Article

Intelligent Space Object Detect Driven by Space Object Data

Qiang Tang , Xiangwei Li * , Meilin Xie , Jialiang Zhen

Posted Date: 9 November 2023

doi: 10.20944/preprints202311.0629.v1

Keywords: space object identification; deep learning; YOLO; deformable convolution

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 November 2023 doi:10.20944/preprints202311.0629.v1

Article

Intelligent Space Object Detect Driven by Space Object Data

Qiang T [1,2], Xiangwei li [1,*], Meilin Xie [1,2] and Jialiang Zhen [1,2]

[1] Xi'an Institute of Optics and Precision Mechanics of CAS, Xi'an 710000, China; xiemeilin6@163.com (M.X.)
[2] University of Chinese Academy of Sciences; Beijing 100049, China; tangqiang22@mails.ucas.ac.cn (Q.T.); zhenjialiang22@mails.ucas.ac.cn (J.Z.)
* Correspondence: xwli@opt.ac.cn (X.L.)

Abstract: With the rapid development of space programs in various countries, the number of satellites in space is increasing, resulting in an increasingly complex space environment. Therefore, improving space object identification technology has become highly important. We proposes a method of applying deep learning to intelligent detection of space object. We utilize 49 authentic 3D satellite models including 16 scenarios to generate a dataset comprising 17,942 images, which contains over 500 actual satellite photos. Additionally, we acquired a substantial amount of annotated data using a semi-automatic labeling method, which resulted in significant labor cost savings, and obtained a total of 39,000 labels. We validate the feasibility of the dataset using YOLOv3 and YOLOv7 models. What's more, we optimize the YOLOv7 model by integrating deformable convolution RepPoint into the YOLOv7 backbone to obtain the YOLOv7-R model. Through training with these two models, experimental results show that YOLOv3 achieves an accuracy of 0.927, YOLOv7 reaches an accuracy of 0.964, and YOLOv7-R achieves the highest accuracy at 0.983. This provides an effective solution for intelligent space object detection.

Keywords: space object identification; deep learning; YOLO; deformable convolution

0. Introduction

With the continuous advancement of space technology, technologies such as positioning and navigation, satellite communication, remote sensing observation and surveillance and reconnaissance, which use satellites as infrastructure, have developed significantly and played more and more important role in both civil and national defense fields. However, the accumulation of space debris and decommissioned satellites have brought about a severe threat to operational satellites. Therefore, the ability of satellites to automatically identify targets in the surrounding environment has become an urgent need to minimize the safety risk of in-orbit satellites [1]. In recent years, image sensors have been widely applied on satellites to enhance the satellite's ability to perceive the surrounding environment, which has significant implications and application value for image-based space object detection methods [2,3].

With the accumulation of massive data and the continuous development of artificial intelligence technology, deep learning has been widely used in rapidly improving the performance of object detection methods. In the field of generalized object detection, Alex et al. firstly proposed deep convolutional neural network [4]. Subsequently, researchers like Alex et al. combined deep neural networks with image object detection, resulting in a series of effective works, progressing from R-CNN to Fast R-CNN, and further to Faster R-CNN [5,6,7]. These works laid the foundation for classical architectures in object detection. However, due to the limitations of the R-CNN architecture, its detection speed cannot meet the demand for real-time detection of certain scenes. To address this, Redmon et al. introduced a regression-based object detection algorithm called YOLO (You Only Look Once) [8]. This method significantly improved detection speed while maintaining a certain level of accuracy. Liu et al. combined the regression idea of YOLO with the anchor box mechanism of Faster

R-CNN, resulting in the SSD (Single Shot MultiBox Detector) algorithm [9]. This algorithm maintained the high-speed performance of YOLO while ensuring precise bounding box localization effects. Redmon and his team continued to refine the YOLO algorithm, leading to the development of YOLO9000 [10], YOLOv3, and subsequently YOLOv7 frameworks [11,12,13,14].

In this work, we take the deep neural network YOLOv7 and its improved version YOLOv7-D, which is an improvement of the backbone network of YOLOv7, and validate the dataset as well as the model's validity. We designed a space object detection network and used the method of model combination to fuse the detection results of space object categories and the detection results of space object load states, and finally realized the automatic perception of space object image information [15,16,17]. The main contributions of this paper are summarized as follows:

1. We create a dataset comprising 17,942 images, including 500 real satellite images, using 49 real satellite 3D models.

2. We select 8282 images, and extract canny edge combining with original data to enhance the edge features. We also simulate satellite conditions in different backgrounds, such as deep space and Earth, and merge color and grayscale images with satellite imagery to simulate various backgrounds. Gaussian noise is added to mimic device degradation and space noise, maximizing the simulation of realistic space capture effects.

3. For dataset labeling, we use a combination of automatic labeling and manual selection, which greatly reduce the manual labeling costs.

4. We train the dataset using YOLOv3, YOLOv7, and YOLOv7-R models. The accuracy of YOLOv3 reaches 0.927, YOLOv7 achieves a training accuracy of 0.964, and YOLOv7-R attains an accuracy of 0.983.

