



OPEN

DATA DESCRIPTOR

# A Large-Scale Synthetic Benchmark Dataset for Non-Cooperative Space Target Perception

Yuxuan Liu<sup>1,2</sup>✉, Chunjiang Bian<sup>1</sup>, Hongbin Nie<sup>1</sup>, Shi Chen<sup>1</sup> & Ziqian Yang<sup>1,2</sup>

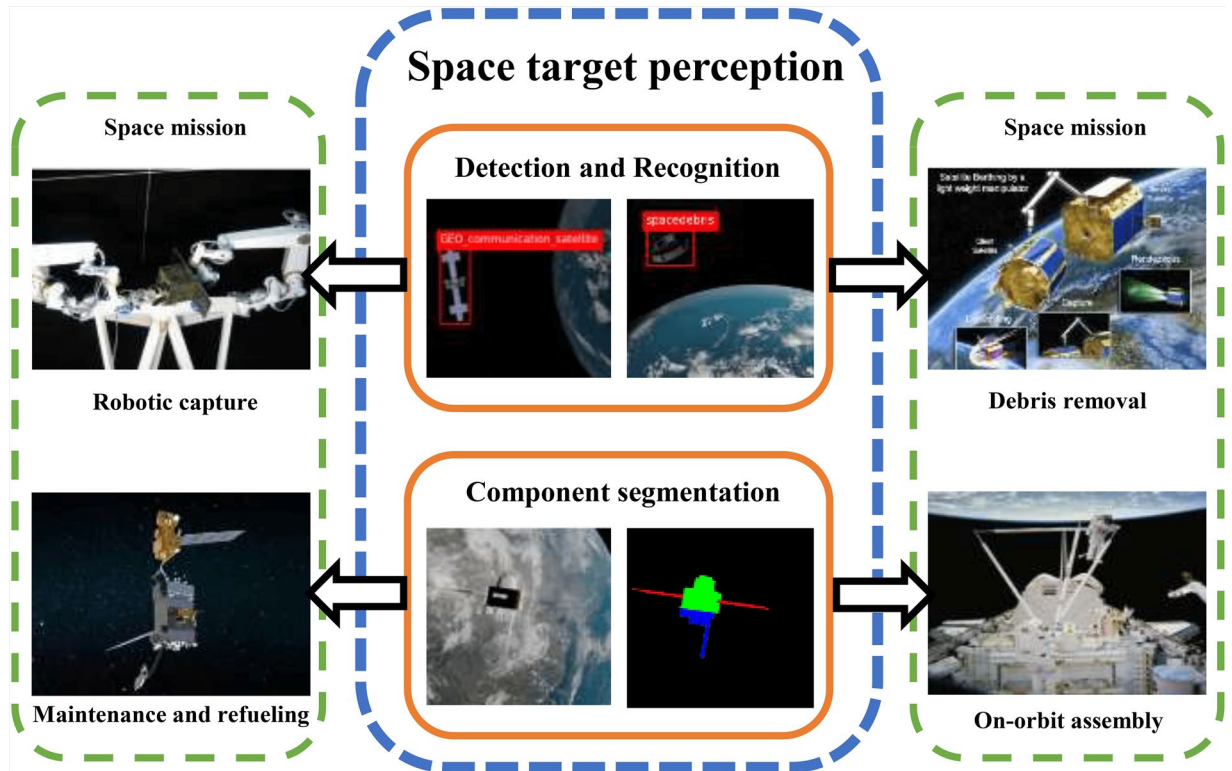
The automatic, accurate perception of targets in space is a crucial prerequisite for many on-orbit aerospace missions. Therefore, research on perception technologies within spaceborne images is meaningful. The development of deep learning has revealed its potential for application to space target perception. However, implementing deep learning models requires large-scale labelled datasets. Therefore, we build a multitask synthetic benchmark space target dataset, NCSTP, to address the limitations of current datasets. First, we collect and modify various space target models for satellites, space debris, and space rocks. By importing them into a realistic space environment simulated by Blender, 200,000 images are generated with different target sizes, poses, lighting conditions, and backgrounds. Then, the data are annotated to ensure the dataset supports simultaneous space target detection, recognition and component segmentation. All data can be used for training space target detection and recognition models. We further annotate the components of each satellite for component segmentation. Finally, we test a series of state-of-the-art object detection and semantic segmentation models on the dataset to establish a benchmark.

## Background & Summary

With the development of space technology and the continuous launch of human spacecraft, the number of non-cooperative space targets is steadily increasing<sup>1,2</sup>. In addition to naturally occurring space rocks, various artificial satellites and debris resulting from the malfunction and disintegration of spacecraft are widely distributed in orbit. Throughout this paper, the term non-cooperative target is used in the rendezvous-and-proximity-operations (RPO) context. Such a target passively drifts in orbit, provides no relative navigation aids or telemetry, and lacks active attitude control. Therefore, these targets must be perceived through vision-based algorithms. To improve satellite safety and spacecraft life, developing on-orbit servicing technologies is crucial. As shown in Fig. 1, accurate aerospace perception<sup>3</sup> is a critical prerequisite for successful on-orbit missions. Tasks such as target monitoring<sup>4</sup> and debris removal<sup>5</sup> rely on space target detection, so these tasks require detecting targets quickly and identifying their types. However, on-orbit assembly<sup>6</sup> and maintenance<sup>7</sup> require the identification of components on satellites, these tasks are based on component segmentation techniques, where various components are differentiated by their semantic labels. Traditional perception methods rely on ground-based optical sensors that are limited by atmospheric turbulence and adverse weather, reducing the accuracy and speed needed for on-orbit servicing tasks. Imagery from spaceborne sensors is more robust in achieving accurate, real-time perception of targets in space.

Artificial intelligence technologies, particularly deep learning, have developed rapidly in recent years. The biggest challenge in applying deep learning to space target perception is the lack of high-quality, large-scale annotated datasets. Obtaining sufficient images of actual targets is extremely difficult, unlike natural images. However, data are the core of deep learning models, and the quality of the training data largely determines the model's performance. Therefore, a dataset that best simulates the actual space environment and contains many well-annotated space targets is urgently needed. In addition, current datasets are typically designed for specific perception tasks, such as space debris detection or satellite antenna recognition. However, actual on-orbit servicing requires the perception of multiple aspects of the same target. Establishing a space target dataset that supports multiple perception tasks is also essential.

<sup>1</sup>National Space Science Center, Chinese Academy of Sciences, Beijing, China. <sup>2</sup>University of Chinese Academy of Sciences, Beijing, China. ✉e-mail: liuyuxuan231@mails.ucas.ac.cn



**Fig. 1** Relationship between space target perception and space missions.

Motivated by these challenges, this study created a benchmark dataset that supports both space target detection and component segmentation tasks. With the help of Blender software, we simulated a realistic space environment that included sunlight illumination, the Earth, a background of stars, and a spaceborne observation camera.

Next, we collected and refined 3D models of 26 space targets, including satellites, space debris, and space rocks. After importing the models into the simulation, we analyzed the surface materials, textures, orbital characteristics, and operational states of the different space targets. We further refined and adjusted the models to make the scene more realistic. By randomly varying lighting conditions and camera positions, we generated 200,000 images to support the subsequent training and validation of deep learning models.

The generated samples were annotated to meet the requirements of both space target detection and component segmentation. For the object detection task, we divided all satellites into 10 categories based on their functions and distinct visual features, along with space debris and space rocks, resulting in 12 categories. Each image was annotated with bounding boxes and corresponding categories. Subsequently, we extracted all images in the dataset of the satellite target category. We used semantic segmentation annotations to label four distinct components: body, solar panels, antenna, and observation payloads. Note that our annotation method bridges two space target perception tasks. The model can detect and classify the type of satellite in an image and segment its components. This approach has not been explored in previous research.

