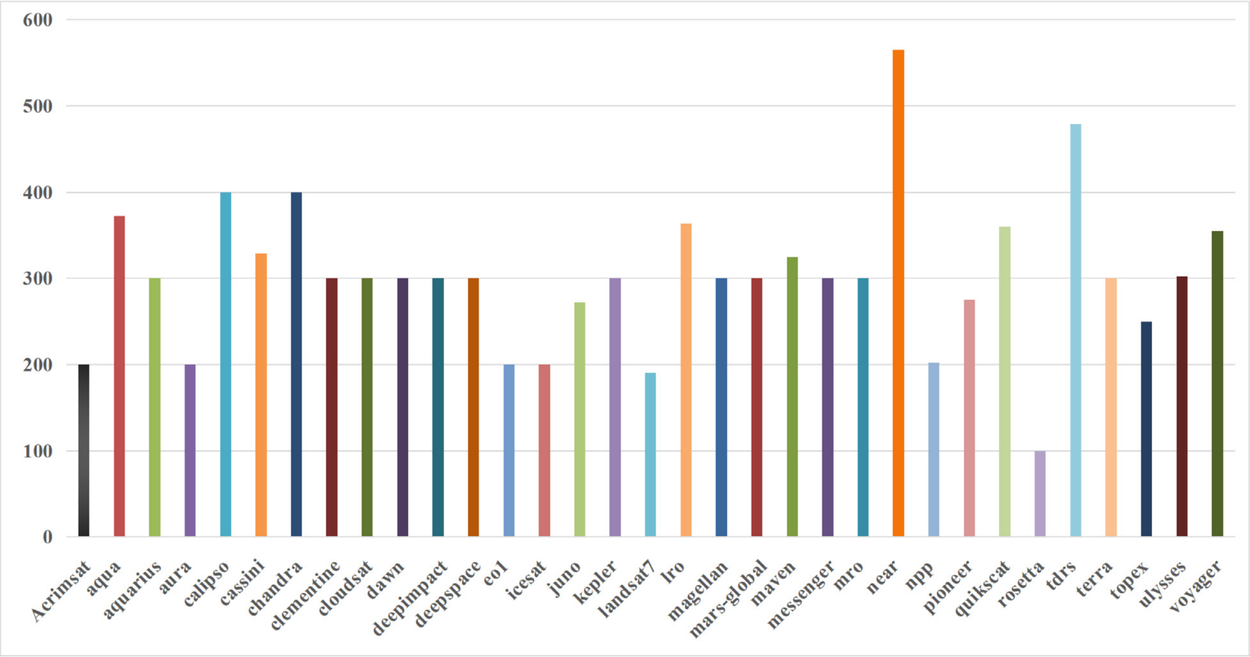

Fig. 3. Satellite names and their corresponding number of images.

payload is a special kind of instrument, such as high-resolution camera (Chandra) and camera system (Voyager).

To our knowledge, UESD is the first publicly available UE4-based dataset for space target recognition with so many components. We will continue to update it in the future. For example, our dataset can be improved by importing more satellite models and building a more refined simulation environment.

4. Our method

Through our experiments (See Section 5), ConvNeXt-B greatly surpasses other models in recognition accuracy, so we choose it as the backbone and propose a new decoder module. The overall architecture of our proposed method is shown in the left half of Fig. 4. The right half of the figure is one Block (Zhuang et al., 2022) in Stage4 and the structure of the Pyramid Pooling Module (PPM) (Heng et al., 2017). Each block in ConvNeXt-B has one depth-wise convolution with a kernel size of 7 × 7, Layer Normalization (LN) (Jimmy et al., 2016), GELU (Dan and Kevin, 2020), and two point-wise convolutions. Depth-wise convolution uses the local contextual information (Wen et al., 2016). Point-wise convolution (1 × 1 Conv) captures the relationship in the channel dimension. PPM fuses the features of four different pyramid scales and can carry both local and global context information as much as possible.

Widely adopted in many methods, UperNet (Te et al., 2018) uses different level features from the encoder process and fuses them with the PPM output. Unlike DeeplabV3+ (Liang et al., 2018) only uses the low-level feature from stage 1, UperNet better uses the feature maps generated by stages 1–3. On the basis of UperNet, we design a new architecture named Loop UperNet (LUperNet).

Given an RGB image$X \in R^{H\times W\times 3}$, our method maps it to a semantic map $Y \in R^{H\times W\times C}$, where H and W donate the height and width of the input image, C is the number of predefined semantic categories. After Stage1-4, the output size becomes half of the input size and the sizes of feature map are changed to $\{H2, W2, 128\}$,$\{H4, W4, 256\}$, $\{H8, W8, 512\}$, $\{H16, W16, 1024\}$, respectively. All channels keep the same during decoder, that is 256. We denote the maps of each stage as $\{F_1, F_2, F_3, F_4\}$ and the maps of decoder as $\{P_1, P_2, P_3, P_4\}$, where P_4 is the feature map coming from PPM. We resize P_4 via bilinear interpolation to the size ofF_3. A convolutional layer is used to adjust F_3 channel dimension and then concatenate these two layers.

$$P_3 = \mathrm{cat}(\mathrm{conv}(F_3), \mathrm{Re}(P_4)) \tag{1}$$

where cat represents the concatenation operation, conv means the 3 × 3 convolutional layer, Re means the bilinear interpolation operation.

In this way, P_3 is combined with F_2 to generateP_2, and thenP_1. However, in order to preserve features as much as possible, $\{P_4, P_3, P_2\}$ are used again to generate new feature maps $\{p_4, p_3, p_2\}$.

$$p_4 = \mathrm{cat}(P_4, \mathrm{conv}(P_4)) \tag{2}$$

Finally, a convolutional layer is applied to fuse these four feature maps $\{P_1, p_2, p_3, p_4\}$.

Our method uses the feature maps twice. For example, the PPM output P_4 is not only used for up-sampling to concatenateP_3, but also directly generates the final feature mapp_4. LUperNet carries out feature fusion to a greater extent because the loops (see the purple, red, and green lines in Fig. 4) emphasize the feature maps again. In addition, the process of up-sampling can sometimes cause the loss of spatial information. Therefore, we fuse the feature before and after conv to compensate for this loss which may hinder the accuracy of small objects. This simple but effective change improves the performance of the network without increasing the computation consumption too much.

