# SDPENet: A Lightweight Spacecraft Pose Estimation Network with Discrete Euler Angle Probability Distribution

Hang Zhou[1], Lu Yao[2], Haoping She[1], and Weiyong Si[3]

***Abstract*—Utilizing deep learning techniques for spacecraft pose estimation enables using low-cost sensors like monocular cameras. However, the existing methods have drawbacks, such as complex models or low estimation accuracy. Therefore, this letter proposes the Spacecraft Discrete Pose Estimation Network (SDPENet). Firstly, we design a feature fusion network and a pose estimation head applicable to the spacecraft pose estimation task and devise the Spatial-Semantic Interaction Attention (SSIA) mechanism for feature fusion. Secondly, the discrete Euler angle probability distribution is proposed to represent the spacecraft attitude, significantly reducing the number of parameters while improving the accuracy. Finally, we put forward three data augmentation methods named CropAndPad, DropBlockSafe and Z-axis Rotation Safe to improve the performance of the network for the spacecraft pose estimation task. The experimental results demonstrate that, compared with the existing works, the errors in the spacecraft position and attitude estimated by SDPENet are reduced by 8.7%-83.1% and 31.7%-87.8% respectively, and simultaneously, the number of parameters is decreased by 33.3%-82.4%.**

***Index Terms*—AI-Based Methods, Computer Vision for Automation, Aerial Systems: Applications**

## I. Introduction

CURRENTLY, numerous spacecraft in space are undertaking different on-orbit missions, including in-orbit services for in-service spacecraft, such as rendezvous and docking, space refueling, etc [1]. Decommissioned spacecraft, malfunctioning spacecraft, and space debris need to be moved to graveyard orbits or made to re-enter the atmosphere to avoid threatening other space facilities and relieve the tense situation of orbital resources [2]. Because of the situations mentioned above, the demands for space missions such as On-Orbit Services (OOS) and Active Debris Removal (ADR) are steadily increasing. They are considered critical capabilities in the aerospace field in the next decade [3]. Numerous research projects have been carried out regarding these space missions, for example, the Phoenix Program of the Defense Advanced Research Projects Agency of the United States [4] and the ClearSpace-1 mission of the European Space Agency [5].

Manuscript received: November, 21, 2024; Revised: January, 7, 2025; Accepted: January, 28, 2025.

This paper was recommended for publication by Editor Giuseppe Loianno upon evaluation of the Associate Editor and Reviewers' comments. (Corresponding authors: Haoping She; Weiyong Si)

[1]Hang Zhou and Haoping She (Corresponding author) are with School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China 3120220120@bit.edu.cn; shehp@bit.edu.cn

[2]Lu Yao is with Space Pioneer, Beijing 100076, China meet_lu@163.com

[3]Weiyong Si (Co-Corresponding author) is with the School of Computer Science and Electronic Engineering at the University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK. w.si@essex.ac.uk

Digital Object Identifier (DOI): see top of this page.

In these tasks, obtaining the position and attitude of spacecraft is of great necessity. However, particular spacecraft, especially non-cooperative ones, do not have recognizable attitude marks. Some vision-based studies suggest that active sensors like Light Detection and Ranging (LIDAR) [6], [7] can be adopted. Despite their application on spacecraft, these sensors are characterized by high cost and power consumption. In contrast, the monocular camera has several advantages: small size, lightweight, low cost, and low power consumption [8]. With the development of deep learning, using a monocular camera in combination with deep learning for spacecraft pose estimation can bypass processes such as edge and corner extraction, and directly generate the attitude of the spacecraft from the images input by the monocular camera in an end-to-end manner.

However, performing pose estimation on the target spacecraft using a monocular camera presents a challenging task. Firstly, the training of the neural network demands a vast amount of data and labels to guarantee the model's prediction accuracy. Secondly, because of the varying sizes and positions of the target spacecraft, its size within the camera's field of view changes significantly. Moreover, in space, the camera might capture the background of the Earth, leading to relatively low pose estimation accuracy. Finally, some algorithms for estimating the pose of spacecraft from two-dimensional images are relatively complex, such as the hybrid modular approaches, which involves multiple modules.

To tackle the aforementioned problems, we propose a lightweight and efficient convolutional neural network, the Spacecraft Discrete Pose Estimation Network (SDPENet), for estimating the pose of spacecraft in an end-to-end manner. This letter's principal contributions are presented below.

1) Apart from the backbone network, a fusion neck and a pose estimation head are designed to enhance the network's feature extraction capability. Additionally, a Spatial-Semantic Interaction Attention (SSIA) mechanism is devised to fuse high-level and low-level features, which improves the network's pose estimation accuracy.
2) We propose a discrete Euler angle probability distribution representation method. This method discretizes the spacecraft's attitude space and predicts the probability distribution within the attitude space via a network, significantly superior to the existing methods.
3) Since deep-learning-based methods rely on a large amount of data for training to enhance the network's performance, three data augmentation methods named CropAndPad, DropBlockSafe and Z-axis Rotation Safe are proposed for the spacecraft pose estimation task.

## II. Related Work

### A. Hybrid modular approaches

Currently, there are two deep-learning-based methods for spacecraft pose estimation. One is the hybrid modular approaches, which commonly integrates deep-learning models with classical computer vision algorithms to conduct pose estimation of the target spacecraft through multiple stages, including i. spacecraft positioning; ii. keypoint detection; iii. pose estimation. For instance, B Chen et al. [9] adopted Faster-RCNN for spacecraft positioning, Pose-HRNet-W32 for keypoint prediction, and PnP+RANSAC for pose estimation. Li K et al. [10] utilized YOLOX-Tiny to predict the spacecraft's position, CSPDarknet53+FPN for keypoint prediction, and ultimately EPnP for spacecraft pose estimation. S. Wang et al. [11] directly employed DarkNet-53+FPNs for keypoint prediction and the PnP algorithm for pose estimation. Z. Wang et al. [12] used a CNN to locate the spacecraft's position and then a Transformer-based model for keypoint prediction.

