A Spacecraft Dataset for Detection, Segmentation and Parts Recognition

Hoang Anh Dung
The University of Adelaide
hoanganhdung@hdu.edu.vn

Bo Chen
The University of Adelaide
bo.chen@adelaide.edu.au

Tat-Jun Chin
The University of Adelaide
tat-jun.chin@adelaide.edu.au

Abstract

Virtually all aspects of modern life depend on space technology. Thanks to the great advancement of computer vision in general and deep learning-based techniques in particular, over the decades, the world witnessed the growing use of deep learning in solving problems for space applications, such as self-driving robot, tracers, insect-like robot on cosmos and health monitoring of spacecraft. These are just some prominent examples that has advanced space industry with the help of deep learning. However, the success of deep learning models requires a lot of training data in order to have decent performance, while on the other hand, there are very limited amount of publicly available space datasets for the training of deep learning models. Currently, there is no public datasets for space-based object detection or instance segmentation, partly because manually annotating object segmentation masks is very time consuming as they require pixel-level labelling, not to mention the challenge of obtaining images from space. In this paper, we aim to fill this gap by releasing a dataset for spacecraft detection, instance segmentation and part recognition. The main contribution of this work is the development of the dataset using images of space stations and satellites, with rich annotations including bounding boxes of spacecrafts and masks to the level of object parts, which are obtained with a mixture of automatic processes and manual efforts. We also provide evaluations with state-of-the-art methods in object detection and instance segmentation as a benchmark for the dataset. The link for downloading the proposed dataset can be found on https://github.com/Yurushia1998/SatelliteDataset.

1. Introduction

Space technologies play a vital role in many critical applications today: communication [1], navigation [2] and meteorology [3] are some prominent examples, thanks to the development of computer vision and machine learning techniques. Within the last two decades, there has been a wide range of machine learning-based applications deployed in the space industry, such as self-navigation system for collision avoidance [4], health monitoring of spacecrafts [5], and asteroid classifications [6], just to name a few. Accompanying the development of space technologies is an increase in demand of space datasets, as most of state-of-art models for space technologies are using deep learning-based methods, which require a significant amount of annotated data for supervised training in order to have good performance. However, one challenge that hinders the advancement of these space technologies is the lack of publicly available datasets, due to sensitivity in the space area and the high cost of obtaining space-borne images.

One important technology in many space applications is the accurate localisation of space objects via visual sensor, such as object detection and segmentation in images, because localisation is a key step towards vision-based pose estimation which is critical for tasks like docking [7], servicing [8], or debris removal [9]. However, a severe challenge for space-based object detection and instance segmentation is the lack of accessible large datasets that have been well annotated. There has been some large scale segmentation dataset such as COCO[10], ImageNet [11], Pascal VOC [12] including masks of a large number of classes for daily life objects and human parts, but there is not any specialized datasets segmenting space objects such as satellites, space station, spacecrafts or other Resident Space Objects (RSOs). The closest and the largest datasets related to this topic so far are the Spacecraft PosE Estimation Dataset (SPEED) [13] and the URSO dataset [14]. However, these datasets are focused on pose estimation and do not provide any segmentation annotations.

Since pixel-level mask are required as ground truth for training, building a segmentation dataset for any new domain, can be very time-consuming. For example, powerful interactive tools [15] are adopted for annotating the MS COCO dataset [10], but it still takes minutes for an experienced annotator labeling one image [10]. As the large amount of parameters in modern neural networks often require being trained on fairly large datasets in the scales from thousands to millions, the total amount of effort it takes to develop such a dataset remains dauntingly expensive.

To reduce the cost and manpower required for image mask annotation, there has been many researches trying to automate or semi-automate the annotation process using unsupervised approaches, such as interactive segmentation [16, 17] where human annotators use a model to create sample masks and interact with the samples iteratively to refine it, or weakly supervised annotation methods [18, 19] where users only need to provide a 'weak' annotation giving minimum information about the mask of the images. Another line of works trying to circumvent the expensive cost of annotation is to rid the need of annotation at all via self-supervised learning, such as [20, 21]. However, self-supervised learning based methods tend to be inferior in detection and segmentation tasks compared to their supervised counterparts.