5. Experiment

In this section, we first use a series of SOTA semantic segmentation models to benchmark our dataset and compare them with our method. More experiments are carried out in the Appendix (see Table 3) to prove the effectiveness of our method. Furthermore, we conduct experiments to

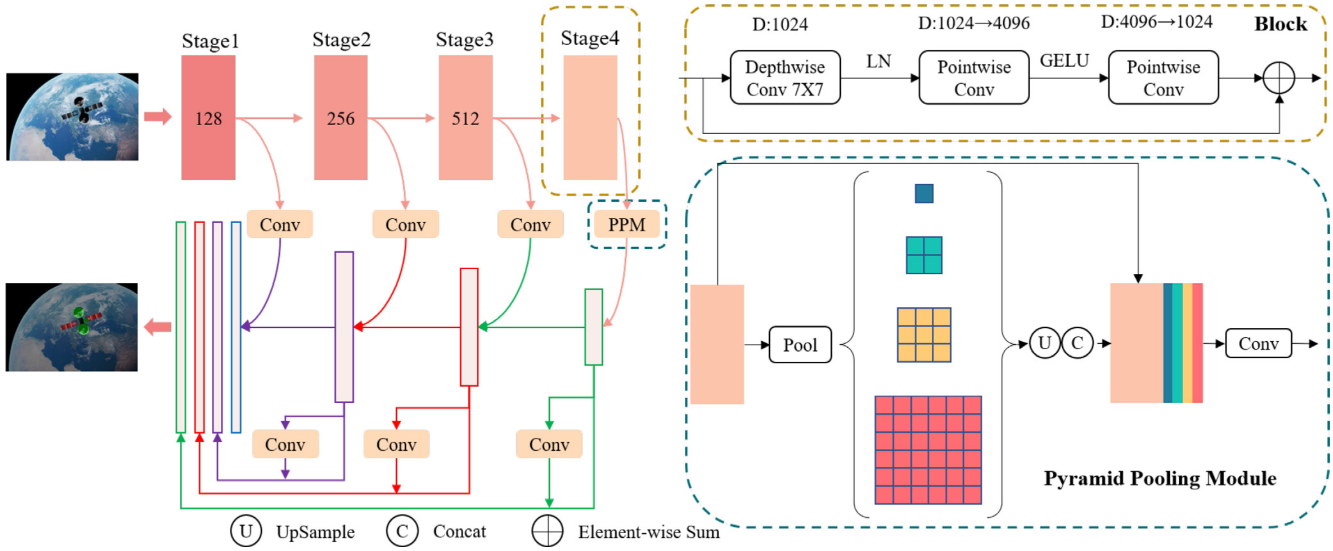

Fig. 4. Overall architecture of our proposed method.

test the generalization of neural networks by adopting a new training method and testing on the URSO dataset. Finally, we introduce the future research direction.

5.1. Benchmark

Settings We conduct experiments on UESD, which consists of 7000, 1500, and 1500 separately for training, validation, and testing. We use MMSEG (MMSegmentation Contributors, 2020) framework to carry out the following experiments. To benchmark our dataset, we choose Danet, Deeplabv3+, OCRnet, and SFnet with the backbone of ResNet-50. The training settings are SGD optimizer with momentum of 0.9, poly learning rate policy, the initial learning rate of 0.001, and weight decay of 0.0005. However, for VAN, ConvNeXt-B, and our method, we use