After constructing the dataset, we conducted a statistical analysis of the space targets in the dataset, calculating their distribution according to their size, position, and other relevant characteristics. Further, we selected 10 representative object detection methods and 10 semantic segmentation methods to test their performance on our dataset. We also analyzed the methods' performance across different target categories for target detection. Similarly, for component segmentation, we evaluated the performance of each algorithm on different components. Beyond the supervised baseline models reported here, NCSTP is designed to support hybrid and self-supervised training paradigms, which hold strong potential for improving synthetic-to-real transfer in space imagery.

According to the results, we analyzed the performance of current methods and the characteristics and task challenges of the dataset, and also outline future directions for research.

**Related dataset.** Many scholars have tried constructing space target datasets with different methods for perceptual tasks in recent years. The spacecraft pose estimation dataset (SPEED)<sup>8</sup> was initially released by ESA and Stanford University to clarify satellite pose estimation. Because of such limitations as restricted pose, single-lighting conditions, and small datasets, this study developed an improved version of the Next-Generation Dataset for Spacecraft Pose Estimation (SPEED+)<sup>9</sup>. URSO<sup>10</sup> comprises synthetic and actual images of the Soyuz and Dragon spacecraft in space, one dataset for the Dragon spacecraft and two datasets for the Soyuz model with different operating ranges. Hoang *et al.*<sup>11</sup> collected 3117 images of satellites and space stations, including

	Images	Image Sources	Satellite	Space debris	Space rock	Target classes	Component classes	Supported task
SPEED	15,300	Synthetic captured	1	—	—	1	—	Pose estimation
SPEED+	69,531	Synthetic captured	1	—	—	1	—	Pose estimation
URSO	5,000	Synthetic	2	—	—	2	—	Pose estimation
BUAA-SID1.0	5,000	Synthetic	20	—	—	20	—	Recognition
Satellite Dataset	3,117	collect online	√	√	—	—	3	Detection Component segmentation
DSTD	50,000	Synthetic	11	35	—	6	—	Recognition
UESD	10,000	Synthetic	33	—	—	—	5	Component segmentation
SPARK	150,000	Synthetic	10	5	—	15	—	Detection Recognition
STAR-24K	24,000	collect online	√	√	√	3	—	Detection Recognition
Ours	200,000	Synthetic	16	6	4	3(coarse) 10(fine)	4	Detection Recognition Component segmentation

**Table 1.** Quantitative statistics for each dataset.

synthetic and authentic images. It is the first publicly available dataset for space target detection and component segmentation. SPARK is a multi-modal dataset that provides paired RGB and depth images. The dataset contains 150,000 images, including 11 types of spacecraft and space debris. Zhang *et al.*<sup>12</sup> proposed a space target recognition dataset containing 50,000 Blender images, including 11 classes of satellites and 35 classes of space debris. STAR-24K<sup>13</sup> is a dataset for space target detection with approximately 24,000 images. It was constructed by extracting images released by the NASA and ESA websites. UESD is a UE4-based dataset proposed by Zhao, *et al.*<sup>14</sup> It is used for spacecraft component segmentation. In Table 1, We compared NCSTP with other space target datasets from multiple considerations, such as the number of images, target categories, annotation methods, and supported perception tasks.

The contributions of our work are summarized as follows.

- (1) This paper constructs a large-scale non-cooperative space target dataset, integrating multiple perception tasks, including space target detection and component segmentation. Fine-grained recognition and sequential perception tasks for the satellite targets are supported, detecting the satellite and then segmenting its components. The NCSTP dataset provides rich, high-quality data samples and benchmark evaluation conditions for space target perception research.
- (2) We present a comprehensive and systematic review of datasets and methods related to non-cooperative space target perception tasks, including space target detection and component segmentation, with analysis and perspective on future research trends.
- (3) This article evaluates the performance of 20 advanced methods for space target detection and component segmentation on NCSTP. It also presents a direction to improve tasks according to challenges and the characteristics of the dataset.

## Methods

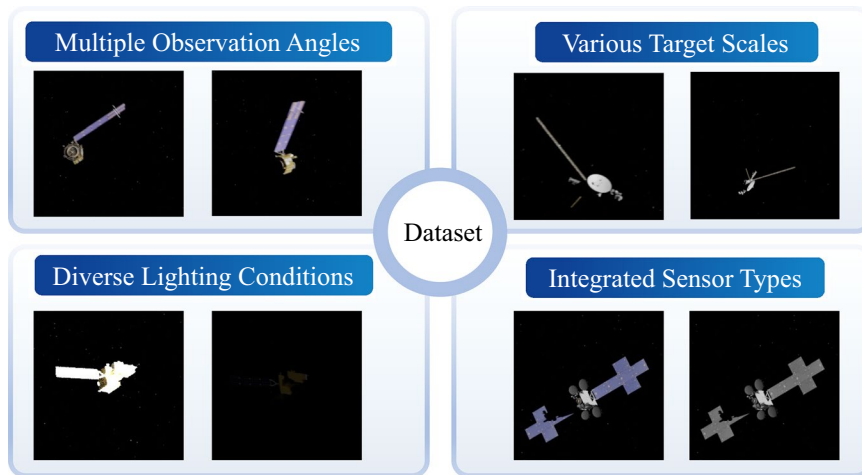
This section provides a detailed description of the generation and annotation process of the NCSTP dataset.

**Data generation.** Training deep learning models requires many samples. However, the availability of directly collected 2D images of space targets is limited. To generate a rich and diverse set of space target images, we chose to use 3D models. After importing the models with optimized material types and spectral characteristics into the simulated space environment, we generated multiple samples of the same target by varying attributes such as size, pose, lighting condition and other properties.

**Space target modelling.** The simulation of space objects can be divided into two parts: structural modeling and optical property modeling. The former involves constructing a three-dimensional geometric model of the object, while the latter requires selecting appropriate materials for the object to ensure it exhibits realistic surface spectral characteristics. We collected 3D models of space targets from official NASA repositories to ensure their reliability and authenticity (<https://science.nasa.gov/3d-resources/>). There were 26 types of 3D models, comprising 16 types of satellites, 6 types of space debris, and 4 types of space rocks.

Although the three-dimensional models obtained by these authoritative agencies ensure the accuracy of the geometric structures, the material and texture information of these models is incomplete. Therefore, we inspected each model and added high-resolution texture maps and adjust its material properties based on the types of materials used in actual satellites.

Notably, the surface of the satellite body is typically covered with a layer of golden material. This material consists of a polymer compound of polyimide, combined with thin metallic films of gold, silver, or aluminum. It is commonly referred to as multi-layer-insulation (MLI). The primary material used in solar panels is silicon. Silicon is a crucial semiconductor material with strong absorption properties for sunlight. Antennas are typically constructed using aluminum alloy as the primary material, characterized by a gray or silver-gray appearance. This material offers excellent corrosion resistance, thermal conductivity, and electrical conductivity. The material



**Fig. 2** Multiple imaging conditions in our dataset. Examples illustrate pose changes from diverse view angles, scale variation due to range, illumination extremes under different Sun–target–sensor geometries and sensor modalities.

of the observation payload is typically related to the type of payload. Here, we model it using a Disney-type mixture of metallic and non-metallic components.

Different materials exhibit distinct thermal physical parameters and optical properties. The reflectivity, metallic, roughness, and IOR of a material all significantly influence its radiative characteristics. In Blender, we carefully configured these parameters to closely align with the actual properties of the materials.

*Simulated space environment.* In this study, we simulated a space environment using Blender software. In addition to the optimized 3D target models, the simulation included the Earth, an observation camera, a star background, and a light source that mimicked the Sun.

Due to the significant distance between space objects and the light source, the sunlight in the scene can be approximated as parallel light that uniformly illuminates the entire target. To simulate this scenario, we utilize the ‘Sun’ type light source in Blender, which is positioned at an infinite distance. The star background is emulated in Blender using star-field textures to approximate realistic stellar density and brightness.

The Earth model also effectively replicated features of the Earth’s surface, such as mountains, deserts, forests, and plains. Surrounding the Earth was an atmospheric layer model, which greatly enhanced the simulated Earth’s appearance. Additionally, the application of different texture maps on the Earth model are controlled by functions to vary Earth backgrounds under different lighting conditions, which made the images more closely resemble the real-space scenarios encountered in service missions.