Our contributions In this work, we aim to contribute to space-based vision researches by creating a new public available space images dataset, as the first step in a long term goal to develop new machine learning algorithms for spacecraft object detection, segmentation and part recognition tasks.

As spaces images are often considered as sensitive data, there are not many real satellite images are publicly available. To enrich our image dataset, we collect 3117 images of satellites and space stations from synthetic or real images and videos. We then use a bootstrap strategy in the annotation process to maximally reduce manual efforts required. We first adopt an interactive method for manual labelling at a small scale, then utilised the labelled data to train a segmentation model for automatically producing coarse labels for more images, which have subsequently gone through manual refinement via the interactive tool. As more finely annotated images are produced, this process repeats and scales up until we finally produces the whole dataset.

To provide a benchmark for our dataset we conduct experiments using state-of-the-art detection and segmentation methods. The performance of our dataset in comparison to popular datasets such as Cityscapces [22] and Pascal VOC [12] indicates that space-based semantic segmentation is a challenging task for models designed based on on-Earth scenarios and poses a open domain for future research.

2. Related works

Image segmentation is a topic that has been studied for a number of decades in the field of computer vision, which has recently regained significant attention due to the success of deep learning. Consequently, the demand for annotated datasets has grown rapidly. There has been various researches that focus on minimizing the cost of training segmentation models via either improving the data annotation techniques or reducing the reliance on labelled data in training. We briefly review notable techniques in data annotation and self-supervised learning.

Data annotation To minimise the amount of human input in image mask annotation, various techniques have been proposed. Maninis *et al.* [19] use extreme points of the objects that are to be segmented (points to the top, the bottom, the left-most and the right-most on the boundary of the object) as a annotation signal. Each extreme point is converted to a 2D Gaussian heatmap and concatenated to the input image as an extra channel of features. The model then learns to utilise this information to produce an accurate mask. Scribble-based techniques [18, 16] on the other hand, use a scribbled-based input as an annotation signal. Unlike extreme points or bounding boxes, scribbles does not give information about the object location, instead it provides information about color and intensity of the objects to be segmented. The model then propagates this object specific information from scribbles to other pixels and estimates the object masks. Other notable techniques in mask annotation include bounding box input [23], object centre input [24], polygon input [25, 26] and interactive approaches [27, 17].

Self-supervised learning This is a topic that has recently gained popular attention as it addresses the lack of training data problem in deep learning methods. There has also been efforts for image segmentation based on self-supervised learning. The main idea of this approach is to facilitate the model to learn information about inherent structures within images by training the model with the same input data with different representation or augmentation. In 2019, Larsson *et al.* [21] used a k-mean clustering to get predicted labels for pixels of 2 different images from the same scene with different weather condition. It uses a correspondent loss of the differences in segmentation of both images to learn meaningful features, as they should have the same labels. Another work [20] uses a self-supervised equivariant attention mechanism to provide additional supervision signal in semantic segmentation. In this work, instead of using images of same scenes at different weather condition, it applies affine transformations on input images and uses an equivariant cross regularization loss to encourage feature consistency in learning.

3. Building the dataset

In this section we describe our methodology for data collection and mask generation throughout the dataset development process. We collect a large number of synthetic and real images from the internet. We then use a bootstrap strategy to effectively reduce the amount of manual labor required for data annotation. We further perform a post-process step to remove similar or identical images, remove

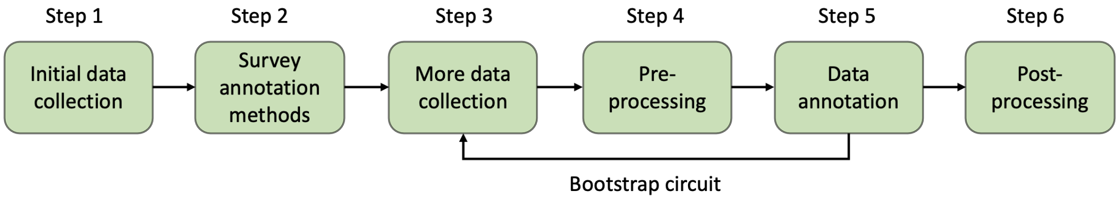

Figure 1: Process from data collection to image segmentation.

texts or refine low-quality annotations. Figure 1 illustrates the step-by-step process of our methodology.