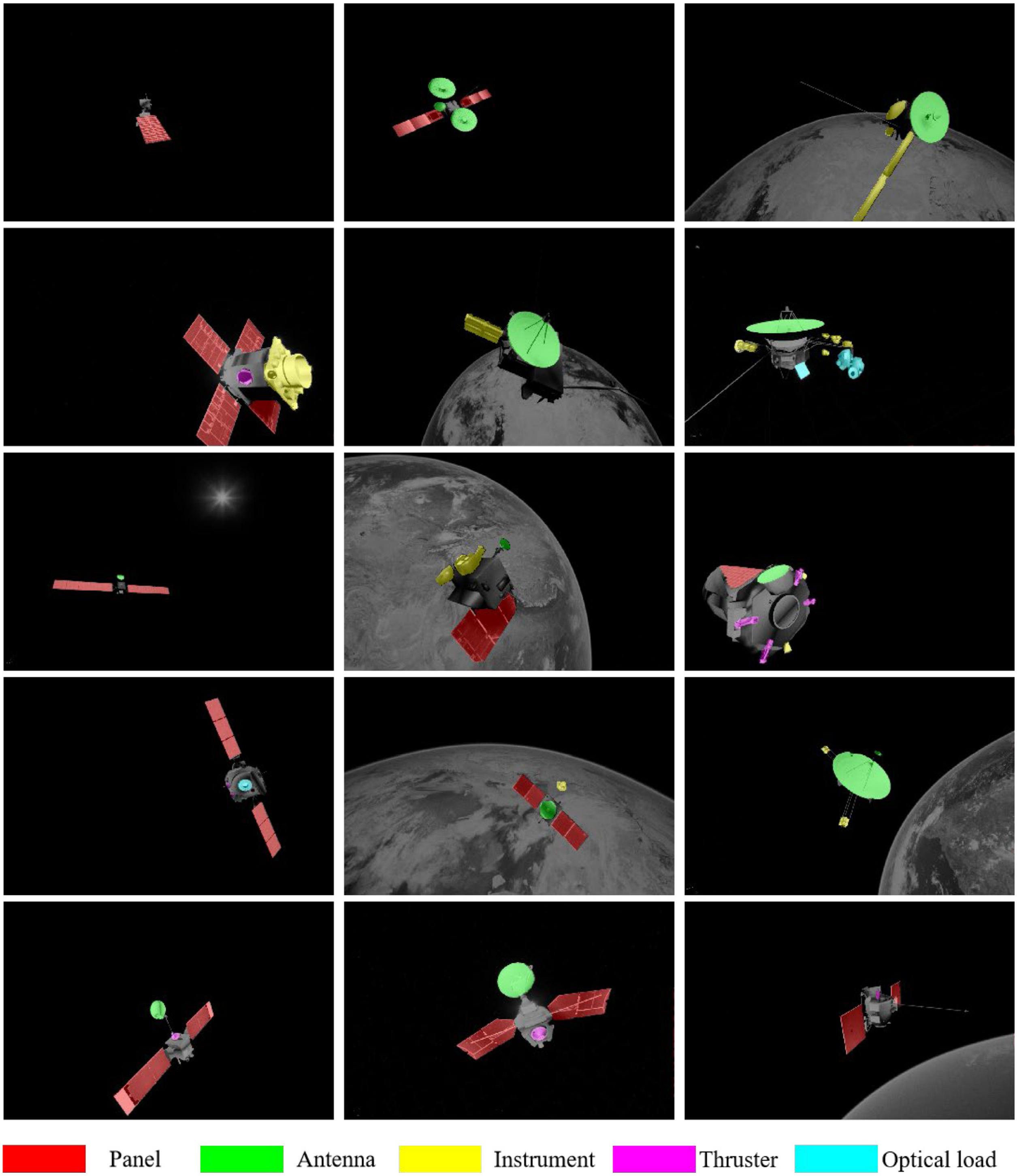

Fig. 5. Recognition results of five components.

Table 1
Compare with the state-of-the-art methods on UESD validation set.

Method	Solar panel	Antenna	Instrument	Thruster	Optical payload	*mIoU*%
Danet	85.69	77.65	54.24	46.16	64.64	65.68
Deeplabv3+	88.67	84.98	66.53	62.25	76.99	75.88
OCRnet	89.03	83.33	63.3	56.21	73.9	73.15
SFnet	89.51	86.6	68.77	67.66	78.42	78.19
VAN-B	90.97	84.99	71.24	66.65	78.36	78.44
ConvNeXt-B	93.85	89.66	72.98	72.92	83.87	82.66
Ours	**94.18**	**89.85**	**77.06**	**76.19**	**85.71**	**84.60**

the AdamW optimizer (Diederik and Jimmy, 2015) with a momentum of 0.9, cosine learning rate schedule (Llya and Frank, 2017), a learning rate of 0.0001, layer-wise-learning rate decay (Kevin et al., 2020; Hang et al., 2021), and weight decay of 0.05. The setting of training parameters is based on the original references. For data augmentation, we apply random cropping with a crop size of 512 × 512, random resizing with a scale range of [0.75, 2], and random flipping. All models are trained for 80 K iterations with a batch size of 4.

Results Fig. 5 visualizes the prediction results of our method on the UESD validation set, in which red represents the solar panel, green represents the antenna, yellow represents the instrument, pink represents the thruster, and cyan represents the optical load. All images are converted to grayscale images for better viewing. Due to the space limitation, Fig. 5 shows the results of 15 satellites, and the remaining 18 satellites are shown in Appendix A.2. For quantitative evaluation, mean of class-wise intersection-over-union (*mIoU*) is used for accurate comparison. Table 1 shows the comparison of our method with different CNN-based models on the UESD validation set.

Our method achieves 84.6 %*mIoU* and shows excellent performance compared with different baselines. For example, we surpass VAN-B by 6.16 %*mIoU* and ConvNeXt-B by 1.94 %*mIoU*. The shape of the solar panel is relatively