To achieve realistic rendering of lighting and material appearances, it is necessary to calculate the radiance reaching the camera after accounting for the physical interactions of light, such as reflection and refraction, on the surfaces of spatial objects. This process requires analyzing the spectral reflectance properties of the object surfaces, specifically the variation of reflectance with respect to wavelength. Our approach primarily involves the definition of a Bidirectional Reflectance Distribution Function (BRDF) model (Eq. 1).

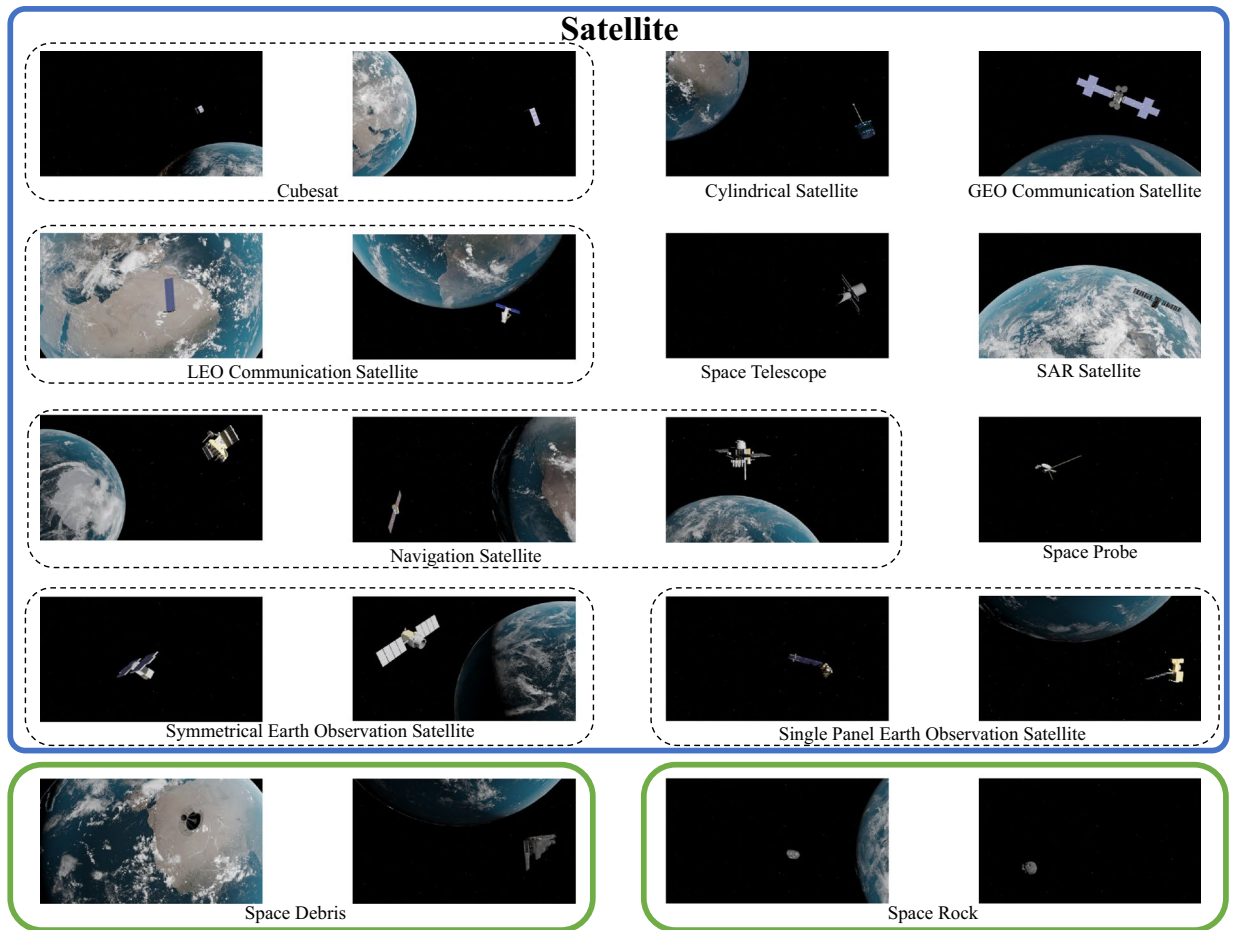
$$f(p, \omega_i, \omega_o) = \frac{dL_o(p, \omega_o)}{dE_i(p, \omega_i)} \quad (1)$$

Here,  $p$  denotes the surface point being shaded. The incident lighting direction  $\omega_i$  is defined as the unit vector pointing from the surface toward the light source, whereas the view direction  $\omega_o$  is the unit vector pointing from the surface toward the camera. The term  $L_o(p, \omega_o)$  represents the outgoing radiance at  $p$  in the direction  $\omega_o$ , and  $dE_i(p, \omega_i)$  denotes the differential irradiance incident at  $p$  from the solid angle around  $\omega_i$ .

The spectral reflectance characteristics vary across different materials. The Cook-Torrance reflection model (Eq. 2), a key component of BRDF, can simulate micro-surface reflections on metallic and non-metallic surfaces. Therefore, we ultimately selected this model to calculate the radiance received by the camera after accounting for physical interactions, such as reflection and refraction, of light with the surfaces of various components of the space target.

$$f_r = \frac{K_d}{\pi} + \frac{K_s}{\pi} \cdot \frac{F \cdot D \cdot G}{\cos\theta_i \cdot \cos\theta_r} \quad (2)$$

Here,  $\theta_i$  denotes the incident zenith angle between the surface normal  $n$  and the incident direction  $l$ , while  $\theta_r$  denotes the reflection zenith angle between  $n$  and the view direction  $v$ . The diffuse reflectance coefficient is represented by  $K_d$ , and the specular reflectance coefficient by  $K_s$ . In addition,  $F$ ,  $D$  and  $G$  denote the Fresnel term, the normal-distribution function of microfacets, and the geometric attenuation term, respectively.



**Fig. 3** Example of generated data and two-level space target categories.

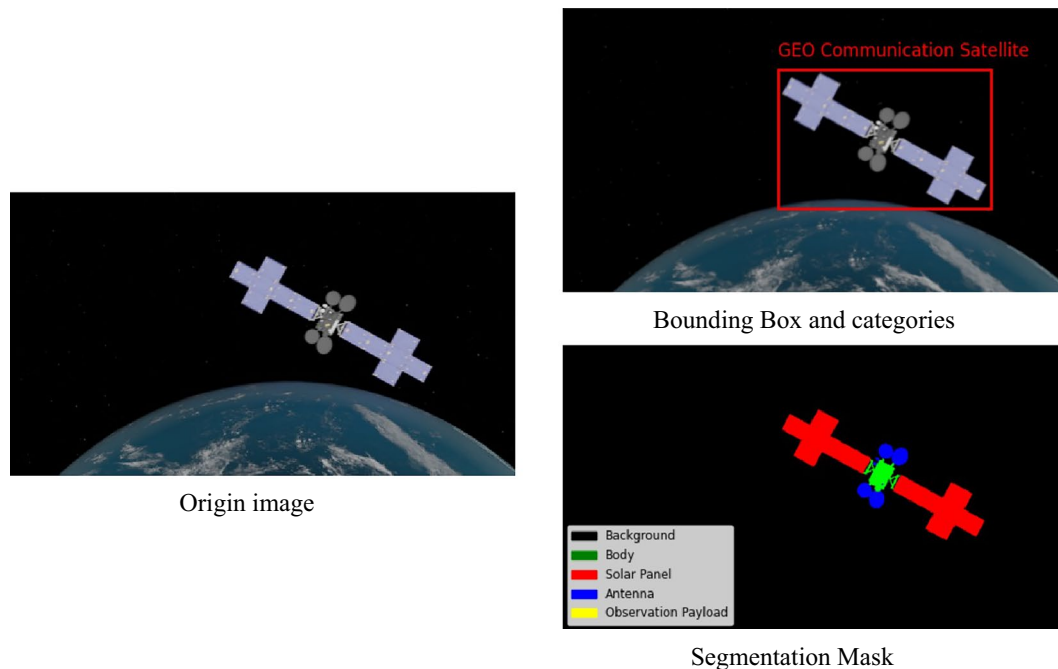
We utilized the Principled BSDF shader within Blender, along with the Cycles rendering engine, to implement this optical model. Through the modelling of the light source and the spectral reflectance characteristics of the material on the surface of the spatial target, we have achieved a lighting effect that closely approximates real-world conditions in space.