3.1. Step 1: Initial data collection

For the dataset to be useful for training practical models, we need to collect a significant amount of data. However at this step only a small amount was needed since we would only use them to test viable annotation methods that are easy to operate and are able to produce satisfactory masks.

3.2. Step 2: Surveying annotation methods

Using the initial amount of data from previous step, we conduct experiments on current various state-of-the-art models and tools for image segmentation. Decomposing spacecrafts into parts requires specialised domain knowledge. For practical reasons, we opt to segment spacecrafts into 3 parts that are commonly observable and easily identified, namely solar panel, main body and antenna.

Among the surveyed methods, self-supervised [20, 21] and weakly supervised segmentation methods [18] have advantages of low human interaction and labor requirements per images. However, their performance are no where near satisfactory, as the satellites often have a lot of unorthodox and small parts. Another drawback of self-supervised or weakly supervised approaches is that it is highly inefficient to further refine their output predictions. On the other hand, interactive segmentation methods such as [19] have advantages of allowing users to improve the mask bit by bit with manual inputs, which are much more suitable for the purpose of this work. After testing various methods, we decided to use Polygon-RNN++ [27], an improved version of Polygon-RNN [26]. This model allows us to break down the object into small convex areas. We can then label each of these convex area with polygons manually based on their position on the spacecraft, and the mask will then be created. The model also allows users to freely modify the mask at pixel level by adding or removing key points. Figure 2 is an example image with masks of each part, labelled using Polygon-RNN++.

3.3. The bootstrap circuit

From step 3 to 5, we employ a bootstrap strategy to make the labelling process semi-automatic, via utilising already annotated images to train a segmentation model to generate initial mask predictions. We first collect more images to expand our data base. A pre-processing step was then conducted to remove similar or duplicate images. Lastly we train segmentation models to predict the initial masks and refine them using Polygon-RNN++ as described in step 2. This bootstrap circuit is repeated and as the training data grows, the segmentation models also improve, which in turn further lowers the cost of labelling more data.

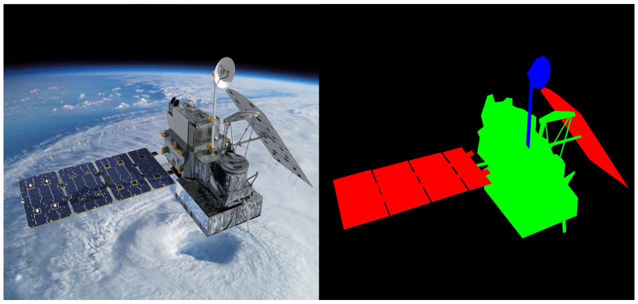

Figure 2: Example of an image collected and its annotated masks. Red mask: solar panel; blue mask: antenna; green mask: main body.

Pre-processing In order to remove duplicate or near-duplicate images due to difference in size, resolution and augmentation, we used Agglomerative Clustering [28] implemented in sklearn [29], combined with a simple searching algorithm. For each image I_i, we create a feature vector $\boldsymbol{f}^{(i)} = [f_1^{(i)}, ..., f_M^{(i)}]^T \in \mathbb{R}^M$ using the color histogram of the image. We then rrun the clustering algorithm based on the chi-square distance

$$d(I_i, I_j) = \frac{1}{2}\sum_{k=1}^{M} \frac{(f_k^{(i)} - f_k^{(j)})^2}{f_k^{(i)} + f_k^{(j)}} \tag{1}$$

between two images I_i and I_j.