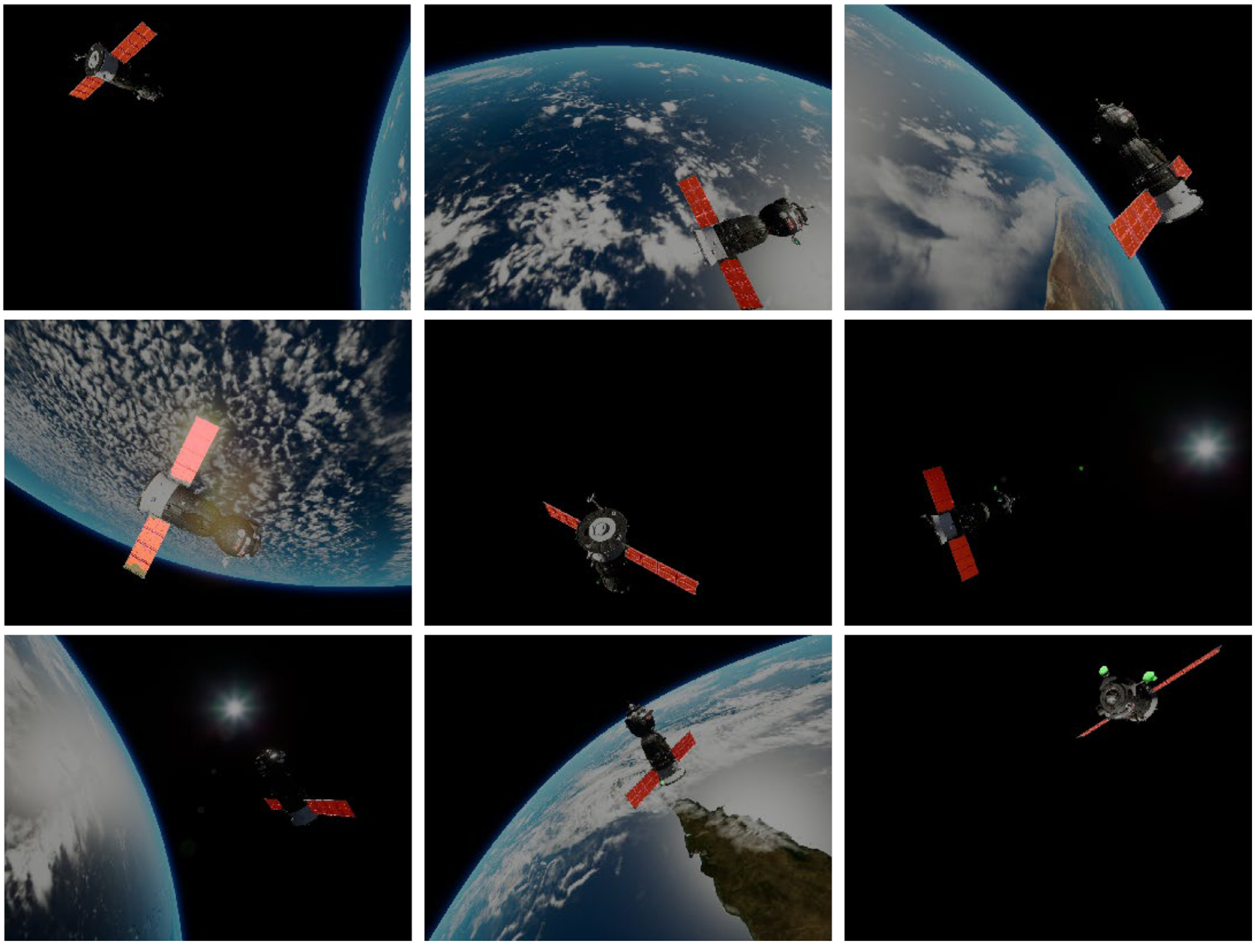

Fig. 6. Recognition results of URSO dataset (Soyuz).

similar, so the recognition result is the highest, reaching 94.18 %. Due to the variety of forms of instrument and thruster, the recognition accuracy is not as high as that of antennas, solar panels, and optical payloads. However, because LUperNet carries out feature fusion as much as possible and compensates for the loss of spatial information caused by the up-sampling process, our method has noticeable improvement for small components. Compared with ConvNeXt-B, the *mIoU* of Instrument and Thruster are increased by 4.08 % and 3.27 %, respectively.

The above experiments prove the effectiveness of our dataset and the excellent performance of our method. For cooperative targets with a known distribution, CNN can accurately recognize the type and determine the location of different components. After that, we can carry out subsequent rendezvous, docking, maintenance, and other tasks.

5.2. Experiments on different datasets

In the above experiments, we train the entire dataset, which means we know all spacecraft structures in advance. However, for non-cooperative satellites, it is difficult to obtain their equipment distribution. In this case, the non-cooperative target may only observe the solar panel and antenna because these two components are usually more evident than the other three components.

Therefore, we select these two components and construct a partial dataset. In addition, we adopt a new training method. We use 23 satellites for training and validation. However, ten new satellites are tested to verify whether the neural network can recognize new targets. The testing set contains Juno, Ulysses, Voyager-1, Chandra, Maven, LRO, Kepler, Aquarius, Pioneer, and Messenger ten satellites. Train-Val and testing sets are different to simulate the recognition of new satellite targets in real situations. Furthermore, to verify the generalization performance, we test whether the network trained on UESD can still recognize components on an entirely different dataset URSO, which has two satellites: Soyuz and Dragon spacecraft. The Dragon spacecraft has no obvious components, so we choose 4000 images of the Soyuz spacecraft.

Unlike the benchmark dataset in Section 5.1, we use SFnet, ConvNeXt-B, and our method to carry out experiments. All models are trained for 40 K iterations, and other settings are the same as those in *Sec* 5.1.

The validation and testing sets have two components, while URSO has only solar panels. The prediction results of the validation and testing set are similar to Fig. 5. Here, we show the recognition results of the Soyuz spacecraft in the URSO dataset, see Fig. 6. It should be noted that for URSO, some parts identified as antennas are not calculated into*mIoU*. Because there are few antennas in the URSO dataset, we only label the solar panel, but we can still see green antennas in some results, such as the last image in Fig. 6.

Quantitative results are shown in Table 2. For comparison, we list only *mIoU* of different datasets. The accuracy of each component can be found in the Appendix (see Table 4).

Table 2
Quantitative results on validation set, testing set, and URSO dataset.

Method	*mIoU*%-val	*mIoU*%-test	*mIoU*%-URSO
SFnet	84.25	81.51	78.2
ConvNeXt-B	91.47	82.94	81.06
Our method	93.02	83.32	81.95

We could find that for 10 test satellites, the performance of the network has decreased (e.g., 84.25 % vs 81.51 %, 91.47 % vs 82.94 %). Compared with ConvNeXt-B, our method achieves 93.02 %*mIoU* by an improvement of 1.55 % and 83.32 %*mIoU* by an improvement of 0.38 % for the validation set and testing set, respectively. For URSO, the decline of recognition effect is more pronounced but still achieves 81.95 %*mIoU* for our methods.

To further explore the universality of our dataset and method, we collect 333 real images of the International Space Station, Soyuz, Progress, Dragon, and other satellites from NASA's website. We carry out experiments to test whether our method can recognize the components of real satellite images. Our method achieves 77.6 %*mIoU* and the results are shown in Fig. 7.

Experimental results show that compared with the known datasets, the recognition effect of the neural network on new targets is reduced. However, it can still maintain a high level for simple components, like solar panels and antennas. Furthermore, even for a completely different dataset URSO and real satellite images, the neural network trained on our dataset can still recognize the components, which proves the universality of UESD and the excellent generalization of our method.

The above experiments test our dataset and method from a series of aspects, but there are still some shortcom-

Table 3
Quantitative results on ADE20K.

Method	Backbone	*mIoU*
SFnet	ResNet-101	44.67
OCRnet	ResNet-101	45.28
UperNet	VAN-B	48.3
UperNet	ConvNeXt-B	53.1
LUperNet	**ConvNeXt-B**	**53.42**

Table 4
Quantitative results of each component on validation set, test set, and URSO dataset.