However, such methods may involve multiple neural network models, and the entire process is relatively complex in terms of computation, which is not conducive to deployment in scenarios where computing resources are limited. Besides, this type of methods usually requires the three-dimensional CAD model of the target satellite to define key points during the key point prediction stage, which may become a limitation of these methods under certain circumstances.

### B. Direct end-to-end approaches

Another approach is the direct end-to-end method. Compared with the hybrid modular approaches, it is simpler and does not require the three-dimensional information of the target. Thus, it is more suitable to be deployed in embedded devices. Compared with the hybrid modular approaches, the end-to-end method requires a single neural network model to directly regress the pose from the image of the target spacecraft without the need for intermediate stages. Phisannupawong et al. [13] proposed a modified pre-trained GoogLeNet to regress the 7D vector $[x, y, z, q_0, q_1, q_2, q_3]$ representing the pose and examined different loss functions. Sharma et al. [14] proposed the SPN model, which consists of five layers of CNN followed by three distinct branches. The first branch is responsible for predicting the bounding box of the spacecraft, while the other two branches are used to predict the pose. Proença et al. [15] proposed URSONet with ResNet as the backbone network, followed by two branches for predicting the position and pose, respectively. The position estimation is regressed through two fully connected layers, while the direction estimation is achieved through classification and soft-assignment coding. Garcia et al. [16] utilized a network with the UNet structure to predict the 3D position and the bounding box of the spacecraft. The cropped bounding box is then input into the second CNN for regressing the spacecraft's pose. Yao et al. [17] proposed Mobile-SPPEDNet with MobileNetv2 as the backbone network, utilized the Coordinate Attention and Spatial Pyramid Pooling (SPP) in the network, and achieved an improvement in pose estimation accuracy with a reduced number of parameters.

However, in the proposed end-to-end method, only the output of the last layer of the neural network is utilized to predict the spacecraft pose, indicating a gap compared with the development of neural networks in the computer vision field. On the other hand, the existing spacecraft attitude representation methods, such as direct representation, classification representation, and probability mass function representation, all present issues such as significant estimation errors, overfitting, or a large number of network parameters.

To address the issues in deep learning methods, this letter presents SDPENet for spacecraft pose estimation. Besides the backbone, it has a neck fusion network with the SSIA mechanism and a pose estimation head. Representing the spacecraft pose via the discrete probability distribution of Euler angles boosts accuracy while maintaining model lightness. Also, considering the drawbacks of current data augmentation, we propose three methods: CropAndPad, DropBlockSafe and Z-axis Rotation Safe.

## III. Proposed Methods

### A. Network Architecture

The architecture of the SDPENetwork is depicted in Figure 1. The backbone of this network is MobileNetV3, and the feature maps of the second, third, fourth, and fifth stages are fed into the fusion neck with Spatial-Semantic Interaction Attention (SSIA) mechanism. Subsequently, the output of the fusion neck is input into the pose estimation head for regressing the spacecraft's position, as well as the discrete Euler angle probability distribution. Finally, the probability distribution is decoded to obtain the spacecraft attitude.

*1) SDPENet:* The backbone of the SDPENet is MobileNetV3-Large, which is pre-trained on ImageNet-1k. It primarily consists of depthwise separable convolutions, significantly reducing network parameters and computational complexity. According to the varying sizes of feature maps, it is divided into different stages. Depthwise separable convolutions with a stride of 2 are employed for downsampling between different stages. Firstly, the image is input into a convolutional layer. In this layer, the number of channels is changed from 1 to 3 to meet the input requirements of the MobileNetV3 network. Moreover, the convolutional layer with 960 channels, originally designed for classification tasks, and the subsequent pooling and convolutional layers are removed.

Yao's research [17] reveals that fusing features from different stages can significantly enhance the pose estimation accuracy for target spacecraft. Consequently, the FPN+PAN structure is employed in the neck section to integrate the feature maps of the 2nd, 3rd, 4th, and 5th stages.

There are three feature maps of different sizes output by the neck. Subsequently, in the head part, global average pooling with sizes of $4 \times 4$, $2 \times 2$ and $1 \times 1$ is applied to the largest, medium-sized, and smallest feature maps, respectively. Compared with uniformly using $1 \times 1$ convolution, this approach can preserve more spatial information of the shallow layer.

The pooled features are then stretched and spliced and then input into a fully connected layer for feature fusion. The fully connected layer outputs a one-dimensional feature vector $F$.