1. Dataset Construction and Labeling

1.1. Dataset Construction

Obtaining such data is challenging because of the limitations imposed by observation conditions and the external environment. However, for space objects such as artificial objects, the load structure and characteristics are known to the observer. Through computer simulation, a large number of synthetic images with labeled information can be generated under given observation conditions. These synthetic images can be combined with a small portion of real data, using transfer learning methods to achieve detection of actual space objects [18].

In our work, we select 49 3D satellite models to produce a dataset with comprehensive orientation. The initial dataset consists of over 9,600 images, including 2,300 images from the BUAA-SID 1.0 dataset (640*480 resolution) which has 20 satellite models [19], 6,200 images (640*480 resolution) with 29 satellite models, and 500 actual satellite images. An additional 500 images were collected online. Due to the discrepancy in image size between the actual satellite dataset and the online collected dataset, they were uniformly resized to 600*600, and ultimately constitute a large-scale satellite dataset with two different sizes. The BUAA-SID 1.0 space object image database is based on space object 3D models, using 3ds Max software to produce a full-viewpoint simulation image sequence. It comprises 56 3D satellite models and their simulation images. The BUAA-SID 1.0 simulated image database contains 25,760 simulated images with the resolution of 320*240 pixels, with 460 images per satellite, of which 230 are full-view 24-bit color images, and the other 230 are the corresponding binary images. We selected 20 full-view 24-bit color satellite simulation images from BUAA-SID 1.0 dataset, choosing 115 images of various postures for each satellite. Using bilinear interpolation, the images were enlarged to 640*480 pixels. Some satellite images are displayed in Figure 1.

doi:10.20944/preprints202311.0629.v1

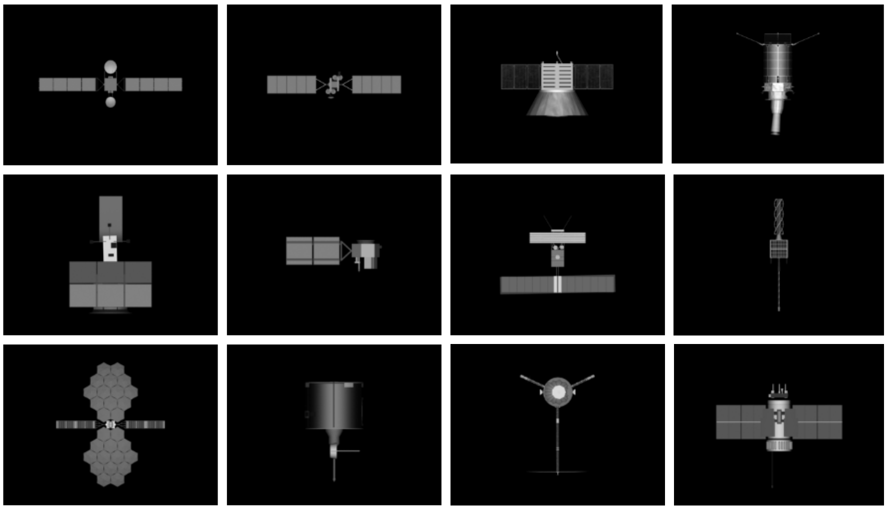


Figure 1. Partial satellite images from the BUAA-SID1.0

In this paper, the steps for generating a space object image by simulation are as follows:

Due to the limited amount of data in the BUSSA-SID1.0 dataset, it is difficult to meet the requirements of deep learning for large datasets. Therefore, we mimic the construction method of the BUAA-SID1.0 dataset and use the software Blender to model 29 different types of space objects. In order to authentically reflect the dynamic characteristics of these space objects, we simulate their motion by rotating and moving these models in the Blender software. For each type of space object, simulated observation scene images can be obtained based on their different postures and different relative positions to the Sun and the telescope. Each type of space object can generate 210 images. These images, after rendering using 3D modeling software, result in a total of 6200 high-resolution spatial object images. Some of these images are shown in Figure 2.

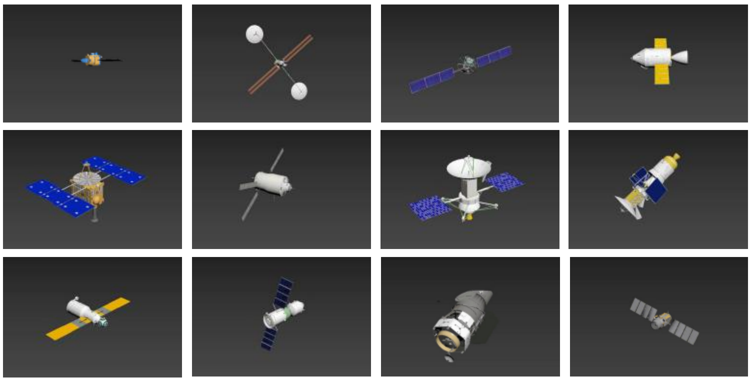


Figure 2. Partial dataset produced for this paper

In addition, a small amount of real data obtained in this paper (which has been authorized for use by the original authors) and the simulation dataset collected by the authors of this paper on the website are shown in Figure 3