	SFnet	ConvNeXt-B	Our method
Panel-val	85.28	92.26	**94.03**
Antenna-val	83.21	90.68	**92.01**
mIoU-val	84.25	91.47	**93.02**
Panel-test	73.85	**76.04**	75.95
Antenna-test	89.18	89.84	**90.68**
mIoU-test	81.52	82.94	**83.32**
panel/*mIoU*-URSO	78.2	81.06	**81.95**

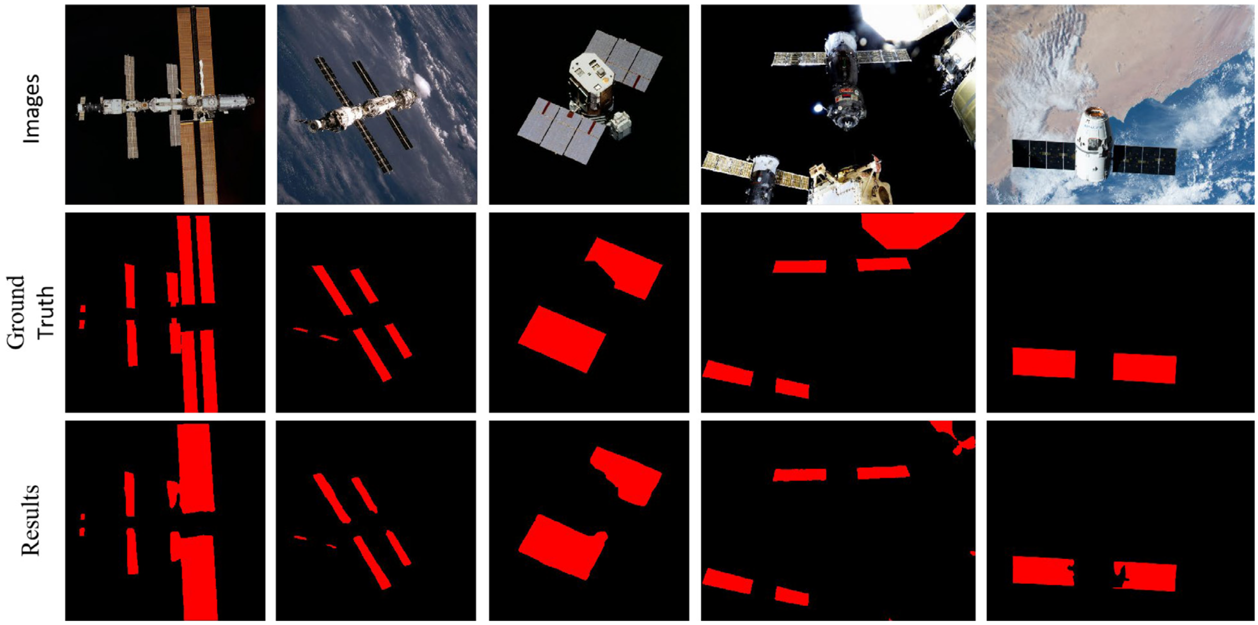

Fig. 7. Comparison of Images, Ground Truth, and Recognition results of NASA real satellite images.

ings. For satellites with complex structures, such as ISS, the recognition effect of some small panels is poor and the boundary of the panels is relatively rough. At the same time, due to the obvious light changes in the satellite images, it is difficult to clearly identify the satellite components in shadows. In addition, the generalization performance of neural networks needs to be improved when facing new satellite targets.

5.3. Future work

In the future, we will continue to study the intelligent recognition of spacecraft components in the following directions:

Continuous improvement of our dataset: We will try to acquire more kinds of satellite models and build the simulation environment with higher quality. We will also increase the number of pictures per satellite and the type of components.

Continuous improvement of networks: For LUperNet, we will make efforts to refine the recognition results of component boundaries and improve the generalization ability of the network. We will also try to improve the recognition accuracy of new targets and the components in shadows. With the progress of deep learning, new SOTA methods will also be used to improve the recognition effect. In addition, semi or weakly supervised methods and few-shot segmentation are under consideration to deal with the problem of insufficient datasets.

6. Conclusion

This paper builds the first publicly available UE4-based dataset for spacecraft component recognition with five distinctive components: Solar panel, Antenna, Instrument, Thruster, and Optical payload. Based on Unreal Engine 4, we establish a near-earth simulation environment that can generate photorealistic images with high-quality earth backgrounds. After modifying and importing 33 different satellite models into the environment, we obtain 10,000 satellite images with various attitudes and backgrounds. Our dataset, named UESD, can be continuously improved by adding new satellite models and adjusting the simulation environment. To benchmark UESD, a series of SOTA models is used to test their performances. In addition, based on ConvNeXt-B, we propose a simple but effective decoder architecture named LUperNet that can carry out feature fusion as much as possible. For cooperative targets with known equipment distribution, our method achieves 84.6 $\%mIoU$ and shows better performance compared with all baselines. Additional experiments are carried out to test the generalization of our method. The results show that even for new satellites targets, our method can still recognize components, and the accuracy maintains a high level. Furthermore, neural networks trained on our dataset can recognize the solar panels on an entirely different dataset URSO and real satellite images, which proves the universality of UESD and the satisfactory generalization performance of our method. We will continue to improve our dataset, recognition accuracy on component boundaries, components in shadows, and the generality of our method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