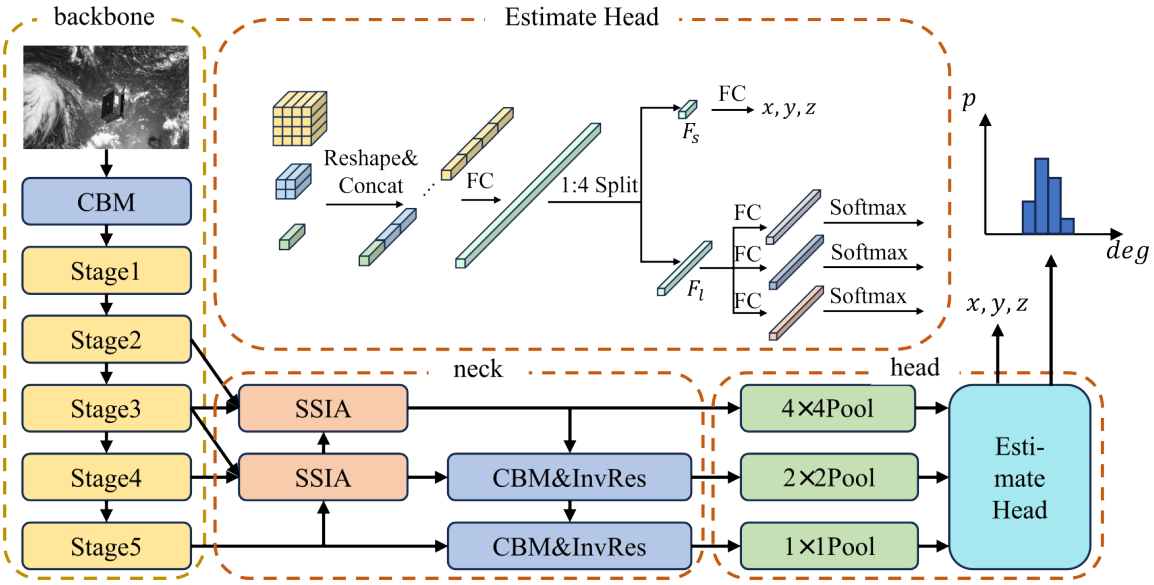



Fig. 1: The structure of SDPENet. The CBM module is a series connection of the convolutional module, batch normalization, and the Mish activation function. The network's input is a gray picture. The position prediction branch directly outputs the spacecraft's position, and the pose prediction branch outputs the discrete Euler angle probability distribution.

Given that position estimation is less complex than attitude estimation, the feature $F$ is partitioned into $F_s$ and $F_l$ at a ratio of 1:3. $F_s$ is fed into the subsequent fully connected layer for position regression. $F_l$ is input into three subsequent fully connected layer branches to output the discrete probability distributions of the three Euler angles, respectively.

*2) spatial-semantic interaction attention mechanism:* The FPN+PAN structure performs feature fusion by concatenating feature maps along the channel dimension. However, this approach is rather simplistic. Define the three inputs of the SSIA mechanism as current-level feature map $F_i$, the deep-level feature map $F_{i+1}$, and the shallow-level feature map $F_{i-1}$. The SSIA mechanism generates spatial and channel-wise attention weights from the $F_{i-1}$ and $F_{i+1}$ respectively and then applies them to $F_i$, thus enabling the network to pay attention to more critical features and enhancing the feature extraction ability of the network. This allows the network to capture more significant information during the feature fusion process.

Since the feature map $F_{i+1}$ contains more high-level semantic information, it is fed into the MLP after global pooling to obtain the attention weight in the channel direction, which is then applied to $F_{i-1}$. Specifically, in this architecture, two branches of pooling operations, namely global average pooling and global maximum pooling, are employed simultaneously. The global average pooling branch calculates the average value of each channel across the entire spatial extent of the feature map. This operation effectively preserves the overall features of the input, providing a comprehensive summary of the information within each channel. On the other hand, the global maximum pooling branch extracts the maximum value within each channel over the spatial domain. This process is particularly useful in capturing the extreme features or the most salient information within the feature map. By combining these two types of pooling results, the network can better utilize both the general and the most distinctive characteristics of the input data. Then we add the results of the two global pooling operations together and input them into a MLP. Subsequently, the weights in the channel direction are obtained through the Sigmoid function.

Meanwhile, the shallow-level $F_{i-1}$ encompasses more spatial information, it is fed into convolutional module with an output channel of 1 to obtain the attention weight in the spatial direction. As same as channel attention, there are two branches involved in this process. The dilations of these two branches are set to 0 and 1 respectively. Dilation is an important parameter in convolutional operations as it determines the spacing between the elements of the kernel during the convolution process. By setting different dilation values for the two branches, the model aims to enhance the receptive field of the attention mechanism. A larger dilation value in one of the branches allows the model to capture a broader range of spatial information, while the other branch with a smaller dilation value can focus on more local details. This combination of different dilation values in the two branches enables the attention mechanism to have a more comprehensive and flexible perception of the spatial structure within the feature maps, thereby improving the performance of the model in handling complex spatial relationships within the data.

### *B. Discrete Euler Angle Probability Distribution*

Unlike directly regressing the position of the target spacecraft, regressing the spacecraft's attitude fails to represent the actual angular distance correctly. Some works [15] experimentally found that when using $L_1$ or $L_2$ loss functions for training, the network has a significant prediction error and

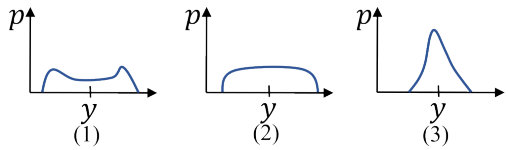



Fig. 2: Different flexible distributions can obtain the same integral target according to Eq. (1).

overfitting. Thus, [15] proposed the Probabilistic Orientation Soft Classification method. The network outputs a probability mass function and fits the quaternion by minimizing the weighted least-squares method. However, this method leads to an overly large dimension (up to $24^3$) of the last fully connected layer, resulting in a vast number of parameters and computational complexity.