doi:10.20944/preprints202311.0629.v1

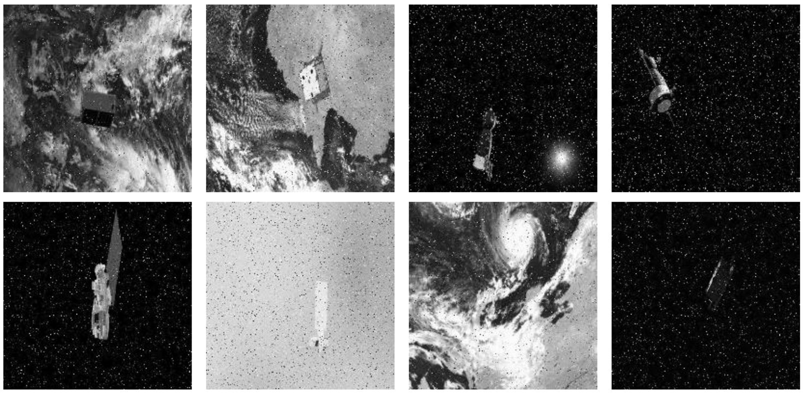


Figure 3. Selected real dataset

1.2. Data Enhancement

In this study, 8,282 simulated images from 49 satellites were selected for real scene simulation. Firstly, the color images were converted to grayscale images, and Gaussian filtering was applied to remove noise in the images [20]. Secondly, we calculated the gradient magnitude and direction of the images by Sobel operator [21], and non-maximum suppression was performed to eliminate non-edge points and refine the edges. The upper and lower threshold values for binarization were set to 30 and 150, respectively, to remove redundant pixels in the images, resulting in a sharpened edge image. The binary image was then combined with the original image in a 0.2:0.8 ratio to obtain a feature image with strong edge characteristics. Sixteen background images were selected, including color images of the Earth and deep space backgrounds, as well as grayscale images of the Earth and deep space backgrounds. To evenly distribute the backgrounds to the satellites and better simulate real world, we randomly divided the dataset to 16 groups of about 500 images each, and each group was fused with one kind of background image [22]. During the fusion process, an appropriate proportion of the background, ranging from 0.17 to 0.25, was chosen based on the specific background. Gaussian noise was added to this part to simulate various natural and artificial noise sources, such as thermal noise in electronic devices and celestial noise in astronomical observations. The reason why we chose Gaussian noise was that the distribution of random noise introduced by various factors in the process of image acquisition, transmission, and processing follows a Gaussian distribution [23]. After multiple experimental comparisons with the goal of minimizing image distortion, Gaussian noise with a mean of 0 and a variance of 0.001 was ultimately added to this part of the dataset. The image fusion process is illustrated in Figure 4, and the finally results are shown in Figure 5.

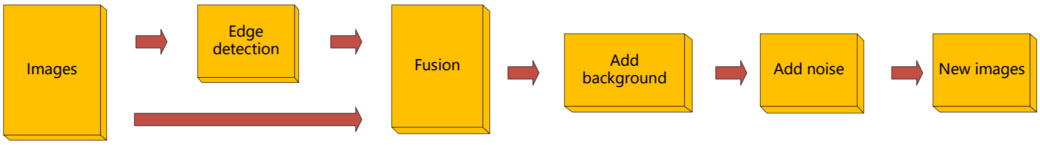


Figure 4. Image enhancement flowchart

doi:10.20944/preprints202311.0629.v1

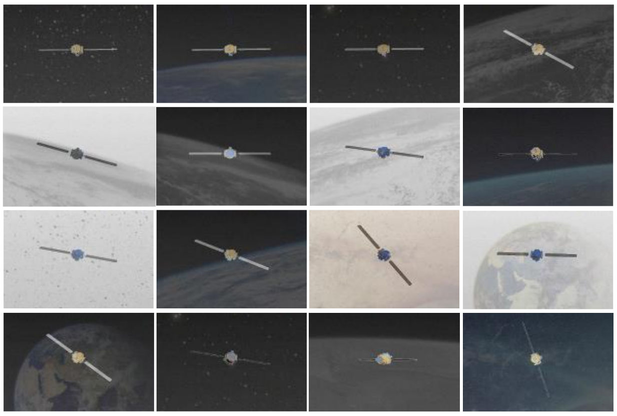


Figure 5. 16 background images of a single satellite

1.3. Dataset Labeling

Dataset labeling is a crucial step in the construction of a dataset, which has a significant impact on the final training results. Typically, dataset labeling is done manually using open-source labeling tools like labelimg. For image object detection tasks, it involves manually marking the areas of interest in the images, including the coordinates, sizes, and categories of the objects. However, this process is labor-intensive, time-consuming, and often inefficient. For this project, we face the specific challenge of satellite object detection, requiring a large-scale dataset with diverse classes and object poses. Unfortunately, the available satellite images are not labeled, necessitating manual labeling efforts. To reduce the workload, we adopted a semi-automatic labeling approach. Initially, a lightweight model was trained on a subset of already labeled data to facilitate the automatic labeling of the remaining unlabeled data [24]. Subsequently, manual corrections were made to the automatically generated labels to rectify any mislabeling or omissions. This approach significantly reduced the human labor and time costs while also contributing positively to the acquisition of a large volume of high-quality labeled data, laying a solid foundation for the development of an outstanding satellite object detection model.