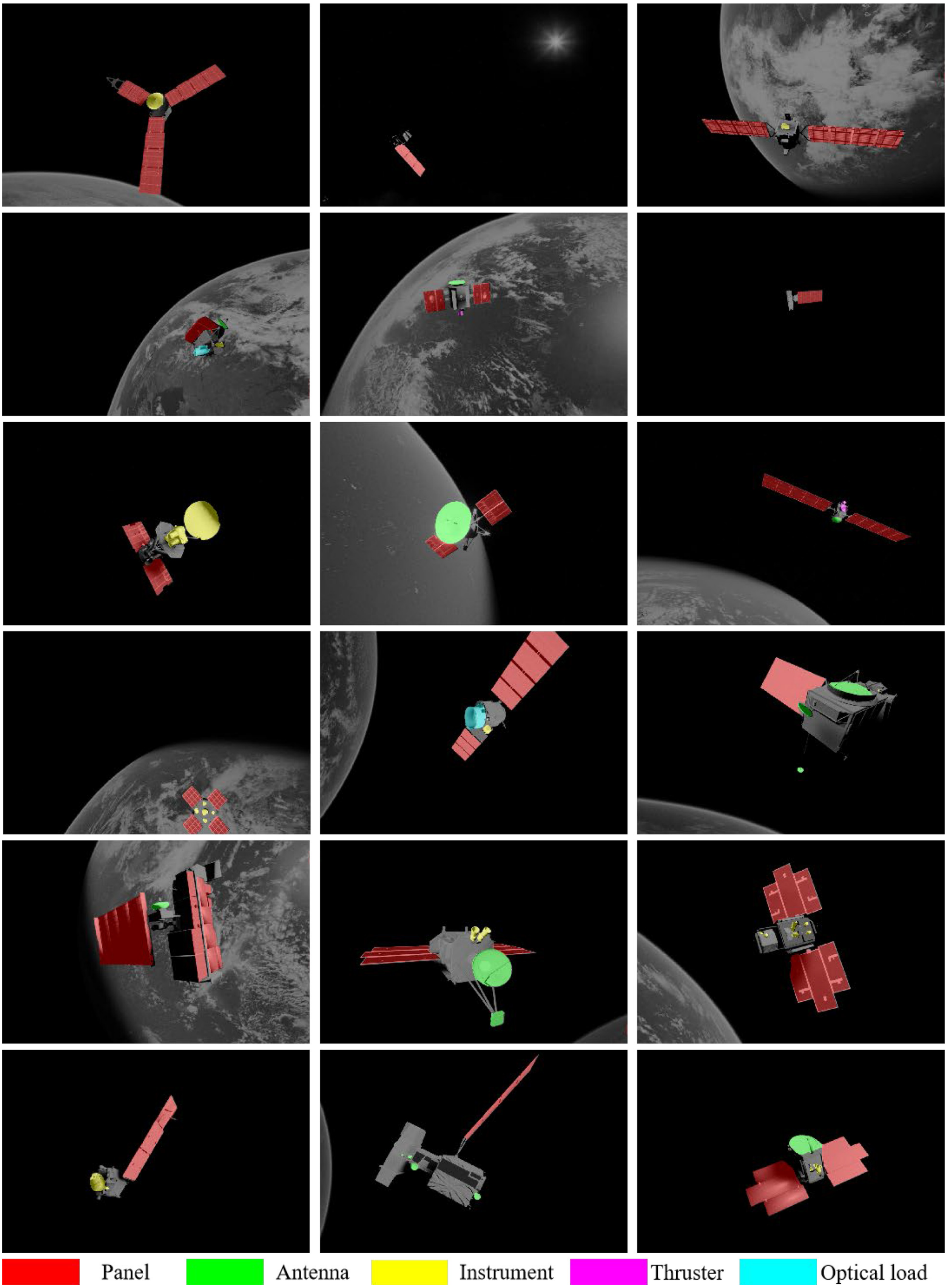

Fig. 8. Results of the remaining 18 satellites.

Acknowledgments

This Study is supported by the National Natural Science Foundation of China (Grant No. 11772023) and The Science and Technology on Space Intelligent Control Laboratory of China (HTKJ2020KL502004).

Appendix A. *A.1*

We conduct experiments on ADE20K (Zhuo et al., 2019) to test the performance of our method. ADE20K contains 150 semantic categories for semantic segmentation. The results of SFnet, OCRnet, VAN-B, and ConvNeXt-B are from the original references of each model. The backbone of our method is ImageNet-22 K pre-trained models. Other training settings are the same as ConvNeXt-B. Quantitative results of different methods on ADE20K are shown in Table 3. Our method surpasses ConvNeXt-B by 0.32*mIoU*, which proves the effectiveness of LUperNet.

A.2

Fig. 8 shows the results of the remaining 18 satellites of UESD dataset. Due to the space limitation, more results can be found in our GitHub.

A.3

In Section 5.2, we conduct experiments on different datasets to test the performance of our method. Table 4 shows the quantitative results of each component. Due to the significant difference between the validation set and the testing set, the recognition accuracy of solar panel and antenna changes obviously (e.g., 85.28 % vs 73.85 %, 83.21 % vs 89.18 %). Our method shows significant improvement in the validation set and is 8.77 and 1.55 *mIoU* higher than SFnet and ConvNeXt-B, respectively. However, our method does not improve significantly for new targets in the testing set.

We could find that there is even a drop in solar panel (75.95 % vs 76.04 %) of our method. We analyze there are two possible reasons. First, the trainval set (23 satellites) and the test set (10 satellites) have obvious differences in the shape of the panel, such as the Juno or Kepler satellites. Therefore, the accuracy of the neural network trained on the training set is decreased when facing the test set with great differences. Second, our method carries out feature fusion as much as possible, which means that our method has a stronger feature extraction ability, so it surpasses other methods in relevant experiments. We deem that a stronger feature extraction ability may make the neural network closer to the target features in the training set. Therefore, when facing the test set with a large difference from the training set, the generalization performance will decline.

Overall, however, our method still surpasses ConvNeXt-B by 0.38 *mIoU* on the testing set and 0.89 *mIoU* on the URSO dataset.

The original intention of the above experiments is to simulate the recognition of new satellites in the real situation. For new targets, the drop in recognition accuracy is expected. Nevertheless, our method can still recognize the obvious components and has higher accuracy than the other two models, which further proves that our method can better meet the challenge of real missions. We will continue to improve the recognition accuracy of neural networks in the face of new satellite targets.

References

Alexey, D., German, R., Felipe, C., Antonio, L., Vladlen, K., 2017. CARLA: An Open Urban Driving Simulator. In: Proc. 1st Annual Conference on Robot Learning, pp. 1–16. https://doi.org/10.48550/arXiv.1711.03938.

Alexey, D., Lucas, B., Alexander, K., Dirk, W., Xiao, H.Z., Thomas, U., Mostafa, D., Matthias, M., Georg, H., Sylvain, G., Jakob, U., Neil, H., 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: Proc. International Conference for LearnIng Representations (ICLR). https://doi.org/10.48550/arXiv.2010.11929.

Anthea, C., Florent, M., Vincent, D., Davide, C., Emmanuel, Z., Christine, E., 2021. Robust Navigation Solution for Vision-Based Autonomous Rendezvous. In: Proc. 2021 IEEE Aerospace Conference (50100), pp. 1–14. https://doi.org/10.1109/AERO50100.2021.9438241.

Ashish, V., Noam, S., Niki, P., Jakob, U., Llion, J., Aidan, N.G., Lukasz, K., Polosukhin, L., 2017. Attention Is ALL You Need. In: Proc. 31st International Conference on Neural Information Processing Systems (NIPS). https://doi.org/10.48550/arXiv.1706.03762.

Bay, H., Tuytelaars, T., Surf, V.G.L., 2006. Speeded Up Robust Features. In: Proc. European Conference on Computer Vision (ECCV). https://doi.org/10.1007/11744023_32.