Inspired by [19], we propose discrete Euler angle probability distribution to learn the spacecraft attitude. Given the range of Euler angle $y$ with minimum $y_0$ and maximum $y_m$ $(y_0 \leq y \leq y_m)$, we can have the estimated Euler angle $\hat{y}$ $(y_0 \leq \hat{y} \leq y_m)$ from the model:

$$\hat{y} = \int_{-\infty}^{+\infty} P\left(y_i\right) y_i \mathrm{d}y_i = \int_{y_0}^{y_n} P\left(y_i\right) y_i \mathrm{d}y_i \tag{1}$$

To be consistent with neural network, we convert the integral over the continuous domain into a discrete representation, via discretizing the range $[y_0, y_m]$ into a set $\{y_i, (i = 1, \cdots, n)\}$ with intervals $s$. Consequently, given the discrete distribution property $\sum_{i=0}^{n} P\left(y_i\right) = 1$, the estimated regression Euler angle $\hat{y}$ can be presented as:

$$\hat{y} = \sum_{i=0}^{n} P\left(y_i\right) y_i \tag{2}$$

As a result, $P(y_i)$ can easily implemented through a softmax layer consisting of $n + 1$ units. However, there are infinite combinations of values for $P\left(y_i\right)$ that can make the final integral result being $y$, as shown in 2, which may reduce the learning efficiency. Intuitively, compare against distribution (1) and (2), distribution (3) is compact and tends to be more confident and precise on the Euler angle estimation, which motivates us to take distribution (3) as the optimization objective. Therefore, the model should be forced to rapidly focus on the values near label $y$. The formula for transforming the Euler angle $y$ into the discrete Euler angle probability distribution are as follows, where $y_i$ and $y_{i+1}$ are the nearest Euler angles on the left and right sides of the Euler angle $y$, respectively.

$$P(y_i) = \frac{y_{i+1} - y}{y_{i+1} - y_i} \tag{3}$$

$$P(y_{i+1}) = \frac{y - y_i}{y_{i+1} - y_i} \tag{4}$$

However, when the Euler angle label $y$ is in close proximity to $y_i$, the probability distribution may tend to be a large unimodal distribution. As indicated in [20], the network performs better when predicting a relatively smooth probability distribution. Consequently, on this basis, the discrete probability of the Euler angle is dispersed to both sides. The specific process is presented in Algorithm 1, where $\alpha$ is the dispersion ratio and $n$ is the number of dispersions. Figure 4 illustrates the coding process of the discrete probability of the Euler angle.

**Algorithm 1** Translate Euler Angle to Discrete Probability Distribution

**Input:** The Euler angle $y$, the stride of discrete Euler angle $s$, the number of dispersions $n$, and the dispersion ratio $\alpha$.

**Output:** The encoded Euler angle discrete probability distribution $\mathbf{Y}$.

$\mathbf{Y} \leftarrow 0$
$m \leftarrow y/s$ ▷ The position of Euler angle $y$ in the discrete grids $y_i$
$l \leftarrow \lfloor m \rfloor$ ▷ The position of the nearest discrete Euler angle on the left side
$r \leftarrow \lceil m \rceil$ ▷ The position of the nearest discrete Euler angle on the right side
**if** $l = r$ **then**
    $\mathbf{Y}_{\lfloor l \rfloor + n} \leftarrow 1$ ▷ The weight is 1 if Euler angle $y$ on the discrete Euler angle precisely
**else**
    $\mathbf{Y}_l \leftarrow (r - m)/(r - l)$ ▷ Allocate weight linearly to the discrete Euler angles on the left side
    $\mathbf{Y}_r \leftarrow (m - l)/(r - l)$ ▷ Allocate weight linearly to the discrete Euler angles on the right side
**end if**
$p \leftarrow 1$
**for** $i = 1, 2, \cdots, n$ **do**
    $\mathbf{Y}_l \leftarrow \mathbf{Y}_l \times (1 - \alpha)$ ▷ Reduce the weight on the left side according to the proportion $\alpha$
    $\mathbf{Y}_r \leftarrow \mathbf{Y}_r \times (1 - \alpha)$ ▷ Reduce the weight on the right side according to the proportion $\alpha$
    $l \leftarrow l - 1$
    $r \leftarrow r + 1$
    $p \leftarrow p \times \alpha$
    $\mathbf{Y}_l \leftarrow (r - m)/(r - l) \times p$ ▷ Allocate the weight linearly to the further left side according to the proportion $\alpha$
    $\mathbf{Y}_r \leftarrow (m - l)/(r - l) \times p$ ▷ Allocate the weight linearly to the further right side according to the proportion $\alpha$
**end for**

### C. Loss Function and Metrics

The loss function of the position regression branch is $L_2$ loss function as shown in (1), where $\hat{t} = [\hat{x}, \hat{y}, \hat{z}]^T$ is the value predicted by the network, $t = [x, y, z]^T$ is the label of the position.

$$L_{pos} = L_2(t, \hat{t}) = (x - \hat{x})^2 + (y - \hat{y})^2 + (z - \hat{z})^2 \tag{5}$$

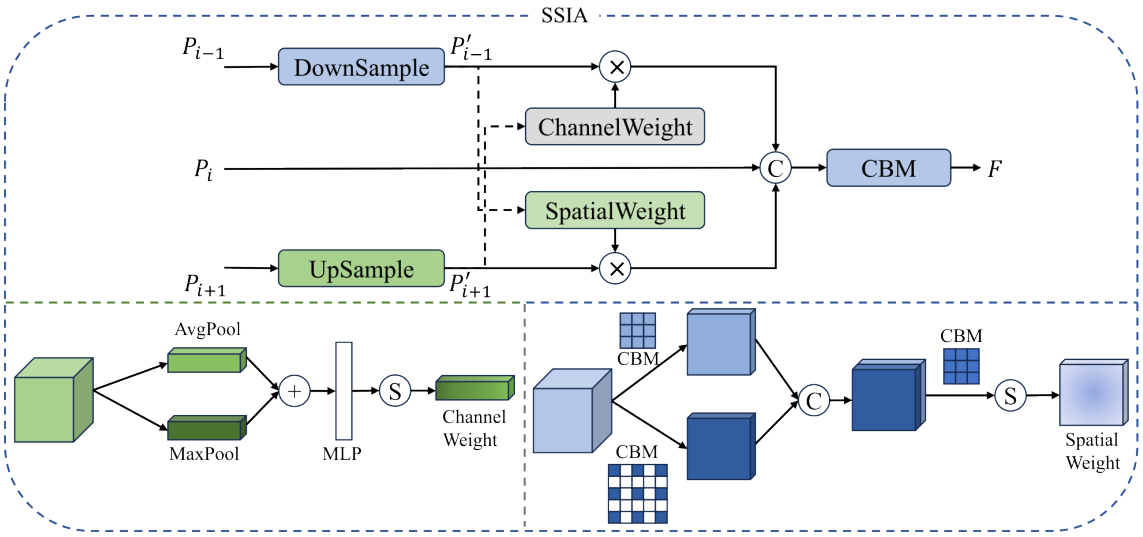



Fig. 3: The structure of SSIA, as well as the calculation processes of the channel attention weights and the spatial attention weights.