In this project, we manually labeled 1,000 images. The label categories for each image included five classes: “Panel”, “Body”, “Antenna”, “Optical-load”, and “Antenna-rod”, as shown in Table 1. To improve the effectiveness of semi-automatic labeling, we randomly labeled dozens of images for each category of satellites in order to make the dataset have a better distribution so as to balance the image categories.

Table 1. Satellite components and abbreviations reference table

Solar panel	Body of the satellite	antenna	Optical load	Antenna rod
Panel	Body	Antenna	Optical-load	Antenna-rod

This semi-automatic labeling method primarily relied on a pre-trained remote sensing aircraft model (plane) achieved through transfer learning [25]. It went through 300 iterations with a batch size of 86, resulting in a small model with a training accuracy of 0.852. This model was then utilized to predict labels for 6,300 unlabeled data points. Since the model cannot achieve completely correct prediction, the Manual screening is performed to remove the incorrectly labeled data, and human-computer interaction is performed to obtain accurately labeled high-quality datasets, which together with the labeled datasets constitute a larger dataset for subsequent training. The semi-automatic labeling approach transform the role of annotators from data labeling to data verification, significantly enhancing work efficiency.

The experimental results indicate that the lightweight model obtained by using plane as the pre-training model trained under fewer samples has better detection ability [26]. The final labeling accuracy is consistently above 94%, with the best performance achieved in label "panel" at an accuracy of 0.98. The labeling of "body", which has a more complex shape with no consistent patterns, still achieves an accuracy of 0.95 or higher. However, the labeling of "antenna", "optical-load" and "antenna-rod" shows slightly lower performance but maintains accuracy above 0.9, albeit with some issues of missing and incorrect labels. After analyzing this issue, it is believed that the reason is mainly caused by the a large variability in the physical forms of these three label categories within the dataset. What's worse, these three types are characterized by a small object area and limited quantities, posing limitations on the ability to train a robust labeling model for these categories using a small sample size. This also implies that semi-automatic labeling is impossible completely replace manual labeling. Overall, this interactive, task-specific semi-automatic labeling method demonstrates significant advantages over pure manual labeling in terms of reducing the time and complexity of the labeling process, as depicted in Figure 6 in the final labeling results.