After the simulated environment was set up and the optimized 3D models imported, we conducted a comprehensive survey of each target's functions and orbital distribution to enable us to generate trajectories for each model. To generate videos, the targets' motion in space was stimulated by continuously altering the target's pose and position, as well as the Earth parameters and lighting conditions. By extracting frames from the video, we obtained the required target images. For each spatial target in the generated data, Fig. 2 shows the different poses caused by various observation angles, different scales resulting from different observation distances, extreme imaging conditions such as high exposure or low lighting due to varying positions relative to the Sun, and RGB and grayscale images captured by different imaging sensors.

**Data annotation.** We designed a two-level space target classification scheme for this study. Based on the origin of the targets, they were classified as satellites, space debris, or space rocks, as shown in Fig. 3. Because space missions require further identification of the types of satellites and their components, we performed a fine-grained classification for all categorized satellites. The functions and structural features were used to divide the satellites into 10 fine-grained categories: CubeSats, cylindrical satellites, geo-communication satellites, LEO communication satellites, the space telescope, SAR satellites, navigation satellites, symmetrical Earth observation satellites, and single-panel Earth observation satellites.

After the two-level classification rules were defined, we classified the images of the 26 space targets previously generated. Specifically, the images of the six types of space debris were categorized as space debris, and the four types of space rocks were classified as space rocks. The 16 types of satellite target images were then further categorized into 10 fine-grained satellite categories according to the second-level classification rules.

Based on the two-level classification criteria, we annotated the boundary boxes and categories of all targets in the images to meet the target detection and recognition requirements. All satellite components in the images identified as "satellite" were further annotated in the form of masks to support component recognition. Four components, solar panels, antennas, bodies, and observation payloads, were annotated according to the structural diagram of each type of satellite. Since our source 3D models arrive in heterogeneous formats and



**Fig. 4** Multi-task annotation provided by our dataset.

conventions, we normalize these component variants into a shared taxonomy so that models learn function rather than shape style.

Solar panels absorb energy from the Sun to power the satellite while the antenna collects signals. The body is the satellite's core, where various support systems are typically integrated. The observation payload typically includes optical and infrared cameras. Because the payload often needs to be oriented toward the target, it is not usually integrated with the body, so we annotated it as a separate component.

The labels for object detection consist of the coordinates of the target's bounding box and the target category. The labels for semantic segmentation are mask images of the same size as the original image, where each pixel value represents a specific component. Figure 4 shows the annotation results for a satellite image.

### Data Records

The NCSTP<sup>15</sup> dataset is available at <https://doi.org/10.6084/m9.figshare.28606754>. Figure 5 visualises the two top-level branches NCSTP\_det\_recog and NCSTP\_comseg, each pre-split into train, valid and test subsets. In the detection branch, lossless PNG frames are stored within the split folders, and the accompanying COCO-style<sup>16</sup> annotation files (*train.json*, *val.json*, *test.json*) are placed in a directory that follows the official COCO field specification. In the segmentation branch, each image is associated with a single-channel, 8-bit mask named *<basename>\_mask.png*.

The final dataset contains 200,000 images with a uniform resolution of  $576 \times 324$  pixels. Each image includes a single space target. We provided annotations for the bounding boxes and category information of the targets for all images so that all images could be used to train and validate object detection and recognition models. We split the NCSTP dataset into training, validation, and test sets at a 7:1:2 ratio, with 140,000, 20,000, and 40,000 images, respectively. Splits were stratified by top-level class (and 10 fine-grained satellite subtypes). We sampled each subset with a fixed seed.

The dataset has 100,000 satellite images. We annotated the components for these images and stored them as mask images with the same resolution. This means that there are a total of 100,000 satellite images of different categories that can be used for model training and validation of component segmentation. Similarly, we split this subset into training, validation, and test sets at a 7:1:2 ratio, with 70,000, 10,000, and 20,000 images, respectively.

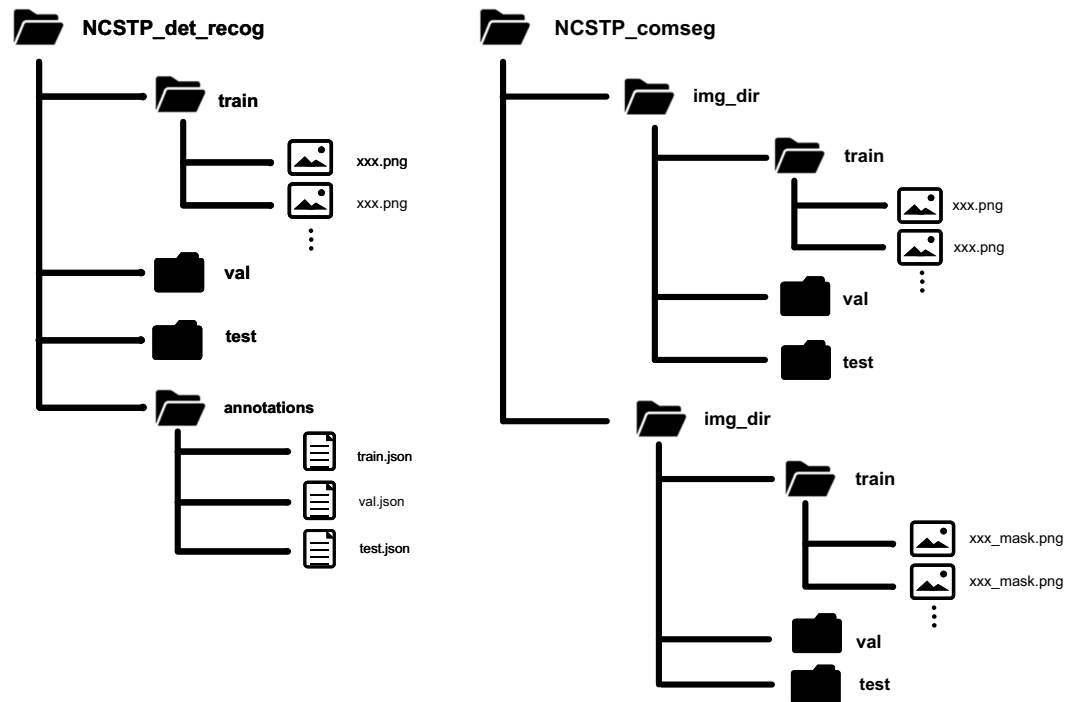
Table 2 shows the statistical results of the dataset for each space target perception task and annotation format.

**Data overview.** The dataset we constructed in this study has unique features and advantages, providing foundational data support for target visual perception tasks in space. Moreover, it presents new challenges for researchers in this field.

We have compiled statistical information on the sample distribution of the NCSTP dataset. Figure 6 illustrates the types of space targets included in the dataset and provides an intuitive visualization of the sample counts for each target class. It also presents a scatter plot of target size distributions. Figure 7 shows the pixel distribution for each component within the satellite category and calculates the proportion of pixels corresponding to each component (excluding the background) in the image.