After the clustering algorithm groups similar images into clusters, we use a searching algorithm to find the top n couples of images with highest similarity based on the chi-square distance, and manually remove those that are nearly the same.

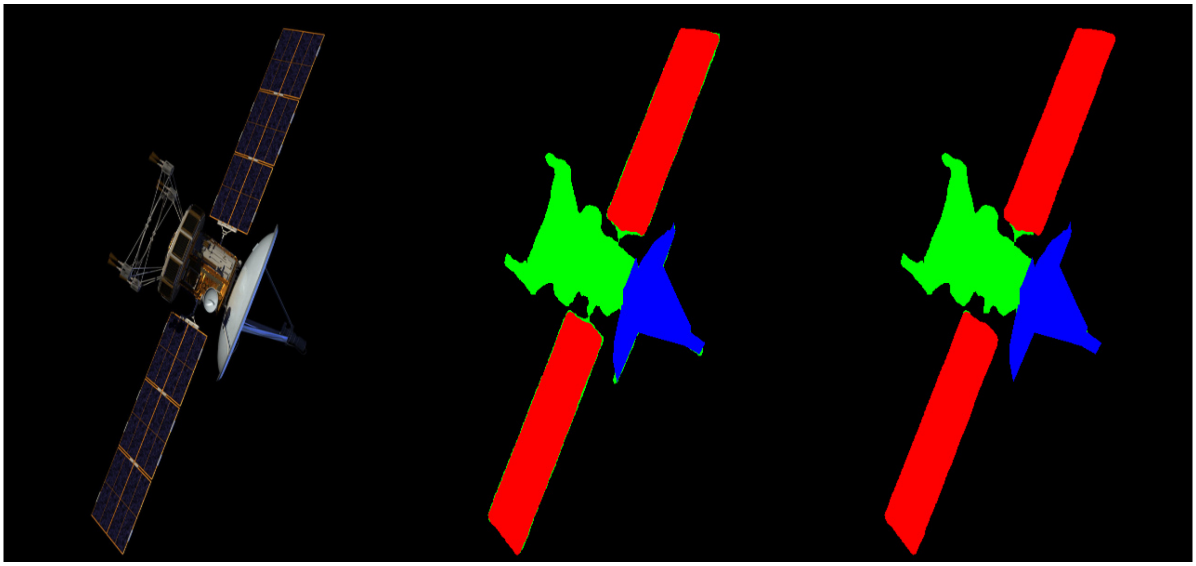

Figure 3: Comparing masks before and after manual refinement. Left: input image. Centre: assembled masks from model predictions. Right: masks after manual refinement. The models here are trained on 1003 annotated images.

Data Annotation We label the initial batch of images manually as described in step 2. For all further iterations, we leverage existing annotations to support the labelling process by training state-of-the-art models for producing initial mask predictions for different parts.

We use the DeepLabV3 [30] architecture and refined initial weights pretrained on ImageNet [11] on our dataset. We train 3 different models to predict: full mask of spacecrafts, mask of the solar panels, and mask of antennae. For each image, we then adopt good predictions of the parts, assembled them, and manually refine the final mask. Figure 3 compares the predicted masks and the refined masks of an example image.

As we accumulate more annotated images, the trained models were further refined with the latest dataset to improve future predictions, which in turn, reduces manual efforts and speeds up the labelling of new images.

3.4. Step 6: Post-processing

After we got a sufficient number of masks from the bootstrap circuit, we start going back to problematic images that are marked as requiring further processing. There are images that has text need to be removed, similar images that failed to be filtered in step 4, images that are deemed to be too difficult to identify even with human vision, etc. We also go back to re-mask those that we deem low in quality. Once we obtain the mask labels, we compute tight bounding boxes of the spacecrafts for each image.