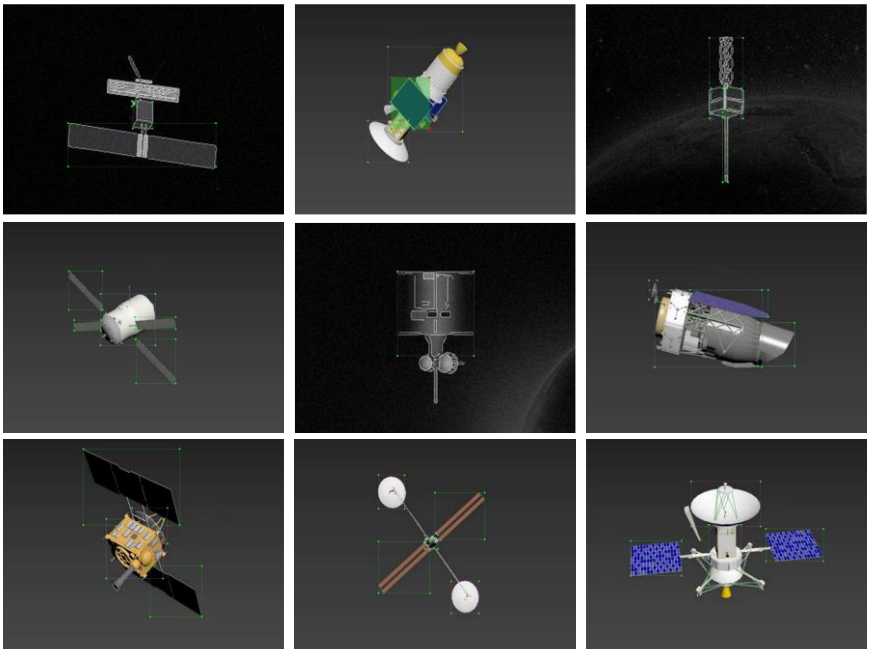


Figure 6. Partial dataset labeling results

2. Space Object Detection Algorithm Model

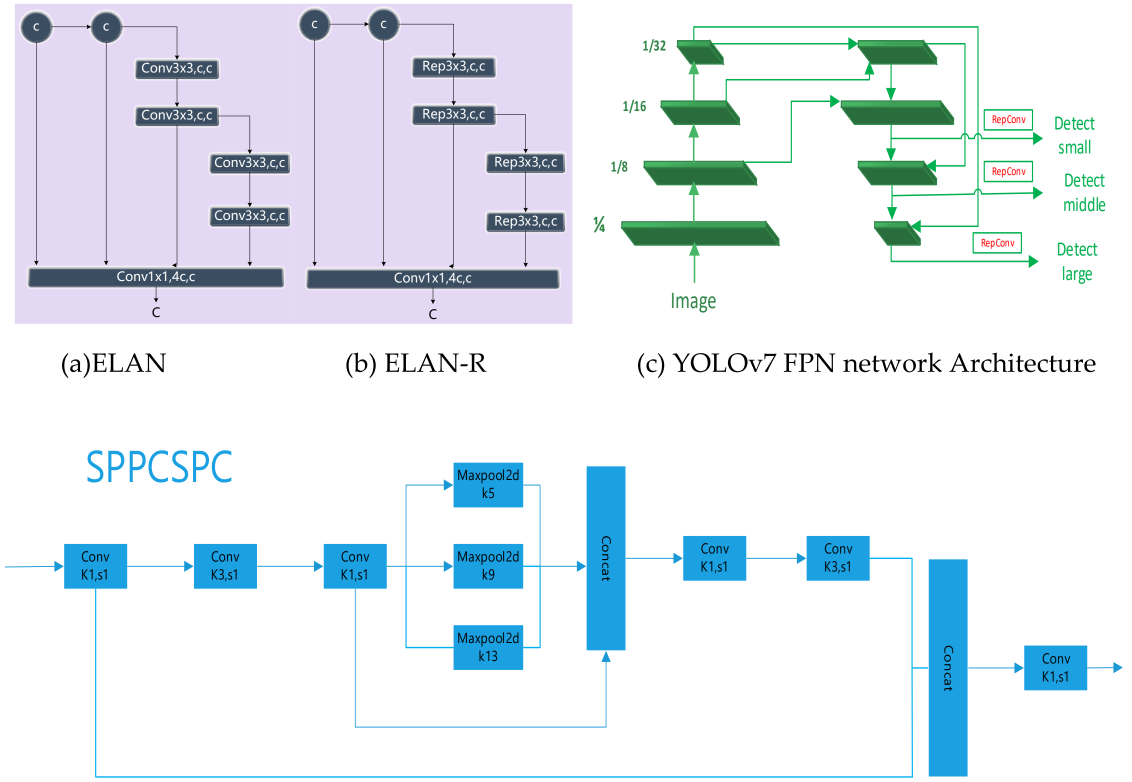
2.1. Introduction to YOLOv7

The YOLO series of algorithms are among the most advanced single-stage object detection algorithms, successfully applied in various computer vision tasks. YOLOv7, a recent iteration, introduces two new efficient network architectures, which are ELAN and E-ELAN, and incorporates model reparameterization into the network architecture. YOLOv7 achieves higher accuracy and faster speed than YOLOv3, YOLOv5, and YOLOX with the same model size.

In this work, we build YOLOv7-R upon YOLOv7, by introducing a convolution module into ELAN to obtain the ELAN-R module, resulting in noticeable accuracy improvements. The spatial pyramid pooling with context-sensitive context (SPPCSPC) module, which combines CSP and SPP structures, is retained. The CSP structure splits input features into two branches: one for feature extraction through convolution layers and another for subsequent SPP processing. This structure reduces computation while enhancing processing speed. The SPP structure uses four different max-pooling operations to obtain various receptive fields, making the algorithm adaptable to different image resolutions. In the final detection layer, the network predicts objects of different sizes on corresponding feature maps and applies non-maximum suppression to remove highly redundant prediction boxes, yielding the final predictions.

2.2. ELAN

Efficient network design typically focuses on parameters, computation, and computational density. The ELAN module incorporates the following design strategies to achieve an efficient network: controlling the lengths of the longest and shortest gradient paths to enable effective learning and convergence in deeper networks. In large-scale ELAN, regardless of gradient path lengths and the number of stacked computation blocks, a stable state can be reached. The ELAN module consists of two branches: the first branch changes channel numbers through a 1x1 convolution, while the second branch first adjusts channel numbers through a 1x1 convolution and then performs feature extraction with four 3x3 convolutions before merging the four features to obtain the final feature extraction results. ELAN employs group convolution to increase feature cardinality and combines features from different groups through shuffle and merge operations without altering the gradient propagation paths of the original architecture. This operation enhances the features learned by different feature maps, improving the utilization of parameters and computations.



(a)ELAN (b) ELAN-R (c) YOLOv7 FPN network Architecture

(d) SPPCSPC spatial pyramid pooling convolutional spatial pyramid convolution structure

Figure 7. (a) ELAN architecture diagram (b) D-ELAN architecture diagram (c) SPPCSPC structure diagram (d) YOLOv7 architecture diagram.

2.3. RepPoints and ELAN-R

As shown in Figure 8, RepPoint is a set of points that adaptively learns to position itself above the object, thereby constraining the spatial range of the object and representing locally significant regions with important semantic information [27,28,29]. The training of RepPoint is jointly driven by object localization and detection, making it closely related to the ground-truth bounding boxes and guiding the detector to correctly classify the objects. The bounding box is a 4-dimensional representation used to encode the spatial position of the object, i.e., $B = (x, y, w, h)$, where x and y represent the center point, and w and h represent the width and height. Modern object detectors rely on bounding boxes to represent objects at various stages of the detection process. However, the 4-dimensional bounding box provides a rough representation of the object's position, considering only the rectangular spatial extent of the object. This overlooks information about the object's shape, pose, and the position of semantically important local areas, which can lead to more accurate localization and better extraction of object features [30]. To overcome these limitations, RepPoints introduces a set of adaptive sample points for modeling [31], represented as:

$$\mathcal{R} = \{(x_k, y_k)\}_{k=1}^{n} \tag{1}$$

where n is the total number of sample points used in the representation. The default value of n is set to 9. The learning of RepPoint is driven by both object localization loss and object detection loss. To calculate the object localization loss, a transformation function τ is first applied to convert RepPoints into pseudo-boxes. The difference between the transformed pseudo-boxes and the ground-truth bounding boxes is then computed.