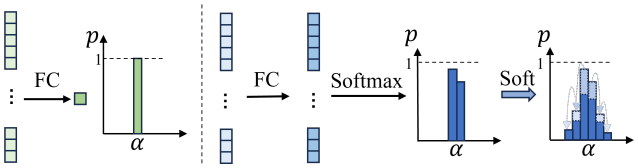



Fig. 4: The schematic diagram of the encoding process for the discrete Euler angle probability distribution transforms the fixed Euler angle into a discrete probability distribution.

The cross-entropy loss function is adopted to force the discrete Euler angle probability distribution approach the encoded Euler angle probability distribution $P(y_i)$.

$$L_p = -\sum_{i=1}^{n} P(y_i)\log\left[P(\hat{y}_i)\right] \tag{6}$$

Although representing the Euler angle with discrete probability distributions and optimizing with the cross-entropy loss function can quickly make the network-predicted attitude converge to the vicinity of the true value, the three Euler angles are predicted independently, failing to reflect their coupling relationship. Thus, additional angle constraints are required for more accurate angle prediction. After decoding the network-predicted discrete Euler angle probability distribution and converting it to a quaternion, an inverse cosine loss function is additionally introduced for the decoded quaternion, where $q$ is the true spacecraft quaternion attitude and $\hat{q}$ is the spacecraft quaternion decoded from the discrete Euler angle probability distribution output by the network.

$$L_{ori} = 2 \cdot \cos^{-1}(\left|\hat{q} \cdot q^T\right|) \tag{7}$$

The total loss function is obtained by weighted summation through coefficients $\beta_i$.

$$L = \beta_1 L_{pos} + \beta_2 L_p + \beta_3 L_{ori} \tag{8}$$

We use the evaluation metrics in the Satellite Pose Estimation Challenge (SPEC) [21] to facilitate comparison with other works. The position error is defined as

$$E_t = \left\|t - \hat{t}\right\|_2 \tag{9}$$

and the the attitude error is defined as

$$E_q = 2 \cdot \cos^{-1}(|\langle \hat{q}, q \rangle|) \tag{10}$$

### D. Data Augmentation

Existing data augmentation techniques for spacecraft pose estimation are mainly divided into pixel-level and spatial-level ones. Both transform original images. The difference is that pixel-level augments only change images without affecting pose labels, while spatial-level augments change both images and labels.

However, not all data augmentation methods enhance network performance. For instance, using random erasing alone reduces pose estimation accuracy [3], because most spacecraft regions may be discarded during this process. Most spatial-level augmentation methods, such as random flipping [10], random rotation [12], are not suitable for direct end-to-end approaches as spacecraft pose labels after spatial transformations are unavailable, therefore, they cannot be applied haphazardly. To address the above issues, we proposed CropAndPad, DropBlockSafe and Z-axis Rotation Safe.

*1) Pixel-level Data Augmentation:* In the training process, many classical pixel-level data augmentation are adopted, such as Gaussian noise addition, blurring, and adjustments in brightness, contrast, saturation, and hue. These augmentation operations are executed with Albumentation [18].

CropAndPad performs random cropping around the spacecraft region to preserve the entire spacecraft image. After the cropping step, a copy-padding is employed rather than scaling the image to restore it to its original size. In this way, the

image is refilled to its original dimensions while maintaining the integrity of the spacecraft's position and orientation within the image.

The DropBlockSafe method, on the other hand, randomly selects and discards some rectangular areas within the image, but these areas are chosen not to cover the target spacecraft. The colors of these discarded rectangular areas are then filled randomly. This helps introduce a certain level of variability in the training data without interfering with learning the spacecraft's features.

*2) Spatial-level Data Augmentation:* We utilized rotating around the Z-axis of the camera to achieve spatial-level data augmentation. Different from simply rotating around the center of the image, this method generates the corresponding spacecraft pose labels according to the camera imaging principle and the rotation matrix. Moreover, we have taken into consideration the situation where the spacecraft is at the edge of the image. We stipulate that $r = a_r/a_o \geq 0.7$ where $a_o$ and $a_r$ are the areas of the spacecraft region in the image before and after rotation respectively. If this condition is not met, random rotation will be carried out again until the condition is satisfied. In this way, we can set the angular range of random rotation around the Z-axis of the camera from $-180^\circ$ to $180^\circ$, thus generating training data with more poses.

Let the world coordinates of a point on the spacecraft be $[X_w, Y_w, Z_w]$, and the corresponding coordinates in the pixel coordinate system be $[u, v]$. The transformation relationship between the two is:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{11}$$

Among them, $\mathbf{K}$ is the internal parameter matrix of the camera, and $[\mathbf{R} \quad \mathbf{T}]$ is the external parameter matrix, whose values are determined by the pose of the spacecraft.