Bo, L., Wei, W., Qiang, W., Fang, Y.Z., Jun, L.X., Jun, J.Y., 2019. Siamrpn++: Evolution of Siamese visual tracking with very deep networks. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4277–4286. https://doi.org/10.1109/CVPR.2019.00441.

Cai, J., Huang, P., Chen, L., Bin, Z., 2015. A Fast Detection Method of Arbitrary Triangles for Tethered Space Robot. In: . *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 120–125. https://doi.org/10.1109/ROBIO.2015.7418754.

Chen, B., Cao, J., Parra, A., Chin, T.J., 2019. Satellite Pose Estimation with Deep Landmark Regression and Nonlinear Pose Refinement. In: Proc. IEEE International Conference on Computer Vision Workshop (ICCVW), pp. 2816–2824. https://doi.org/10.1109/ICCVW.2019.00343.

Chen, Y.L., Gao, J.M., Zhang, Y., Zheng, D., 2021. Satellite Components Detection from Optical Images Based on Instance Segmentation Networks. J. Aerospace Informat. Syst 18, 6. https://doi.org/10.2514/1.I010888.

Dan, H., Kevin, G., 2020. Gaussian Error Linear Units (GELUs). arXiv:1606.08415. https://doi.org/10.48550/arXiv.1606.08415.

David, N., Diane, L., Andrea, V., 2017. Anchor-net: A weakly supervised network to learn geometry-sensitive features for semantic matching. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2867–2876. https://doi.org/10.1109/CVPR.2017.306.

Diederik, P.K., Jimmy, B., 2015. Adam: A Meth for Stochastic Optimization. In: Proc. International Conference for Learning Representations (ICLR). https://doi.org/10.48550/arXiv.1412.6980.

Gao, H., Zhuang, L., Laurens, M., Kilian, W., 2017. Densely connected convolutional networks. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

Hang, B.B., Li, D., Fu, R.W., 2021. BEiT: BERT Pre-Training of Image Transformer. arXiv: 2106.08254, https://doi.org/10.48550/arXiv.2106.08254.

Hao, Y.Z., Yong, X., 2017. Space Target Recognition based on Deep Learning. In: Proc. 20th International Conference on Information Fusion. https://doi.org/10.23919/ICIF.2017.8009786.

Heng, S.Z., Jian, P.S., Xiao, J.Q., Xiao, G.W., Jia, Y.L., 2017. Pyramid Scene Parsing Network. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239. https://doi.org/10.1109/CVPR.2017.660.

Hoang, D.A., Chen, B., Chin, T.J., 2021. A spacecraft dataset for detection, segmentation and parts recognition. arXiv: 2106.08186,. https://doi.org/10.48550/arXiv.2106.08186.

Hyeonseob, N., Bohyung, H., 2016. Learning multi-domain convolutional neural networks for visual tracking. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4293–4302. https://doi.org/10.1109/CVPR.2016.465.

Hyeonseob, N., Seunghoon, H., Bohyung, H., 2015. Learning deconvolution network for semantic segmentation. In: Proc. IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/ICCV.2015.178.

Jie, H., Li, S., Gang, S., 2018. Squeeze-and-Excitation Networks. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7132–7141. https://doi.org/10.1109/CVPR.2018.00745.

Jimmy, L.B., Jamie, R.K., Geoffrey, E.H., 2016. Layer Normalization. arXiv: 1607.06450, https://doi.org/10.48550/arXiv.1607.06450.

Jonathan, L., Evan, S., Trevor, D., 2014. Fully Convolutional Networks for Semantic Segmentation. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440. https://doi.org/10.48550/arXiv.1411.4038.

Ju, H.M., Jong, M.L., Jean, P., Cho, M., 2020. Learning to compose hypercolumns for visual correspondence. In: *Proc. European Conference on Computer Vision (ECCV)*. https://doi.org/10.48550/arXiv.2007.10587.

Jun, F., Jing, L., Hai, J.T., Yong, L., Yong, J.B., Zhi, W.F., Han, Q.L., 2019. Dual Attention Network for Scene Segmentation. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3141–3149. https://doi.org/10.1109/CVPR.2019.00326.

Kai, M.H., Xiang, Y.Z., Shao, Q.R., Jian, S., 2016. Deep residual learning for image recognition. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

Karen, S., Andrew, Z., 2015. Very deep convolutional networks for large-scale image recognition. In: *Proc. International Conference on Learning Representations (ICLR)*. https://doi.org/10.48550/arXiv.1409.1556.

Kevin, C., Minh, T.L., Quoc, V.L., Christopher, D.M., 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In: Proc. International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.2003.10555.

Li, L.Z., Zhang, T., 2020. Feature detection and recognition of spatial noncooperative objects based on deep learning (In Chinese). CAAI Trans. Intell. Syst. 15 (6), 1154–1162 https://kns.cnki.net/kcms/detail/detail.aspx?FileName=ZNXT202006019&DbName=CJFQ2020.

Liang, C.C., George, P., Isonas, K., Kevin, M., Alan, L.Y., 2014. Semantic Image Segmentation with Deep Convolutional Nets and fully Connected CRFs. arXiv:1412.7062, https://doi.org/10.48550/arXiv.1412.7062.

Liang, C.C., George, P., Iasonas, K., Kevin, M., Alan, L.Y., 2017. Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFS. IEEE Trans. Pattern Anal. Mach. Intell. 40 (4), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184.

Liang, C.C., Yu, K.Z., George, P., Florian, S., Hartwig, A., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Proc. European Conference on Computer Vision (ECCV)*. https://doi.org/10.48550/arXiv.1802.02611.

Llya, L., Frank, H., 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. *Proc. International Conference on Learning Representations (ICLR)* https://arxiv.org/abs/1608.03983v5.

Mate, K., Sumant, S., Tae, H.P., Dario, I., Marcus, M., Simone, D., 2020. Satellite Pose Estimation Challenge: Dataset, Competition Design, and Results. IEEE Trans. Aerosp. Electron. Syst. 56 (5), 4083–4098. https://doi.org/10.1109/TAES.2020.2989063.