$$R_r = \{(x_k + \Delta x_k, y_k + \Delta y_k)\}_{k=1}^{n} \tag{2}$$

Here, $\{\Delta x_k, \Delta y_k\}_{k=1}^{n}$ represents the offsets. Unlike the issue of inconsistent scale in traditional bounding box regression parameters, all the offsets in the refinement process of RepPoint are handled on the same scale, avoiding scale discrepancies. During training, annotations in the form of bounding boxes are required, and predictions are converted into bounding box form during evaluation. This requires a method to transform RepPoint into bounding boxes. A predefined transformation function τ is used for this purpose: Rp-Bp, where Rp represents the RepPoint of an object P, and τ(Rp) represents the pseudo-box. The transformation function T performs a minimum-maximum operation along the two axes to determine Bp, which is the bounding box over the sample points. By replacing the 3x3 convolution kernel in the backbone with RepPoint, ELAN-R is obtained. The head part still uses the original ELAN modules. With these modifications, YOLOv7-R is created to enhance the model's geometric deformation modeling capability [32].

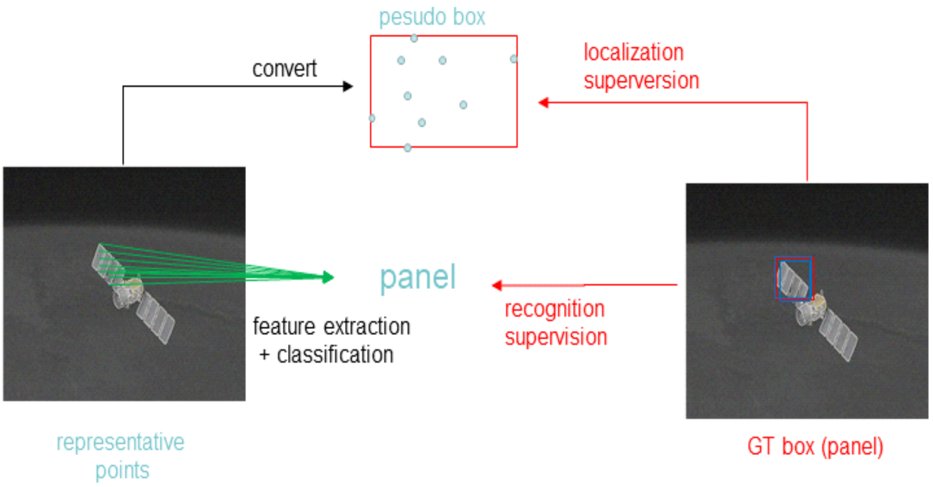


Figure 8. Illustration of the principle of deformable convolution.

3. Experimental Results and Analysis

The experiments in this study involve a dataset of 17,942 images, including 500 real data images. The input image size is 640x640. The training-test split ratio is 8:2, using a random allocation. The experiments were conducted on Ubuntu 18.04, with hardware consisting of two NVIDIA A5000 GPUs, each with 24GB of VRAM, an AMD 7371 CPU, and 56GB of RAM. A pre-trained remote sensing plane model was selected as the initial model. During training, each batch size was set to 30, and training was performed for 300 epochs. The initial learning rate was set to 0.01, gradually reduced to a final value of 0.001 using a cosine decay strategy. The optimization algorithm utilized the Adam optimizer with a momentum parameter of 0.937 and a weight decay of 0.0005. The loss function employed CIoU loss [33] for object localization and cross-entropy loss for classification. Data enhancement included operations such as Mosaic image stitching, image translation, scaling,

doi:10.20944/preprints202311.0629.v1

distortion, flipping, and random adjustments to hue, saturation, and brightness. Additionally, MixUp image blending technique was introduced to enhance data diversity.

To improve training efficiency and robustness, multi-scale training and rectangular box inference techniques were employed. Through these training enhancements, the training accuracy results shown in Table 2 were achieved. After introducing RepPoint, a significant accuracy improvement was observed. Compared to YOLOv7, YOLOv7-R enhanced with RepPoint exhibited a 1.9 percentage point increase in accuracy. The detection results are illustrated in Figure 9, and the table also presents accuracy comparisons among different models. YOLOv3 employed ResNet53 as the backbone network, which, although less capable in feature extraction compared to advanced models like YOLOv7, achieved an overall accuracy of 0.927. YOLOv7, which used the ELAN module for feature extraction, attained a higher accuracy of 0.964. The ELAN module, by controlling the longest and shortest paths, enabled the network to learn deeper semantic information and achieve faster convergence. YOLOv7-R enhanced with the RepPoint module achieved the best training results, reaching an accuracy of 0.983. This is attributed to the introduction of the RepPoint module, which enabled the backbone network to better extract feature information from satellite images. RepPoint enable precise localization at semantically meaningful positions, allowing for refined localization and improved extraction of object features in subsequent detection stages. The fact that object detection feedback does not necessarily aid object detection represented by bounding boxes further confirms the advantages of RepPoint in flexible object representation.