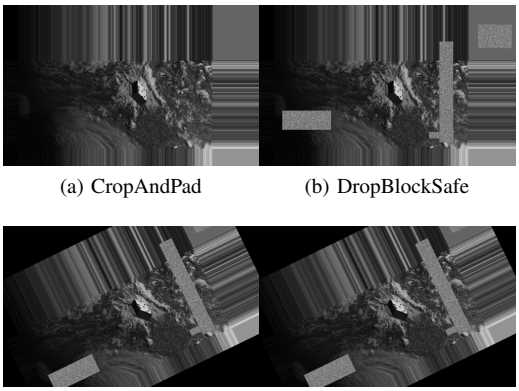
The above formula allows one to obtain (11). The left side of the equation represents the rotated image. On the right side of the equation, $\mathbf{R}_a$ is the rotation matrix around the Z-axis of the camera coordinate system. $\mathbf{R}_a [\mathbf{R} \quad \mathbf{T}]$ is the external parameter matrix of the camera after data augmentation, corresponding to the pose of the spacecraft after the image rotation.

$$\mathbf{K}\mathbf{R}_a\mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}\mathbf{R}_a \begin{bmatrix} \mathbf{R} & \mathbf{T} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_W \\ 1 \end{bmatrix} \tag{12}$$

## IV. Experimental Results

### A. Dataset

A series of experiments were conducted using the Spacecraft Pose Estimation Dataset (SPEED) [14]. This dataset is derived from the Tango spacecraft of the PRISMA mission, enabling researchers to test the performance of spacecraft pose estimation and evaluate diverse methods on the same image dataset. The dataset encompasses 12,000 images with spacecraft pose labels, with an image size of 1200×1920. The distances between the spacecraft and the camera corresponding to these images range approximately from 5 meters to 40 meters. In the SPEED dataset, the image backgrounds consist partly of dark space and partly of a rendered Earth. Given that the SPEED test set's labels have not been publicized, in this letter, the training set and validation set are partitioned at a ratio of 0.85:0.15, and training and evaluation are subsequently performed on this basis.



(a) CropAndPad (b) DropBlockSafe

(c) Z-axis Rotation Safe (d) Albumentation

Fig. 5: The result of spatial-level data augmentation and pixel-level data augmentation. The order is CropAndPad, DropBlockSafe, Z-axis Rotation Safe and Albumentation.

### B. Implement Details

A single NVIDIA RTX 4090 is utilized. The optimizer adopted is AdamW, with the weight decay regularization coefficient set to 1e-5. The initial learning rate is 0.001, and a single-cycle cosine decay learning rate adjustment strategy is employed, where the number of warm-up rounds is 5. The batch size is set to 32. The backbone network is initialized with the pre-trained weights of MobileNetv3-Large on ImageNet-1k. After data augmentation, the images are resized to $480 \times 768$ to enhance training and inference efficiency. In the loss function, $\beta_1$, $\beta_2$ and $\beta_3$ are set to 1, 10, and 6 respectively so that each loss is in the same order of magnitude. Table I presents the probabilities of the data augmentation we employed. We additionally carried out sun light data augmentation with a probability of 0.5 on the training and validation sets in the sun light experiment, so as to verify the performance of the model in the real space environment.

### C. Results

In order to verify the effectiveness of the network structure proposed in this study, ablation experiments were conducted on the SSIA mechanism within the neck and head parts. The results are presented in Table II. Each network structure designed herein can perform feature extraction and fusion more effectively, thereby gradually enhancing the network's performance.

TABLE I: DATA AUGMENTATION AND CORRESPONDING PROBABILITIES

| Data augmentation | Probabilities |
|---|---|
| Gaussian noise | 0.2 |
| Blurring | 0.2 |
| Brightness/contrast/saturation/hue | 0.1 |
| CropAndPad | 0.5 |
| DropBlockSafe | 0.5 |
| Z-axis Rotation Safe | 0.8 |
| Sun light (only for sun light exp.) | 0.5 |

TABLE II: ABLATION EXPERIMENT RESULTS OF DIFFERENT NETWORK ARCHITECTURES

| Structure | $E_t(m)$ | $E_q(^\circ)$ |
|---|---|---|
| Backbone | 0.2737 | 8.4901 |
| +FPN&PAN | 0.1727 | 2.3694 |
| +SSIA | 0.1646 | 2.1256 |
| +head | **0.1323** | **1.7003** |

Table III presents the influence of the discrete Euler angle probability distribution with varying parameters on the network prediction results. The network performs optimally when $s = 5$, $n = 2$ and $\alpha = 0.1$. The parameter $s$ governs the stride of the discrete Euler angle. An increase in $s$ leads to enhanced network robustness and a reduction in the dimension of the final output, thereby enabling more sufficient feature utilization. The parameters $n$ and $\alpha$ control the smoothness of the discrete distribution of the Euler angle. A smoother distribution results in a better network prediction effect.

Furthermore, Table IV illustrates the influence of data augmentation on network performance. It is evident that pixel-level data augmentation mitigates the pose estimation error of the network to a certain degree and enhances the network performance. Given that rotation around the Z-axis of the camera substantially enriches the spacecraft poses within the dataset, the spatial-level data augmentation approach significantly reduces the errors in both position and pose estimation. The combined effect of all these data augmentation methods allows the network to learn more abundant features and remarkably improves the network's performance.

Table V presents a comparison between our work and other related studies. The proposed model in our research features a relatively low number of parameters. This advantage stems from the lightweight design of MobileNetV3-Large and the utilization of discrete Euler angle probability distribution for representing, which reduces the dimension of the final detection head from $N^3$ to $3N$. Additionally, this model's position and pose estimation errors are decreased by $8.7\% - 83.1\%$ and $31.7\% - 87.8\%$ , respectively. In contrast to the URSONet model with similar pose estimation performance, the parameter number of our model is reduced by $33.3\% - 82.4\%$. Although the performance of the model declined to some extent in the sun light experiment, it still outperformed most other models. Furthermore, Figure 6 illustrates some pose estimation results of SDPENet on the validation set.