Meng, H.G., Cheng, Z.L., Zheng, N.L., Ming, M.C., Shi, M.H., 2022. Visual Attention Network. arXiv:2202.09741, https://doi.org/10.48550/arXiv.2202.09741.

MMSegmentation Contributors. 2020. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation.

Olaf, R., Philipp, F., Thomas, B., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention, https://doi.org/10.48550/arXiv.1505.04597.

Pablo, M.G., Sergiu, O., Alberto, G.G., Alvaro, J.A., Siegio, O.E., Jose, G.R., 2018. UnrealRox: An Extremely Photorealistic Virtual Reality Environment for Robitics Simulations and Synthetic Data Generation. arXiv: 1810.06936. https://doi.org/10.48550/arXiv.1810.06936.

Pedro, F.P., Yang, G., 2019. Deep Learning for Spacecraft Pose Estimation from Photorealistic Rendering. arXiv: 1907.04298, https://doi.org/10.48550/arXiv.1907.04298.

Qi, H., Ze, J.F., Qi, D., Lei, S., Ming, M.C., Jia, Y.L., Jing, D.W., 2022. On the Connection between Local Attention and Dynamic Depth-wise Convolution. In: *Proc. International Conference on Learning Representations (ICIR)*. https://doi.org/10.48550/arXiv.2106.04263.

Qiang, W., Li, Z., Luca, B., Wei, M.H., Philip, H.S.T., 2019. Fast Online Object Tracking and Segmentation: A Unifying Approach. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.48550/arXiv.1812.05050.

Roland, B., Jeremy, L., Cyril, R., Keyvan, K., Gregory, J., Aurore, M., Noela, D., Ahmad, B., 2018. Scientific image rendering for space scenes with the SurRender software. In: *Proc. 69th International Astronautical Congress (IAC)*. https://doi.org/10.48550/arXiv.1810.01423.

Shital, S., Debadeepta, D., Chris, L., AirSim, A.K., 2017. High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In: Proc. Field and Service Robotics Conference. https://doi.org/10.48550/arXiv.1705.05065.

Shuai, L., 2019. Intelligent Control and Recognition of Space Robot Capturing Non-cooperative Targets. Dalian University of Technology.

Shuai, Y.H., Qin, Y.W., Song, Y.Z., Shi, P.Y., Xu, M.H., 2019. Dynamic context correspondence network for semantic alignment. In: *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 2010–2019. https://doi.org/10.1109/ICCV.2019.00210.

Si, X.Z., Jia, C.L., Heng, S.Z., Xia, T.Z., Ze, K.L., Ya, B.W., Yan, W.F., Jian, F.F., Tao, X., Philip, H.S., Li, Z., 2021. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6877–6886. https://doi.org/10.48550/arXiv.2012.15840.

Steve, M.P., Martin, I., Dunstan, M., Matthews, D., 2013. Plant Surface Simulation with PANGU. In: *Proc. Space OPS 2004 Conference*. https://doi.org/10.2514/6.2004-592-389.

Sumant, S., Simone, D., 2019. Pose Estimation for Non-Cooperative Rendezvous Using Neural Networks. Proc. AAS/AIAA Astrodynamics Specialist Conference https://doi.org/10.48550/arXiv.1906.09868.

Te, T.X., Ying, C.L., Bo, L.Z., Yu, N.J., Jian, S., 2018. Unified Perceptual Parsing for Scene Understanding. In: *Proc. European Conference on Computer Vision (ECCV)*. https://doi.org/10.48550/arXiv.1807.10221.

Wen, J.L., Yu, J.L., Raquel, U., Richard, Z., 2016. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In: Proc. Neural Information Processing Systems (NIPS). https://doi.org/10.48550/arXiv.1701.04128.

Xiang, T.L., An, S.Y., Zhen, Z., Hou, L.Z., Mao, K.Y., Kui, Y.Y., Yun, H.T., 2020. Semantic Flow for Fast and Accurate Scene Parsing. In: Proc. European Conference on Computer Vision (ECCV). https://doi.org/10.48550/arXiv.2002.10120.

Xiang, X.C., Zhi, T., Yu, Q.W., Bo, Z., Hai, B.R., Xiao, L.W., Hua, X.X., Twins, C.H.S., 2021. Revisiting the Design of Spatial Attention in Vision Transformer. In: *Proc. International Conference on Neural Information Processing Systems (NeurIPS)*. https://doi.org/10.48550/arXiv.2104.13840.

Xiao, H.D., Xiang, Y.Z., Yi, Z.Z., Jun, G.H., Gui, G.D., Jian, S., 2022. Scaling Up Your Kernels to 31x31: Revisiting Large Kernel Design in CNNs. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.48550/arXiv.2203.06717.

Xu, Y.F., Zhang, D.Z., Wang, L., Hua, B.C., 2020. Lightweight feature fusion network design for local feature recognition of non-cooperative target(In Chinese). Infrared Laser Eng.. 49, 7. https://doi.org/10.3788/IRLA20200170.

Yu, H.Y., Xiao, K.C., Xi, L.C., Jing, D.W., 2020. Segmentation Transformer: Object-Contextual Representations for Semantic Segmentation. In: *Proc. European Conference on Computer Vision (ECCV)*. https://doi.org/10.48550/arXiv.1909.11065.

Ze, L., Yu, T.L., Yue, G., Han, H., Yi, X.W., Zheng, Z., Stephen, L., Baining, G., 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: In *Proc. International Conference on Learning Representation (ICLR)*. https://doi.org/10.48550/arXiv.2103.14030.

Zhang, H.P., Liu, Z.Y., Jiang, Z.G., An, M., Zhao, D.P., 2010. BUAA-SID 1.0 Space Object Image Dataset (In Chinese). Spacecraft Recovery Remote Sens. 31 (4) https://kns.cnki.net/kcms/detail/detail.aspx?FileName=HFYG201004014&DbName=CJFQ2010.

Zhuang, L., Han, Z.M., Chao, Y.W., Christoph, F., Trevor, D., Saining, X., 2022. A ConvNet for the 2020s. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.48550/arXiv.2201.03545.

Zhuo, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A., 2019. Semantic understanding of scenes through the ade20k dataset. Int. J. Comput. Vision 127, 302–321. https://doi.org/10.1007/s11263-018-1140-0.