Table 2. Comparison of training results between YOLOv3 using the ResNet53 network architecture, YOLOv7, and YOLOv7-R. The difference between the two lies in the backbone module, with YOLOv7 utilizing the ELAN module and YOLOv7-R utilizing the ELAN-R module.

	Panel	body	antenna	Optical-load	Antenna-rod	accuracy
YOLOv3	0.915	0.931	0.934	0.892	0.905	0.927
YOLOv7	0.978	0.983	0.962	0.938	0.959	0.964
YOLOv7-R	0.986	0.99	0.972	0.974	0.993	0.983

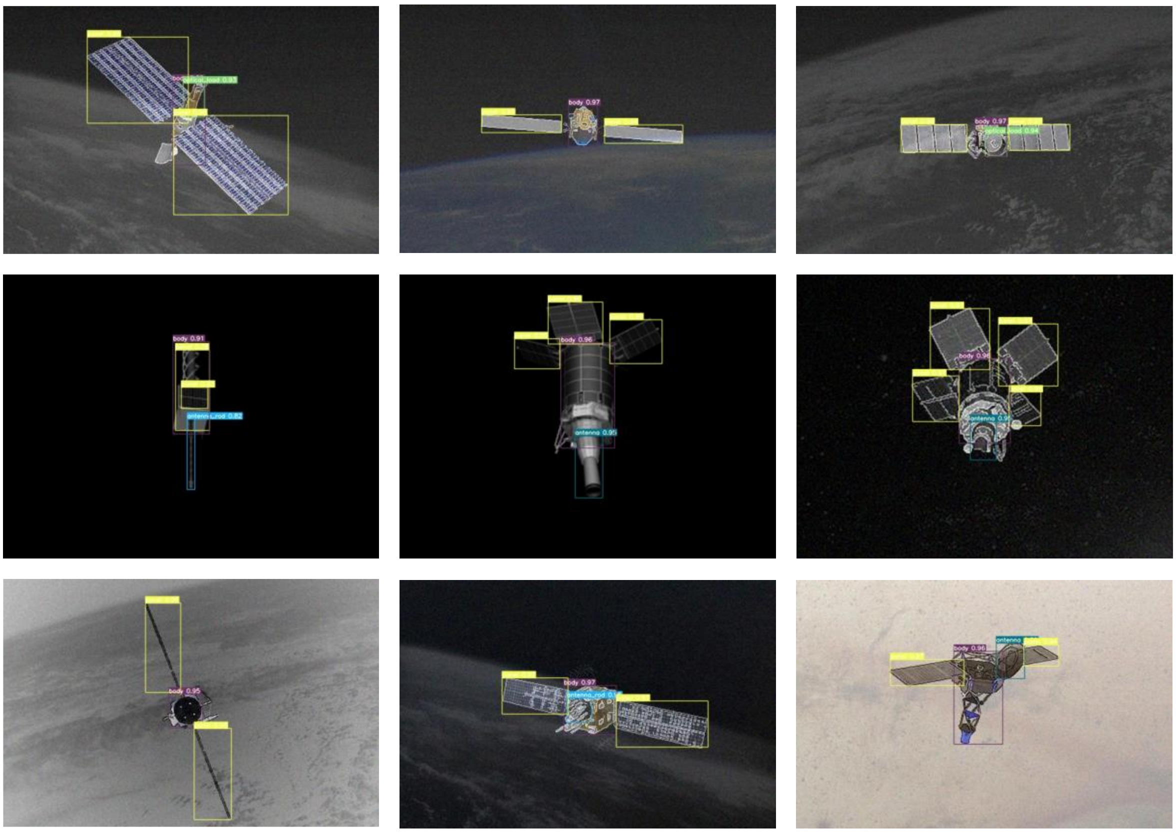


Figure 9. Satellites detection results

4. Conclusion

With the continuous advancement of space exploration, the demand for various aspects of space has been increasing. However, obtaining real satellite data remains challenging. In this project, we utilized data obtained from 3D satellite models and incorporated technical approaches to simulate realistic space satellite images as closely as possible. By applying deep learning to a novel domain, we achieved promising results. Nonetheless, this project still has certain limitations. Most of the realistic remote sensing images are large size data, which can consume significant memory resources during training. Compression techniques might lead to the loss of features for small objects and decrease in their detection accuracy. The dataset lacks diversity in terms of object variations due to the scarcity of genuine satellite data. There is considerable room for optimization in deep learning detection models, especially in achieving accurate identification of ultra-small objects. Further exploration is required in these aspects.