The features of the NCSTP dataset are summarized as follows:

## NCSTP



**Fig. 5** Directory structure of the NCSTP dataset.

	Images	Annotation	Classes	Supported Perception task
Satellite	100,000	Bounding Box categories Component mask	10	Detection Fine-grained Recognition Component segmentation
Space debris	60,000	Bounding Box categories	—	Detection Recognition
Space rock	40,000	Bounding Box categories	—	Detection Recognition

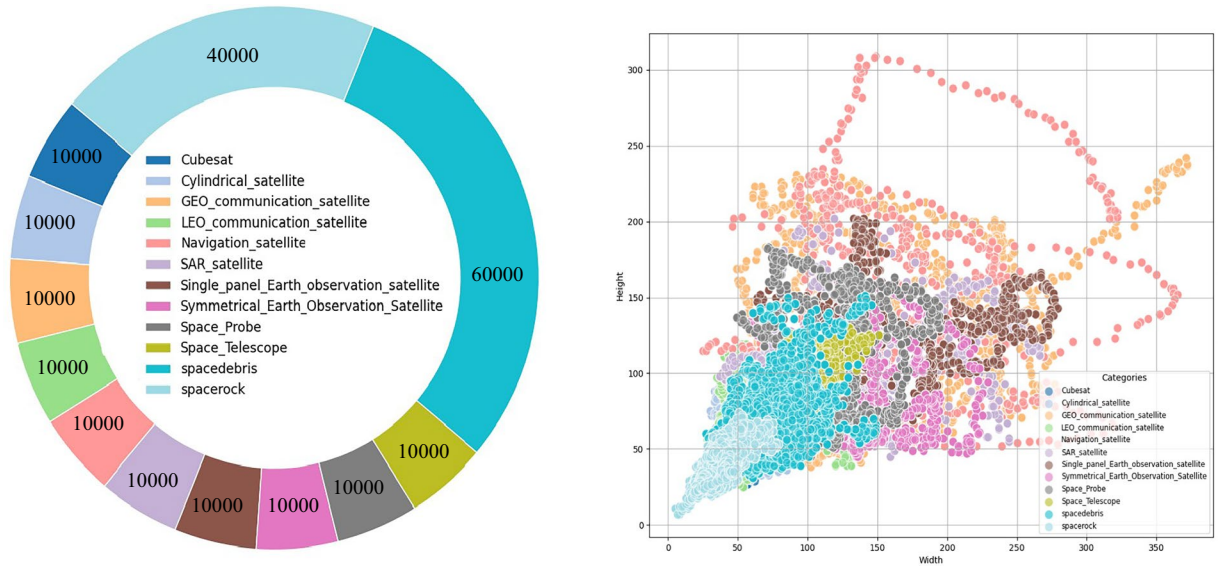
**Table 2.** NCSTP dataset statistics.

- (1) Large scale: Our dataset is currently the most extensive space object dataset, containing 200,000 images. These images can be used for space object detection and recognition tasks. Among them, 100,000 images categorized as satellites can be used for component segmentation tasks.
- (2) Multiple and sequent tasks: Our dataset currently supports the most diverse range of space object perception tasks, including detection, recognition, and satellite component segmentation. Additionally, our dataset supports subsequent tasks for the same object.
- (3) Reasonable classification approach: We designed a two-level classification approach for all space objects. We further performed a fine-grained classification of satellite objects based on their functions and significant structural features.

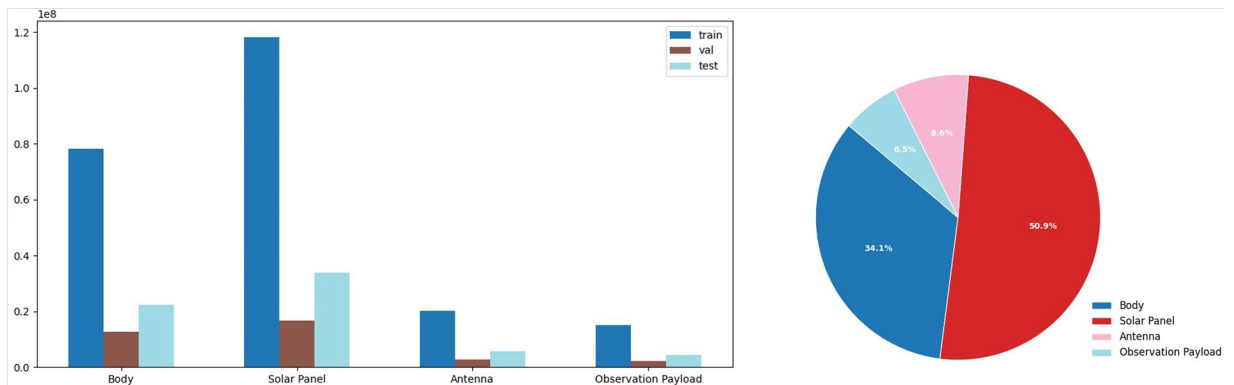
### Technical Validation

Based on the NCSTP, we evaluated and compared the performance of the proposed method with SOTA models. The technical validation included two parts: 1. Space Target Detection and Recognition, 2. Space Target Component Segmentation.

**Space target detection and recognition.** In computer vision research, deep learning-based object detection models can simultaneously provide the bounding boxes and categories of objects in an image. Because each image in our dataset contains only one space target, a deep learning-based object detection algorithm can simultaneously perform both detection and recognition tasks for an image in the NCSTP dataset. We selected 10 object detection methods for experiments on detection and recognition: Faster R-CNN<sup>17</sup>, YOLOv3<sup>18</sup>, Centernet<sup>19</sup>, DETR<sup>20</sup>, Sparse R-CNN<sup>21</sup>, YoloF<sup>22</sup>, deformable DETR<sup>23</sup>, YoloX<sup>24</sup>, VitDet<sup>25</sup>, and DiffusionDet<sup>26</sup>. These algorithms incorporate various methods, including two-stage and one-stage methods based on CNNs, anchor-free methods, transformer-based approaches, and the latest methods based on diffusion models. These models were fine-tuned for 12 epochs based on pre-trained weights. The performance of each method was evaluated after the final epoch. ResNet50 was used as the backbone of all methods except for YOLOv3, which uses Darknet53; YOLOX, which



**Fig. 6** The quantity and size distribution of generated data according to target category.



**Fig. 7** The distribution of pixel counts and proportions for each component.

uses CSPDarknet; and ViTDet, which uses ViT-base. All methods were trained and tested on a server equipped with an NVIDIA RTX 4090 GPU, to ensure fairness in the comparative experiments.

**Evaluation metrics.** We quantitatively evaluated the performance of the models using accuracy, model size, and computational complexity. For accuracy, we used mean average precision (mAP) as the standard metric to evaluate the model’s performance. mAP is the average of the average precision (AP) across different categories, where AP is the area under the precision-recall curve after interpolation. Calculating mAP requires precision and recall values. Precision is the ability of a model to identify only relevant objects. It is the percentage of correct positive predictions. Recall is the ability of a model to find all ground-truth bounding boxes. It is the percentage of correct positive predictions among all given ground truths. The formulas for calculating precision and recall are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

where TP represents true positives (correctly detected objects), FP represents false positives (incorrectly detected objects), and FN represents false negatives (undetected objects). The above formulas require us to predefine what differentiates a correct detection from an incorrect detection. A common way to do so is using the intersection over union (IoU). In the object detection scope, the IoU measures the overlapping area of the predicted bounding box  $B_p$  and the ground-truth bounding box  $B_g$  divided by the area of their union; that is,



Method	mAP	Flops(G)	Params(M)	backbone	mAP_small	mAP_medium	mAP_large
Faster R-CNN	87.0	208	41.4	Resnet50	75.8	85.4	91.5
YOLOv3	72.6	11.6	61.6	Darknet53	60.4	69.9	78.7
Centernet	80.8	12.1	32.1	Resnet50	65.2	77.5	87.3
DETR	78.3	96.5	41.6	Resnet50	49.6	76.6	86.7
Sparse R-CNN	93.7	152	106	Resnet50	84.3	92.6	97.3
YOLOF	87.3	99.0	42.6	Resnet50	77.1	85.8	91.0
Deformable DETR	78.0	193	40.1	Resnet50	47.9	77.3	84.4
YOLOX	84.5	13.3	8.94	CSPdarknet	72.3	82.4	89.2
ViTDet	84.7	279	101	ViT-base	70.9	83.0	89.5
DiffusionDet	88.5	105	111	Resnet50	75.0	86.6	93.3