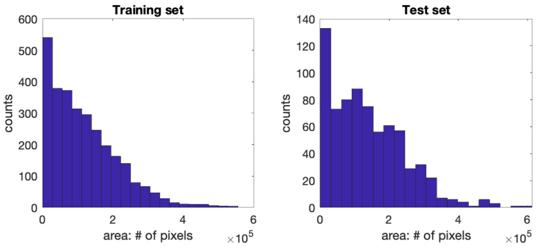

Figure 4: Histograms of the spacecraft mask areas in the training and test set.

4. Dataset statistic

The final dataset consists of 3117 images with uniform resolutions of 1280×720. It includes masks of 10350 parts of 3667 spacecrafts. The spacecraft objects are also various in range, they can be as small as 100 pixels to as large as occupying nearly the whole images. On average, each spacecraft takes up an area of 122318.68 pixels, while each part of antenna, solar panel and main body occupies areas of 22853.64, 75070.76 and 75090.92 pixels, respectively. For the purpose of standardising benchmarking segmentation methods, we divide our dataset into a training and a test subsets, which consists of 2516 and 600 images respectively. Figure 4 provides the distribution of spacecraft sizes in the training and test sets.

5. Experiments

In this section we conduct various experiments to benchmark our dataset with state-of-the-art models in tasks in-

Model	mAP	AP50
YoloV3	0.700	0.852
YoloV3-spp	0.736	0.868
YoloV4-pacsp	0.807	0.896
YoloV4-pacsp-x	0.788	0.896
EfficientDet 7D	**0.880**	**0.904**
EfficientDet 7DX	0.707	0.902

Table 1: Mean AP50:95 and AP50 of different models on datasets

cluding object detection, instance segmentation and semantic segmentation.

5.1. Spacecraft object detection

To serve as the benchmark for spacecraft object detection, we train object detection models such as various versions of YOLO [31, 32] or EfficientDet [33]. We use the same training and evaluation settings as in their original codes, except that we change the training input size of YOLO models to 1280x1280, while testing size remains the same (640x640). We initialise YOLO models with ImageNet pretrained weights, and COCO pretrained weights for EfficientDet models. All models were trained for 20 epochs (around 50000 iterations for EfficientDet). For evaluation metrics, we use meanAP [10] and meanAP50 [10]. As shown in Table 1, our experiments suggest that EfficientDet has better detection performance than the YOLO variants in the space-based scenario.

5.2. Spacecraft instance segmentation

To benchmark our dataset with state-of-the-art models for spacecraft instance segmentation, We use a variety of segmentation models to test their performances on our dataset, including HRNet [34], OCRNet [35], OCNet [36], ResneSt [37] and DeepLabV3+ Xception [38]. We maintain as much as possible the original training settings as in the respective papers, with some small dataset or hardware specific adaptations. All models are trained with around 40 epochs (For HRNet, AspOCNet and OCRNet, we train them with 13000-15000 iterations). The training batch sizes for DeepLabV3+ Xception were 8 while the rest of models had batch size 4. We use pixel accuracy (PixAcc) [39] and mean intersection-of-union (mIoU) [40] to compare model performances.

For ResneSt models, we use 3 different backbone with DeepLabV3, including ResneSt101, ResneSt200 and ResneSt269, with DeepLabV3 and an extra auxillary header as segmentation head, so the loss is a weighted combination of losses between DeepLabV3 header output and auxillary header output. All input training images are cropped to size 480 × 480, except for ResneSt269 which had input size 420 × 420.

Model	PicAcc	mIoU	mIoU (No BG)
DeepLabV3+Xception	0.965	0.78	0.714
ASPOCNET	0.972	0.803	0.744
OCRNet	0.972	0.802	0.742
HRNetV2+OCR+	0.974	0.797	0.735
ResneSt101	0.977	0.822	0.767
ResneSt200	**0.978**	**0.838**	**0.79**
ResneSt269	0.977	0.835	0.786

Table 2: Performances of different state-of-the-art models for whole spacecraft segmentation.