References

1. Sharma S, D'Amico S .Neural Network-Based Pose Estimation for Noncooperative Spacecraft Rendezvous[J].IEEE Transactions on Aerospace and Electronic Systems, 2020(6):56.
2. Phisannupawong, Thaweerath , et al. "Vision-Based Spacecraft Pose Estimation via a Deep Convolutional Neural Network for Noncooperative Docking Operations." (2020).
3. Zhu, Xiaozhou , et al. "Flying spacecraft detection with the earth as the background based on superpixels clustering." IEEE International Conference on Information & Automation IEEE, 2015.
4. Girshick, Ross , et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." IEEE Computer Society (2014).
5. Technicolor, The Stanford / , et al. "ImageNet Classification with Deep Convolutional Neural Networks [50].".
6. Girshick, Ross . "Fast R-CNN." International Conference on Computer Vision IEEE Computer Society, 2015.
7. Ren, Shaoqing , et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." IEEE Transactions on Pattern Analysis & Machine Intelligence 39.6(2017):1137-1149.
8. Joseph Redmon, et al."You Only Look Once: Unified, Real-Time Object Detection.." CoRR abs/1506.02640.(2015).
9. Berg, Alexander C. , et al. "SSD: Single Shot MultiBox Detector.", 10.1007/978-3-319-46448-0_2. 2015.
10. Joseph Redmon,and Ali Farhadi."YOLO9000: Better, Faster, Stronger.." CoRR abs/1612.08242.(2016).
11. Redmon, Joseph , and A. Farhadi . "YOLOv3: An Incremental Improvement." arXiv e-prints (2018).
12. Bochkovskiy, Alexey , C. Y. Wang , and H. Y. M. Liao . "YOLOv4: Optimal Speed and Accuracy of Object Detection." (2020).
13. Ge, Zheng , et al. "YOLOX: Exceeding YOLO Series in 2021." (2021).
14. Wang, Chien Yao , A. Bochkovskiy , and H. Y. M. Liao . "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." arXiv e-prints (2022).
15. Ploughshares, Project . "Space Security 2007." Project Ploughshares (2007).
16. West, Jessica . "The Space Security Index : Changing Trends in Space Security and the Outer Space Treaty." (2007).
17. Nicolas, Alexandre , Pascal Cté, and S. Ariafar . "The use of space surveillance systems to improve global security and international relations." 57th International Astronautical Congress 2006.
18. Pan, Sinno Jialin , and Q. Yang . "A Survey on Transfer Learning." IEEE Transactions on Knowledge and Data Engineering 22.10(2010):1345-1359.
19. Zhang H , Liu Z , Jiang Z .BUAA-SID1.0 Space Object Image Dataset[J].Spacecraft Recovery & Remote Sensing, 2010.
20. S., et al. "Direct synthesis of the Gaussian filter for nuclear pulse amplifiers." Nuclear Instruments and Methods 138.1(1976):85-92.
21. Lan, Wang . "An Improved Canny Edge Detection Algorithm." Microcomputer Information (2010).
22. Vallejo, Manuel C. , et al. "Epidural Labor Analgesia: Continuous Infusion Versus Patient-Controlled Epidural Analgesia With Background Infusion Versus Without a Background Infusion." Journal of Pain 8.12(2007):970-975.
23. Valenti, Davide , C. Guarcello , and B. Spagnolo . "Switching times in long-overlap Josephson junctions subject to thermal fluctuations and non-Gaussian noise sources." Physical review, B. Condensed matter and materials physics (2014).
24. Bo, Yuan , H. Gang , and S. Jiaguang . "Automatic dimension placement algorithm." Journal of Tsinghua University (2000).

25. Elhenawy, Mohammed , et al. "Detecting Driver Distraction in the ANDS Data using Pre-trained Models and Transfer Learning." The Australasian Road Safety Conference 2019.
26. Jun-Ying, Wang , C. Na , and L. Wei-Yi . "Application of structural EM algorithm to learning Bayesian networks for small sample." Journal of Yunnan University(Natural Sciences Edition) (2007).
27. Zhu, Xizhou , et al. "Deformable ConvNets v2: More Deformable, Better Results." (2018)..
28. [32]J. Dai et al., "Deformable Convolutional Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 764-773, doi: 10.1109/ICCV.2017.89.
29. [33] Z. Yang, S. Liu, H. Hu, L. Wang and S. Lin, "RepPoints: Point Set Representation for Object Detection," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 9656-9665, doi: 10.1109/ICCV.2019.00975.
30. [34]J. Doi and M. Yamanaka, "Discrete finger and palmar feature extraction for personal authentication," in IEEE Transactions on Instrumentation and Measurement, vol. 54, no. 6, pp. 2213-2219, Dec. 2005, doi: 10.1109/TIM.2005.858820.
31. [35] X. Pan, S. Zhu, Y. He, X. Chen, J. Li and A. Zhang, "Improved Self-Adaption Matched Filter for Moving Object Detection," 2019 IEEE International Conference on Computational Electromagnetics (ICCEM), Shanghai, China, 2019, pp. 1-3, doi: 10.1109/COMPEM.2019.8778943.
32. Lin T Y , Dollar P , Girshick R ,et al.Feature Pyramid Networks for Object Detection[J].IEEE Computer Society, 2017.DOI:10.1109/CVPR.2017.106.
33. Zheng Z , Wang P , Liu W ,et al.Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression[J].arXiv, 2019.DOI:10.1609/aaai.v34i07.6999.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.