**Table 3.** Comparison of benchmark evaluation results for target detection and recognition.

Method	CU	CY	GE	LE	NA	SA	SI	SY	SP	ST	DE	RO
Faster R-CNN	84.8	81.3	89.9	87.2	90.6	84.5	92.0	90.7	91.7	90.8	83.1	75.7
YOLOv3	66.3	66.9	73.3	72.0	78.8	74.0	77.9	68.9	71.8	80.9	73.7	68.1
Centernet	76.0	75.5	86.8	78.8	84.1	79.6	88.0	86.0	89.1	86.0	75.3	66.4
DETR	69.2	73.2	85.9	77.7	85.9	81.4	89.2	85.7	87.1	83.5	73.5	48.9
Sparse R-CNN	93.9	90.5	96.3	94.0	96.4	92.1	96.8	96.6	97.7	97.5	91.1	82.4
YOLOF	86.7	82.8	90.5	88.0	91.7	85.7	91.3	89.7	90.6	88.8	82.9	78.3
Deformable DETR	73.0	74.9	84.9	76.7	81.9	80.2	84.4	83.8	84.4	83.7	73.1	56.2
YOLOX	80.6	79.5	91.2	84.6	86.6	81.1	90.7	89.2	91.0	87.4	79.0	73.1
ViTDet	81.3	79.3	88.7	84.7	88.3	84.2	91.0	87.1	90.2	89.0	79.8	72.6
DiffusionDet	85.6	82.3	91.9	89.1	92.5	86.8	93.1	91.5	92.0	91.7	83.2	76.8

**Table 4.** Comparison of mAP for each class.

$$IOU = \frac{area(B_p \cap B_g)}{area(B_p \cup B_g)} \quad (5)$$

By comparing the IoU with a given threshold,  $t$ , we can classify a detection as correct or incorrect. If  $IOU \geq t$ , then the detection is considered correct. If  $IOU < t$ , the detection is incorrect.

After obtaining the precision and recall values, we can calculate the AP, which considers both metrics. Before this, we need to interpolate the P-R curve to smooth it and reduce the impact of curve fluctuations. Given a recall value  $r$ , the interpolated precision  $P_{interp}$  is the maximum precision value between the current recall value  $r$  and the next recall value  $r_n$ . We can calculate the AP value by averaging the precision values corresponding to all different recall points. This is equivalent to the area under the interpolated precision-recall curve and the X-axis, as follows:

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1}) \quad (6)$$

where

$$P_{interp}(r) = \max_{r_n \geq r} P(r_n) \quad (7)$$

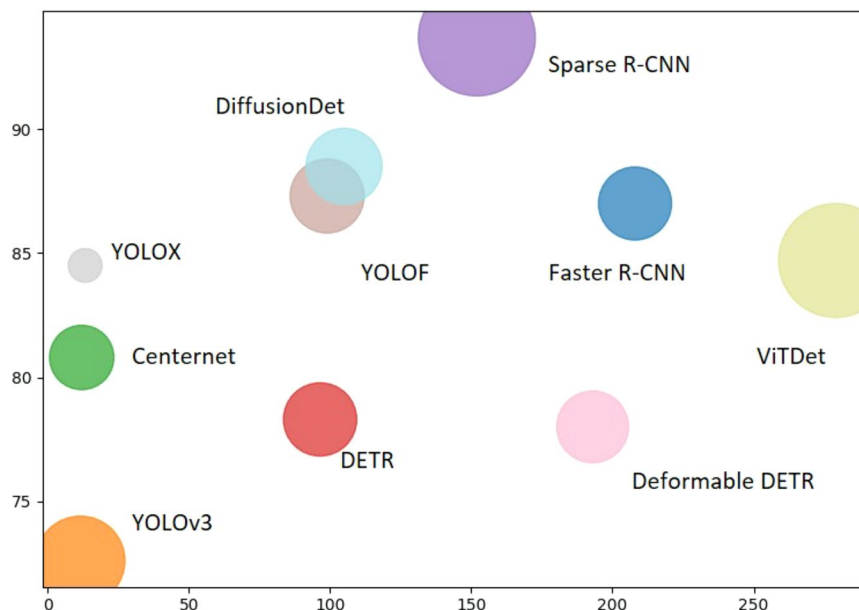
The mAP is simply the average AP over all classes; that is,

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (8)$$

where  $AP_i$  is the AP in the  $i$ th class, and  $N$  is the total number of classes in the dataset.

In addition, we calculated the FLOPs and the number of parameters (Params) for each model. FLOPs are commonly used to measure the computational complexity of a model, indicating how many floating-point operations are required during execution. Params is the total number of parameters that need to be trained in the model, reflecting the size of the model.

**Results and analysis.** Table 3 presents the overall performance of the 10 algorithms on the NCSTP dataset. Specifically, we provide the backbone, mAP, FLOPs, and Params of each algorithm. We set the IoU threshold  $t$  range from 0.5 to 0.95, with a step size of 0.05. The final mAP value was obtained by calculating the mAP at each



**Fig. 8** Object detection model performance scatter plot on NCSTP dataset.

threshold within this range and then averaging the results. Additionally, following the definition of COCO<sup>16</sup>, we calculated the mAP values for each method on large, medium, and small objects separately.

Table 4 shows the mAP values of the 10 models for each space object category. The abbreviations of the category names in the table correspond to their full names as follows: CubeSat (CU), cylindrical satellite (CY), geo-communication satellite (GE), LEO communication satellite (LE), space telescope (ST), SAR satellite (SA), navigation satellite (NA), symmetrical Earth observation satellites (SY), single-panel Earth observation satellites (SI), space debris (DE), and space rocks (RO).

Figure 8 shows the overall performance of each algorithm in spatial object detection and recognition on the NCSTP dataset. Each circle in the figure represents a model. The horizontal axis indicates the FLOPs of the model, with further rightward positions representing higher computational costs. The vertical axis represents mAP, with higher positions indicating better model accuracy. The radius of each circle corresponds to Params in the model, with larger radii indicating more parameters. At the per-class level, satellite subclasses are generally easier than debris and rocks. DETR variants show the steepest drop on rocks, while Sparse R-CNN remains robust.

While Sparse R-CNN yields the highest AP on NCSTP, it is relatively demanding for embedded deployment. For on-orbit scenarios, compute and power budgets are tight, so models with small footprints are preferable. YOLOv3 requires the fewest computational resources, but its mAP is lower than other algorithms, indicating lower accuracy. It also requires more storage space. YOLOX, however, has the smallest Params and a relatively lower FLOPs value, but its accuracy is worse than half of the other algorithms. Taking into account factors such as model accuracy, required storage space, and computational resources, YOLOF demonstrates a more balanced performance in on-orbit space target detection and recognition.

Overall, transformer-based algorithms typically have more parameters. Additionally, all methods experience a performance drop on small targets, indicating that further research is needed to improve the detection and recognition of distant spatial objects. We should consider the accuracy requirements and constraints of on-orbit resources for actual missions to select the most suitable method.

**Space target component segmentation.** This study transformed space target component recognition into a semantic segmentation problem. For each input image, we labelled the various components of the satellite in the form of masks and separated them from the background. Semantic segmentation aims to assign each pixel in an image to a predefined object class.

We selected 10 semantic segmentation methods for satellite component recognition experiments, namely FCN<sup>27</sup>, U-Net<sup>28</sup>, DeepLabV3+<sup>29</sup>, GCnet<sup>30</sup>, Fast-SCNN<sup>31</sup>, SETR<sup>32</sup>, Segformer<sup>33</sup>, K-net<sup>34</sup>, MaskFormer<sup>35</sup>, and Mask2Former<sup>36</sup>. These methods cover two types: CNN- and transformer-based approaches. All these models were fine-tuned for 160,000 items based on pre-trained weights. The performance of each method was evaluated after the final epoch. For the backbone of the methods, except for SETR, which uses Vit-Large; Segformer, which uses mit\_b0; and Fast-SCNN, which does not have a backbone—all other methods use ResNet50. As in space target detection and recognition experiments, all methods were trained and tested on a server with an NVIDIA RTX 4090 GPU.