Model	Body	Solar panel	Antena
DeepLabV3+ xception	0.767	0.802	0.575
ASPOCNET	0.800	0.842	0.588
HRNetV2+ OCR+	0.814	0.856	0.533
OCRNet	0.803	0.839	0.585
ResneSt101	0.834	0.868	0.600
ResneSt200	**0.842**	**0.878**	0.640
ResneSt269	0.830	0.870	**0.65**

Table 3: mIoU by spacecraft parts in different models

For DeepLabV3+ with Xception, we use Resnet101 as backbone for the model and crop the input image to size 513×513. Similar to ResneSt models, we use full images instead of a crop for testing.

For HRNet, we use model HRNet48W OCR with pretrained HRNet48W as backbone. Similar to ResneSt models, we also use an extra auxillary head and weighted auxillary loss in this model. All original images and masks are resized to 1024×512 for training and 2048×1024 for validation, similar to how HRNet processes Cityscape Images. On the other hand, OCNet and OCRNet use Imagenet pretrained weights Resnet101 as backbone. All models use SGD as optimizer with weight decay.

Table 2 reports the segmentation results across different methods on our dataset. Because a significant part of the image is background, which does not contribute much meaningful information and can affect the result of model evaluations, we represent two mIoU results of including and excluding the background class.

5.3. Spacecraft parts segmentation

Table 3 shows the performance of state-of-the-art models segmenting spacecraft parts on our dataset. The body and the solar panel of the spacecraft have been decently segmented as reflected by mIoU. The performance for antenna on the other hand, is fairly poor since they are quite unorthodox in shape and difficult to identify. Also, it is noticeable that performance on solar panel is higher than the other parts. This is because solar panels are oftentimes clearly

Model	Pascal VOC	City.	City. val	Ours
DeepLabV3+ Xception	0.890	0.821	0.827	0.714
ASPOCNET	-	0.817	-	**0.744**
OCRNet	0.843	0.824	0.806	0.742
HRNetV2+OCR+	0.845	**0.845**	0.811	0.735
ResneSt101	-	0.804	-	0.767
ResneSt200	-	0.833	0.827	0.790
Average	0.859	0.824	0.818	0.749

Table 4: mIoU of state-of-the-art models across different datasets, treating spacecraft parts as classes of objects.