TABLE III: THE RESULT OF THE DISCRETE EULER ANGLE PROBABILITY DISTRIBUTION IMPLEMENTATION WITH DIFFERENT PARAMETERS

| $\alpha$ | $s$ | $n$ | $E_t(m)$ | $E_q(^\circ)$ |
|---|---|---|---|---|
| 0.1 | 1 | 0 | 0.2246 | 2.0808 |
| | | 2 | 0.3123 | 2.8390 |
| | | 4 | 0.1748 | 1.9956 |
| | 2 | 0 | 0.1727 | 2.1679 |
| | | 2 | 0.1870 | 2.4387 |
| | | 4 | 0.1672 | 1.7952 |
| | 5 | 0 | 0.1407 | 2.1882 |
| | | 2 | **0.1323** | **1.7003** |
| | | 4 | 0.1357 | 2.1240 |
| 0.01 | 5 | 2 | 0.1588 | 1.8513 |
| 0.2 | 5 | 2 | 0.1433 | 1.7094 |

TABLE IV: THE IMPACT OF DATA AUGMENTATION ON NETWORK PERFORMANCE

| Augmentation | $E_t(m)$ | $E_q(^\circ)$ |
|---|---|---|
| +Noise/Blur/ColorJitter | 0.3562 | 12.3661 |
| +CropAndPad | 0.4777 | 10.8769 |
| +DropBlockSafe | 0.2737 | 8.4901 |
| +Rotate | **0.1323** | **1.7003** |

## V. CONCLUSIONS

We have proposed a lightweight spacecraft pose estimation network named SDPENet, which employs MobileNetV3 as the backbone network. We have designed the SSIA and applied it in the neck part, enhancing the feature fusion ability of the network. Moreover, we have specifically designed the head part of the network for the spacecraft pose estimation task.

In addition, we have proposed a discrete Euler angle probability distribution method to represent the spacecraft attitude. This method can significantly improve the accuracy of spacecraft attitude estimation while ensuring that the number of model parameters remains rather low. Finally, we have designed pixel-level data augmentation and spatial-level data augmentation, which have greatly enhanced the accuracy and generalization ability of the model.

However, we have only tested one discrete Euler angle probability distribution approach and only experimented with limited parameters, without testing other combinations. Our experiments were only conducted on the SPEED dataset. In the future, we will test multiple discrete Euler angle probability

TABLE V: COMPARISON OF SPACECRAFT POSE ESTIMATION ERRORS AMONG DIFFERENT MODELS, AS WELL AS THE NUMBER OF PARAMETERS OF DIFFERENT MODELS

| Model | Params(M) | $E_t(m)$ | $E_q(^\circ)$ |
|---|---|---|---|
| LSPnet[16] | 47.8 | 0.4560 | 13.9600 |
| SPN[14] | - | 0.7832 | 8.4254 |
| URSONet[15] | 11.4~42.8 | 0.1450 | 2.4900 |
| Mobile-URSONet[22] | 7.4 | 0.5600 | 6.2900 |
| Mobile-SPEEDNet[17] | 7.1 | 0.2500 | 5.2100 |
| SDPENet(Ours) | 7.6 | **0.1323** | **1.7003** |
| SDPENet(Sun light exp.) | 7.6 | 0.2353 | 3.1243 |

distribution approaches, conduct experiments on multiple sets of parameters, and carry out experimental verification on multiple datasets.

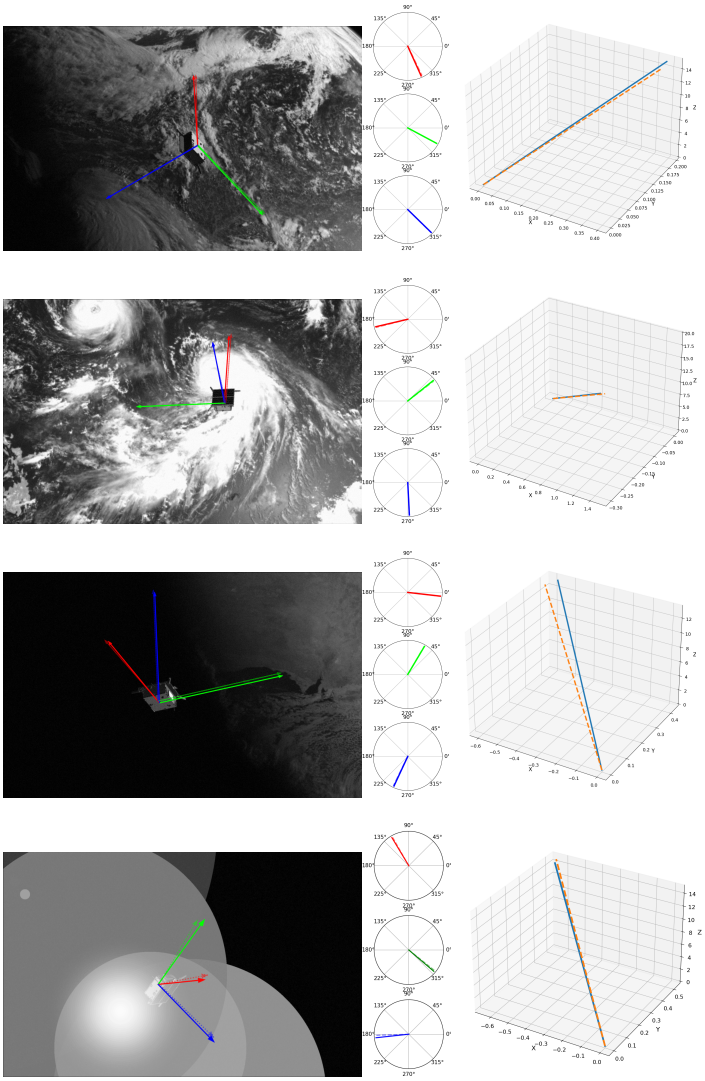


Fig. 6: The pose estimation results of SDPENet are presented as follows: The first column shows the labels and prediction results of the spacecraft pose; the second column presents the labels and estimation results of the three axes; the third column gives the labels and estimation results of the position. The solid line represents the label content, and the light-colored dashed line represents the pose estimation results. The image in the last row shows the results of sun light experiment.