**Evaluation metrics.** Regarding object detection and recognition tasks, we also evaluated the performance of these models on segmenting satellite components in the NCSTP dataset according to accuracy, model size, and computational complexity. For the model size and computational complexity, we still use Params and FLOPs as evaluation metrics. The accuracy of the semantic segmentation models was based on the mIOU.

Method	mIOU	mACC	Flops(G)	Params(M)	backbone	Body		Solar panel		Antenna		Observation load	
						Iou	Acc	Iou	Acc	Iou	Acc	Iou	Acc
FCN	77.7	83.2	58.0	47.1	Resnet50	69.52	72.75	90.04	94.19	75.73	81.29	75.57	84.72
Unet	80.3	85.8	203	29.0	Resnet50	71.83	75.59	91.24	96.64	80.0	87.14	78.21	83.73
DeepLabV3+	83.0	88.7	177	41.2	Resnet50	72.86	76.56	93.53	96.52	84.11	91.95	81.46	89.71
Gcnet	81.4	86.2	198	47.3	Resnet50	72.06	75.69	92.47	95.54	81.75	88.24	79.28	85.25
Fast-SCNN	43.90	56.0	0.936	1.40	—	41.37	74.49	45.94	46.48	38.15	40.41	46.44	62.46
SETR	71.2	79.7	367	309	ViT-Large	65.43	71.42	86.42	92.86	66.38	78.78	66.37	75.9
Segformer	79.9	86.4	7.90	3.72	Mit_b0	70.47	75.07	91.48	95.6	79.94	87.34	78.0	87.44
K-net	82.4	87.9	273	78.9	Resnet50	73.4	76.38	92.26	96.81	83.02	90.14	80.87	88.61
MaskFormer	84.8	89.5	181	45.0	Resnet50	74.5	77.33	93.44	96.61	86.69	93.48	84.48	90.72
Mask2Former	88.1	91.0	226	44.0	Resnet50	76.66	78.25	95.69	97.75	91.28	95.01	88.8	93.1

**Table 5.** Comparison of benchmark evaluation results for component segmentation.

In semantic segmentation, the IoU is the ratio of the intersection and union of the ground-truth labels and the predicted values for a specific class. The mIoU is the average of the IoU for each class in the dataset. Its calculation formula is as follows:

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

where  $P_{ij}$  represents the number of pixels of class  $i$  that are predicted as class  $j$ . In a confusion matrix for semantic segmentation,  $P_{ij}$  indicates how many pixels from the ground truth of class  $i$  are misclassified as class  $j$  by the model. The formula can also be written as follows:

$$mIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{TP}{FN + FP + TP} \quad (10)$$

where TP represents true positives, the number of correctly predicted pixels for a specific class; FN represents false negatives, the pixels of a specific class that were not predicted correctly; and FP represents false positives, the pixels incorrectly predicted as a specific class.

In addition to the mIOU, we also calculated another metric to measure model accuracy, mACC. Accuracy refers to the proportion of correctly predicted pixels to the total pixels. mACC is the average accuracy across all categories:

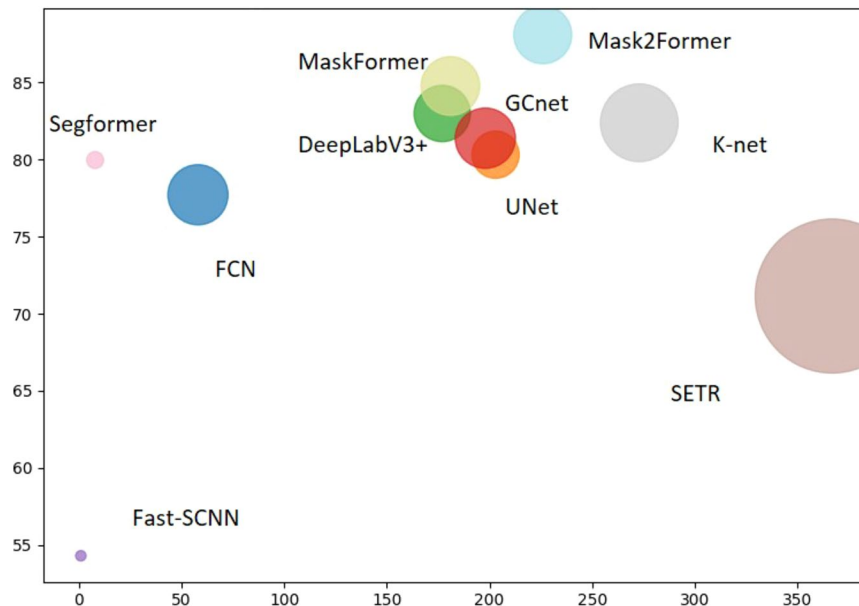
$$mACC = \frac{1}{k+1} \sum_{i=0}^k \frac{TP}{FN + TP} \quad (11)$$

Furthermore, the mACC only considers the ratio of true positives to false negatives, reflecting the model's accuracy. However, unlike the mIOU, it does not account for the impact of false positives. Compared with the mACC, the mIOU can more rigorously evaluate the performance of segmentation models, especially in cases of class imbalance, and better reflect the model's fine-grained prediction capabilities. Therefore, this paper primarily uses mIOU to measure the model's accuracy.

It is worth mentioning that in the images of the NCSTP dataset, aside from the pixels occupied by space targets, a significant portion is labeled as background with a pixel value of 0. Because our evaluation focuses solely on the segmentation accuracy of satellite components, the mIoU and mACC values we calculate exclude the background part (reduce zero label).

**Results and analysis.** Table 5 presents the performance of these 10 representative semantic segmentation algorithms on the NCSTP dataset. The table also presents the mIOU and mACC values for each model across all component categories and the IoU and accuracy values for the four components: body, solar panel, antenna, and observation load. In addition, we list the backbone used by each model, along with the number of parameters and FLOPs. Consistent with the object detection algorithms, to comprehensively compare the overall performance of the models, we have plotted the mIOU, Params, and FLOPs information of all models in Fig. 9, where the X-axis represents FLOPs, the Y-axis represents mIOU, and each colored circle represents a model. The radius of the circle indicates the size of the model's Params.

Among all the semantic segmentation models, Mask2Former<sup>36</sup> achieved the highest accuracy, with an mIOU of 88.1. Its Params was at a moderate level, but its FLOPs value was relatively high, indicating that this model's deployment and inference require considerable computational resources. In contrast, the Fast-SCNN model had very small Params and FLOPs, allowing it to be deployed and run under extremely limited storage and computational resources. However, its mIOU was also much lower than that of other models. Additionally, SegFormer and DeepLabv3+ exhibited high mIOU while maintaining relatively low Params and FLOPs, making their overall performance quite impressive.



**Fig. 9** Semantic segmentation model performance scatter plot on NCSTP dataset.

A detailed analysis of the models' performance for each component category revealed that all models achieved higher segmentation accuracy for solar panels than other components. This may be because solar panels typically have clear boundaries with other components, possess similar structures across different satellites, and generally occupy more pixel values than other types of components. In addition, the practical implication of this bias is that, in space operations, different components are of varying levels of interest. However, if a given space operation task places greater emphasis on antennas or payloads, it becomes necessary to design more targeted perception algorithms specifically tailored to those components.

The semantic segmentation algorithm can be chosen for tasks with different resource constraints. If higher accuracy is desired and resources are sufficient, Mask2Former<sup>36</sup> can be selected. However, if the model needs to be deployed on platforms with limited computational and storage resources, such as small satellites or space drones, Fast-SCNN would be a better choice. While this sacrifices accuracy, it meets the requirements for real-time tasks.

### Data availability

Our dataset<sup>15</sup> is publicly available at <https://doi.org/10.6084/m9.figshare.28606754>.

### Code availability

The code used in this research is available at <https://github.com/LYXLYXlyv/NCSTP>.