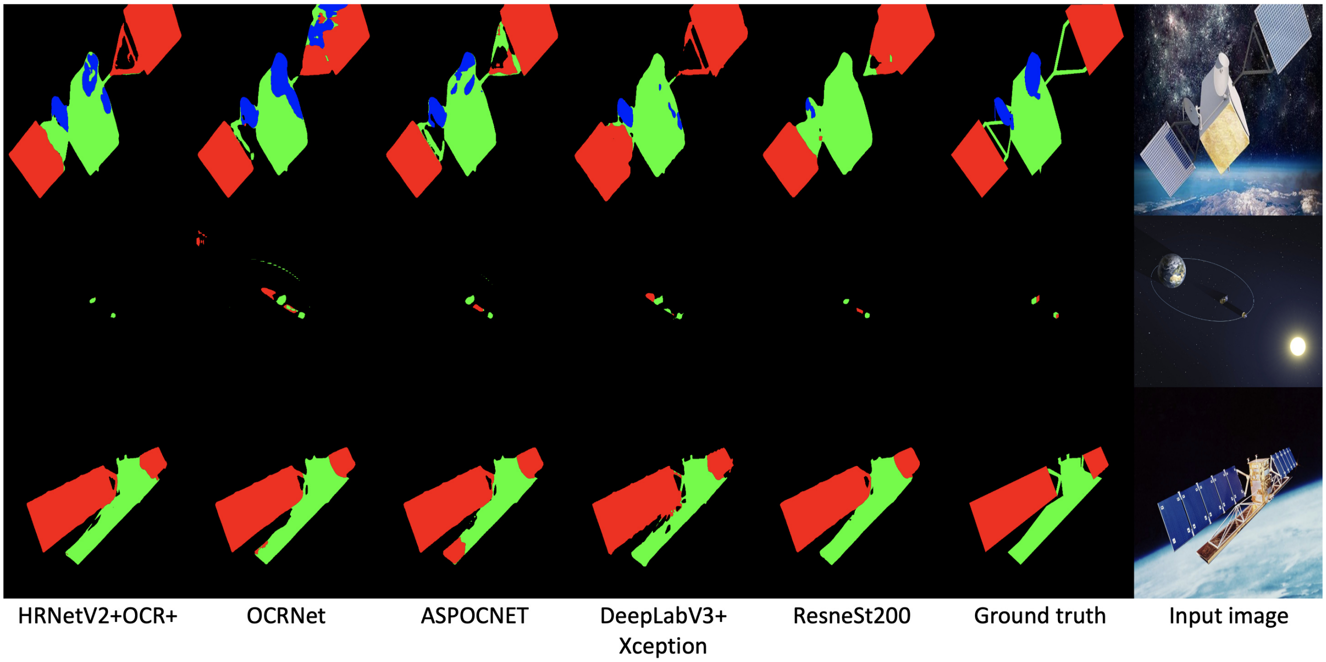

Figure 5: Predicted parts masks from different models.

separated from the other two parts while antenna and main bodies are in many cases embedded with each other.

Because our dataset is the first publicly available space dataset for spacecraft segmentation, it targets different type of objects in a unique scenario when compared to popular segmentation datasets such as Cityscapces [22] or Pascal VOC [12]. Nonetheless, for the sake of benchmarking the performance of space-based semantic segmentation against other on-Earth scenarios, we compare performances of state-of-the-art models across different datasets, by treating spacecraft parts as classes of objects for our dataset.

Table 4 reports the result of semantic segmentation on 4 datasets: Pascal VOC, Cityscapes, Cityscapes Val and our dataset. Overall, the average mIoU of Earth-based datasets are 6% to 11% higher compared to that of our dataset. It appears that current state-of-the-art models have inferior performance when deployed directly to identify and segment spacecraft parts.

In Figure 5 we provide a few qualitative results of our dataset from models in Table 4. As we can see, the complex structures in spacecrafts can result in precarious predictions of the masks of parts. Additionally, all models struggle to well identify the antenna of the spacecraft in the first row of Figure 5, which complies with the mIoU scores in Table 3.

Overall, the task of space-based semantic segmentation might not be directly solvable by models designed for on-Earth scenarios. Our dataset thus serves to address this gap by bringing novel challenges in this task.

6. Conclusion

We propose a space image dataset for vision-based spacecraft detection and segmentation tasks. Our dataset consists of 3117 space-based images of satellites and space stations, with annotations of object bounding boxes, instance and parts masks. We use a bootstrap strategy during dataset development to reduce manual labors. We conduct experiments in object detection, instance and semantic segmentation using state-of-the-art methods and benchmark our dataset for future space-based vision research.

Bibliography

[1] A. Vanelli-Corali, G. E. Corazza, G. K. Karagiannidis, P. T. Mathiopoulos, D. S. Michalopoulos, C. Mosquera, S. Papaharalabos, and S. Scalise. Satellite communications: Research trends and open issues. In *2007 International Workshop on Satellite and Space Communications*, pages 71–75, 2007. 1

[2] P. Enge. Satellite navigation: Present and future. *URSI Radio Science Bulletin*, 2012(341):5–9, 2012. 1

[3] Chris Kidd, Vincenzo Levizzani, and Peter Bauer. A review of satellite meteorology and climatology at the start of the twenty-first century. *Progress in Physical Geography: Earth and Environment*, 33(4):474–489, 2009. 1

[4] Thomas Uriot, D. Izzo, L. Simoes, R. Abay, Nils Einecke, Sven Rebhan, Jose Martinez-Heras, F. Letizia, J. Siminski, and K. Merz. Spacecraft collision avoidance challenge: design and results of a machine learning competition. *ArXiv*, abs/2008.03069, 2020. 1