## REFERENCES


[1] B. B. Reed, R. C. Smith, B. J. Naasz, J. F. Pellegrino and C. E. Bacon, "The Restore-L Servicing Mission," AIAA SPACE 2016, pp. 5478, Sep. 2016

[2] J. L. Forshaw, G. S. Aglietti, N. Navarathinam, H. Kadhem, T. Salmon, A. Pisseloup, E. Joffre, T. Chabot, I. Retat, R. Axthelm, S. Barraclough, A. Ratcliffe, C. Bernal, F. Chaumette, A. Pollini and W. H. Steyn, "RemoveDEBRIS: An in-orbit active debris removal demonstration mission," Acta Astronautica, vol. 127, pp. 448-463, Oct.-Nov. 2016

[3] L. Pauly, W. Rharbaoui, C. Shneider, A. Rathinam, V. Gaudillière and D. Aouada, "A survey on deep learning-based monocular spacecraft pose estimation: Current state, limitations and prospects," Acta Astronautica, vol. 212, pp. 339-360, Nov. 2023

[4] "Kelvins-Pose Estimation Challenge." [Online]. Available: https://kelvins.esa.int/satellite-pose-estimation-challenge/home/

[5] "ESA commissions world's first space debris removal." [Online]. Available: https://www.esa.int/Safety Security/Clean Space/ESA commissions world s first space debris removal

[6] R. Opromolla, G. Fasano, G. Rufino and M. Grassi, "Uncooperative pose estimation with a LIDAR-based system," Acta Astronautica, vol. 110, pp. 287-297, May-Jun. 2015

[7] M. A. Musallam, V. Gaudilliere, E. Ghorbel, K. A. Ismaeil, M. D. Perez, M. Poucet and D. Aouada, "Spacecraft Recognition Leveraging Knowledge of Space Environment: Simulator, Dataset, Competition Design and Analysis," 2021 IEEE International Conference on Image Processing Challenges (ICIPC), pp. 11-15, Sep. 2021

[8] L. P. Cassinis, R. Fonod and E. Gill, "Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft," Progress in Aerospace Sciences, vol. 110, Oct. 2019

[9] B. Chen, J. Cao, A. Parra and T. Chin, "Satellite Pose Estimation with Deep Landmark Regression and Nonlinear Pose Refinement," 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 2816-2824, Oct. 2019

[10] K. Li, H. Zhang and C. Hu, "Learning-Based Pose Estimation of Non-Cooperative Spacecrafts with Uncertainty Prediction," Aerospace, vol. 9, pp. 592, Oct. 2022

[11] S. Wang, S. Wang, B. Jiao, D. Yang, L. Su, P. Zhai, C. Chen and L. Zhang, "CA-SpaceNet: Counterfactual Analysis for 6D Pose Estimation in Space," 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 10627-10634, Oct. 2022

[12] Z. Wang, Z. Zhang, X. Sun, Z. Li and Q. Yu, "Revisiting Monocular Satellite Pose Estimation With Transformer," IEEE Transactions on Aerospace and Electronic Systems, vol. 58, pp. 4279-4294

[13] T. Phisannupawong, P. Kamsing, P. Torteeka, S. Channumsin, U. Sawangwit, W. Hematulin, T. Jarawan, T. Somjit, S. Yooyen, D. Delahaye and P. Boonsrimuang, "Vision-Based Spacecraft Pose Estimation via a Deep Convolutional Neural Network for Noncooperative Docking Operations," Aerospace, vol. 7, pp. 126, Aug. 2020

[14] S. Sharma and S. D'Amico, "Pose Estimation for Non-Cooperative Rendezvous Using Neural Networks," IEEE Transactions on Aerospace and Electronic Systems, vol.56, pp. 4638-4658, Dec. 2020

[15] P. F. Proença and Y. Gao, "Deep Learning for Spacecraft Pose Estimation from Photorealistic Rendering," 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 6007-6013, May-Aug. 2020

[16] A. Garcia, M. A. Musallam, V. Gaudilliere, E. Ghorbel, K. Al Ismaeil, M. Perez and D. Aouada, "LSPnet: A 2D Localization-oriented Spacecraft Pose Estimation Neural Network," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2048-2056, Jun. 2021

[17] L. Yao, H. She, W. Si, H. Zhou, B. Yang and Z. Xu, "Mobile-SPEEDNet: A Lightweight Network for Non-Cooperative Spacecraft Pose Estimation," 2024 IEEE International Conference on Industrial Technology (ICIT), pp. 1-6, Mar. 2024

[18] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," Information, vol. 11, 2020

[19] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang and J. Yang, "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems, vol.33, pp. 21002-21012, Dec. 2020

[20] C. Zhang, P. Jiang, Q. Hou, Y. Wei, Q. Han, Z. Li and M. Cheng, "Delving Deep Into Label Smoothing," IEEE Transactions on Image Processing, vol. 30, pp. 5984-5996, Jan. 2021

[21] 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, vol. 56, pp. 4083-4098, Oct. 2020

[22] J. Posso, G. Bois and Y. Savaria, "Mobile-URSONet: an Embeddable Neural Network for Onboard Spacecraft Pose Estimation," 2022 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 794-798, May-Jun 2022