Received: 19 March 2025; Accepted: 30 September 2025;

Published online: 12 November 2025

### References

- Zhang, H., Zhang, Y., Feng, Q. & Zhang, K. Review of machine-learning approaches for object and component detection in space electro-optical satellites. *Int. J. Aeronaut. Space Sci.* **25**, 277–292, <https://doi.org/10.1007/s42405-023-00653-w> (2024).
- Wang, H. *et al.* A robust space target extraction algorithm based on standardized correlation space construction. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **17**, 10188–10202, <https://doi.org/10.1109/JSTARS.2024.3397462> (2024).
- Chen, H. L., Sun, Q. Y., Li, F. F. & Tang, Y. Computer vision tasks for intelligent aerospace perception: an overview. *Sci. China Technol. Sci.* **67**, 2727–2748, <https://doi.org/10.1007/s11431-024-2714-4> (2024).
- Hongshan, N. *et al.* Research on space-based space target detecting and tracking algorithm. In *2010 International Conference on Intelligent System Design and Engineering Application (ISDEA)* 253–256. IEEE. <https://doi.org/10.1109/ISDEA.2010.180> (2010).
- Zhao, P. Y., Liu, J. G. & Wu, C. C. Survey on research and development of on-orbit active debris removal methods. *Sci. China Technol. Sci.* **63**, 2188–2210, <https://doi.org/10.1007/s11431-020-1661-7> (2020).
- Lillie, C. On-orbit assembly and servicing for future space observatories. In *AIAA SPACE 2006 (Space 2006)*. <https://doi.org/10.2514/6.2006-7251> (2006).
- Guariniello, C. & DeLaurentis, D. A. Maintenance and recycling in space: functional dependency analysis of on-orbit servicing satellites team for modular spacecraft. In *AIAA SPACE 2013 Conference and Exposition (AIAA 2013-5327)*. AIAA. <https://doi.org/10.2514/6.2013-5327> (2013).
- Kisantal, M. *et al.* Satellite pose estimation challenge: dataset, competition design, and results. *IEEE Trans. Aerosp. Electron. Syst.* **56**, 4083–4098, <https://doi.org/10.1109/TAES.2020.2989063> (2020).
- Park, T. H., Märtens, M., Lecuyer, G., Izzo, D. & De Amico, S. SPEED+: next-generation dataset for spacecraft pose estimation across domain gap. In *2022 IEEE Aerospace Conference (AERO)* 1–15.
- Proença, P. F. & Gao, Y. Deep learning for spacecraft pose estimation from photorealistic rendering. In *2020 IEEE International Conference on Robotics and Automation* 6007–6013. <https://doi.org/10.1109/ICRA40945.2020.9197244> (2020).
- Dung, H. A., Chen, B. & Chin, T.-J. A spacecraft dataset for detection, segmentation and parts recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* 2012–2019. IEEE. <https://doi.org/10.1109/CVPRW53098.2021.00229> (2021).

12. Zhang, Z. P., Deng, C. W. & Deng, Z. Y. A diverse space target dataset with multidebris and realistic on-orbit environment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **15**, 9102–9114, <https://doi.org/10.1109/JSTARS.2022.3203042> (2022).
13. Zhang, C. Y. *et al.* STAR-24K: a public dataset for space common target detection. *KSI Trans. Internet Inf. Syst.* **16**, 365–380, <https://doi.org/10.3837/tiis.2022.02.001> (2022).
14. Zhao, Y. P., Zhong, R. & Cui, L. Y. Intelligent recognition of spacecraft components from photorealistic images based on Unreal Engine 4. *Adv. Space Res.* **71**, 3761–3774, <https://doi.org/10.1016/j.asr.2022.09.045> (2023).
15. Liu, Y. NCSTP: a large-scale dataset for non-cooperative space target perception. *figshare*. <https://doi.org/10.6084/m9.figshare.28606754> (2025).
16. Lin, T.-Y. *et al.* Microsoft COCO: common objects in context. In *Computer Vision – ECCV 2014* (eds Fleet, D. *et al.*) 740–755, Springer. [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48) (2014).
17. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 1137–1149, <https://doi.org/10.1109/TPAMI.2016.2577031> (2017).
18. Redmon, J. & Farhadi, A. YOLOv3: an incremental improvement. Preprint at <https://arxiv.org/abs/1804.02767> (2018).
19. Duan, K. W. *et al.* CenterNet: keypoint triplets for object detection. In ICCV 2019 6568–6577. *IEEE*. <https://doi.org/10.1109/ICCV.2019.00677> (2019).
20. Carion, N. *et al.* End-to-end object detection with transformers. In *ECCV 2020 213–229*. Springer. [https://doi.org/10.1007/978-3-030-58452-8\\_13](https://doi.org/10.1007/978-3-030-58452-8_13) (2020).
21. Sun, P. *et al.* Sparse R-CNN: end-to-end object detection with learnable proposals. In *CVPR 2021 14449–14458*. *IEEE*. <https://doi.org/10.1109/CVPR46437.2021.01422> (2021).
22. Chen, Q. *et al.* You only look one-level feature. In *CVPR 2021 13034–13043*. *IEEE*. <https://doi.org/10.1109/CVPR46437.2021.01284> (2021).
23. Zhu, X. *et al.* Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR 2021 894–909* (2021).
24. Ge, Z., Liu, S., Wang, F., Li, Z. & Sun, J. YOLOX: exceeding YOLO series in 2021. Preprint at <https://arxiv.org/abs/2107.08430> (2021).
25. Li, Y., Mao, H., Girshick, R. & He, K. Exploring plain Vision Transformer backbones for object detection. In *ECCV 2022 280–296*. Springer. [https://doi.org/10.1007/978-3-031-20086-1\\_17](https://doi.org/10.1007/978-3-031-20086-1_17) (2022).
26. Chen, S., Sun, P., Song, Y. & Luo, P. DiffusionDet: diffusion model for object detection. In ICCV 2023 19773–19786. *IEEE* (2023).
27. SShelhamer, E., Long, J. & Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 640–651, <https://doi.org/10.1109/TPAMI.2016.2572683> (2017).
28. Ronneberger, O. *et al.* U-Net: convolutional networks for biomedical image segmentation. In *MICCAI 2015 234–241*. Springer. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28) (2015).
29. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. Encoder–decoder with atrous separable convolution for semantic image segmentation. In *ECCV 2018 833–851*. Springer (2018).
30. Cao, Y., Xu, J., Lin, S., Wei, F. & Hu, H. GCNet: non-local networks meet squeeze-excitation networks and beyond. In *ICCV Workshops 1971–1980*. *IEEE* (2019).
31. Poudel, R. P. K., Liwicki, S. & Cipolla, R. Fast-SCNN: fast semantic segmentation network. In *BMVC 2019 289–301* (2019).
32. Zheng, S. *et al.* Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR 2021 6877–6886*. *IEEE* (2021).
33. Xie, E. *et al.* SegFormer: simple and efficient design for semantic segmentation with transformers. In *NeurIPS 2021 12077–12090* (2021).
34. Zhang, W., Pang, J., Chen, K. & Loy, C. C. K-Net: towards unified image segmentation. In *NeurIPS 2021 10326–10338* (2021).
35. Cheng, B., Schwing, A. G. & Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS 2021 17864–17875* (2021).
36. Cheng, B., Misra, I., Schwing, A. G., Kirillov, A. & Girdhar, R. Masked-attention Mask Transformer for universal image segmentation. In *CVPR 2022 1280–1289*. *IEEE* (2022).

## Acknowledgements

This research was funded by the Pre-Research Project on Civil Aerospace Technologies (D030312).

## Author contributions

Conceptualization, Y.L. and H.N.; methodology, Y.L.; validation, Y.L., H.N. and S.C.; formal analysis, Y.L.; investigation, Y.L.; resources, Y.L.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, N.H. and Z.Y.; visualization, Y.L.; supervision, S.C.; project administration, C.B.; funding acquisition, C.B.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to Y.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, 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 you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. 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-nc-nd/4.0/>.

© The Author(s) 2025