[5] T. Yairi, N. Takeishi, T. Oda, Y. Nakajima, N. Nishimura, and N. Takata. A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction. *IEEE Transactions on Aerospace and Electronic Systems*, 53(3):1384–1401, 2017. 1

[6] V Carruba, S Aljbaae, R C Domingos, A Lucchini, and P Furlaneto. Machine learning classification of new asteroid families members. *Monthly Notices of the Royal Astronomical Society*, 496(1):540–549, 05 2020. 1

[7] T. Phisannupawong, P. Kamsing, P. Tortceka, and S. Yooyen. Vision-based attitude estimation for spacecraft docking operation through deep learning algorithm. In *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, pages 280–284, 2020. 1

[8] Benjamin B Reed, Robert C Smith, Bo J Naasz, Joseph F Pellegrino, and Charles E Bacon. The restore-l servicing mission. In *Proceedings of the AIAA space conference*, page 5478, 2016. 1

[9] Jason L Forshaw, Guglielmo S Aglietti, Nimal Navarathinam, Haval Kadhem, Thierry Salmon, Aurélien Pisseloup, Eric Joffre, Thomas Chabot, Ingo Retat, Robert Axthelm, et al. Removedebris: An in-orbit active debris removal demonstration mission. *Acta Astronautica*, 127:448–463, 2016. 1

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft coco: Common objects in context. *ECCV*, 2014. 1, 5

[11] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1, 4

[12] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010. 1, 2, 6

[13] M. Kisantal, S. Sharma, T. H. Park, D. Izzo, M. Märtens, and S. D'Amico. Satellite pose estimation challenge: Dataset, competition design, and results. *IEEE Transactions on Aerospace and Electronic Systems*, 56(5):4083–4098, 2020. 1

[14] Pedro F. Proença and Y. Gao. Deep learning for spacecraft pose estimation from photorealistic rendering. *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6007–6013, 2020. 1

[15] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Opensurfaces: A richly annotated catalog of surface appearancen. *SIGGRAPH*, 2013. 1

[16] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[17] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. *CVPR*, 2019. 2

[18] Di Lin, Jifeng Dai, Jiaya Jia1, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[19] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. *CVPR*, 2018. 2, 3

[20] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention

mechanism for weakly supervised semantic segmentation. *CVPR*, 2020. 2, 3

[21] Mans Larsson, Erik Stenborg, Carl Toft, Lars Hammarstrand, Torsten Sattler, and Fredrik Kahl. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. *ICCV*, 2019. 2, 3

[22] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 6

[23] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. *ICLR*, 2015. 2

[24] Amy Bearman1, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. *ECCV*, 2016. 2

[25] Bharath Hariharan, Pablo Arbeláez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. *ICCV*, 2011. 2

[26] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, pages 5230–5238, 2017. 2, 3

[27] David Acuna, Huan Ling, Amlan Kar, and S. Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018. 2, 3

[28] William HE Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24, 1984. 3

[29] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013. 3

[30] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 4

[31] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018. 5

[32] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020. 5

[33] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5

[34] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 5

[35] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 173–190, Cham, 2020. Springer International Publishing. 5

[36] Y. Yuan and Jingdong Wang. Ocnet: Object context network for scene parsing. *ArXiv*, abs/1809.00916, 2018. 5

[37] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi-Li Zhang, Haibin Lin, Yu e Sun, Tong He, Jonas Mueller, R. Manmatha, M. Li, and Alex Smola. Resnest: Split-attention networks. *ArXiv*, abs/2004.08955, 2020. 5

[38] Liang-Chieh Chen, Y. Zhu, G. Papandreou, Florian Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *ArXiv*, abs/1802.02611, 2018. 5

[39] Irem Ulku and Erdem Akagunduz. A survey on deep learning-based architectures for semantic segmentation on 2d images, 2020. 5

[40] R. Padilla, S. L. Netto, and E. A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020. 5