IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 14, 2020, accepted November 25, 2020, date of publication November 30, 2020,
date of current version December 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041415

Fast and Accurate Spacecraft Pose Estimation From Single Shot Space Imagery Using Box Reliability and Keypoints Existence Judgments

YURONG HUO[1], **(Member, IEEE), ZHI LI**[2], **AND FENG ZHANG**[3]
[1]College of Aerospace Science and Technology, Space Engineering University, Beijing 101416, China
[2]Science and Technology on Complex Electronic System Simulation Laboratory, Space Engineering University, Beijing 101416, China
[3]No. 32032 of PLA, Beijing 100000, China

Corresponding author: Zhi Li (lz11zys@163.com)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61906213.

ABSTRACT Real-time 6DOF (6 Degree of Freedom) pose estimation of an uncooperative spacecraft is an important part of proximity operations, e.g., space debris removal, spacecraft rendezvous and docking, on-orbit servicing, etc. In this article, a novel efficient deep learning based approach is proposed to estimate the 6DOF pose of uncooperative spacecraft using monocular-vision measurement. Firstly, we introduce a new lightweight YOLO-liked CNN to detect spacecraft and predict 2D locations of the projected keypoints of a prior reconstructed 3D model in real-time. Then, we design two novel models for predicting the bounding box (bbox) reliability scores and the probability of keypoints existence. The two models not only significantly reduce the false positive, but also speed up convergence. Finally, the 6DOF pose is estimated and refined using Perspective-n-Point and geometric optimizer. Results demonstrate that the proposed approach achieves 73.2% average precision and 77.6% average recall for spacecraft detection on the SPEED dataset after only 200 training epochs. For the pose estimation task, the mean rotational error is 0.6812°, and the mean translation error is 0.0320m. The proposed approach achieves competitive pose estimation performance and extreme lightweight ($\sim$ 0.89 million learnable weights in total) on the SPEED dataset while being efficient for real-time applications.

INDEX TERMS Spacecraft pose estimation, 6DOF pose, object detection, monocular vision, space imagery.

I. INTRODUCTION

Spacecraft 6DOF pose estimation is an important part of space proximity operations, e.g., space debris removal, on-orbit servicing, etc. The main solution is to estimate the pose of the spacecraft through monocular cameras, stereo cameras, RGB-D images or data with depth information such as point clouds obtained by LiDAR [1], [2]. Considering the size, mass, power, computation, particular mission scenario, and sustained future costs, monocular sensors can ensure rapid pose determination with lower power, lower hardware complexity and cost, and mass requirements, in contrast to LiDAR and stereo camera sensors. However, since the monocular camera cannot directly measure the relative distance, the calculation complexity is increased, and the monocular camera may be less robust against variable in the lighting conditions of the space environment and the earth background. Therefore, the relative navigation sensor selection is still an open problem. At present, several navigation systems that use satellite-borne monocular camera for close-range operations have been proposed to accomplish fast target tracking and pose estimation [3]–[7]. Traditional pose estimation methods rely on the handcrafted 2D-2D or 2D-3D keypoint and descriptor correspondences [8], [9]. These algorithms are available for objects with sufficient texture, but typically failed when dealing with objects with weakly textured or without texture. To alleviate such problems, most recent approaches began to rely on supervised training with spacecraft pose annotations.

With the success of deep learning in object recognition, deep neural network has been gradually applied to objects' 6D pose estimation. Multiple end-to-end CNN-based neural

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea F. Abate.

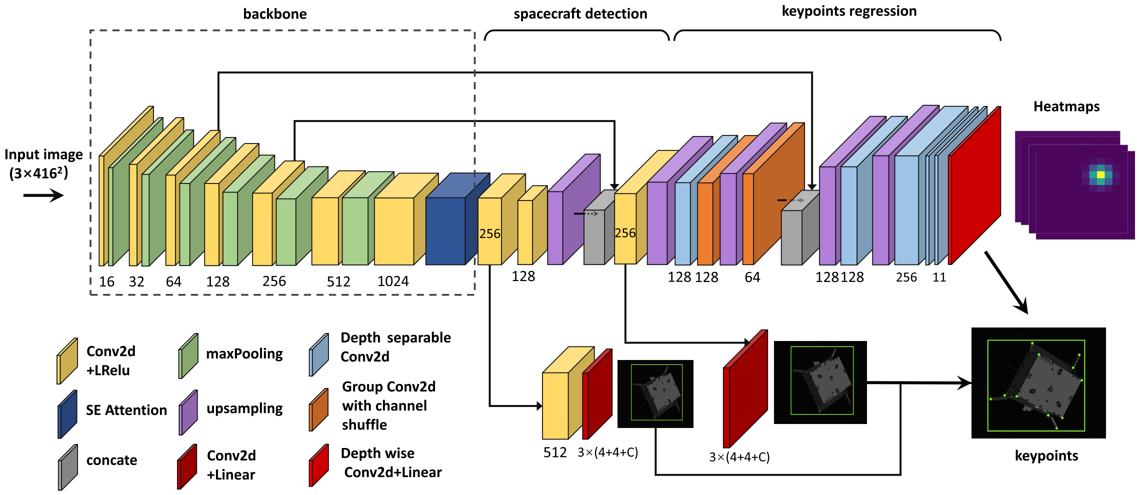


FIGURE 1. The architecture of keypoints regression network in our approach.

networks [10]–[13] have been proposed to map RGB images to 6D poses directly. Although end-to-end poses regression methods are simple, it is not clear whether such end-to-end algorithms have learned enough feature representations for pose estimation. Therefore, the generalization performance of the above models is not robust, and such models are suitable for specified cases. Instead of end-to-end algorithms, the CNN based keypoints regression methods were proposed to solve the problem above. Such methods usually use CNN-based networks to regress 2D-3D keypoints and then use geometric algorithms such as PnP to generate a 6D pose [14]–[18]. However, the above methods are either a large amount of network parameters, or a complicated training method.

In this article, we propose an approach to implement spacecraft 6DOF pose estimation in real-time via the learnable method and geometric algorithm. The key component of the approach is a new and lightweight tiny-YOLOv3 based neural network for predicting the 2D locations of the projected keypoints of the reconstructed 3D model beforehand. The network model contains two sub-nets: spacecraft detection sub-net and keypoints regression sub-net. The proposed network not only achieved encouraging performance, but also the extreme lightweight. And it can be easily modified for multi-tasks.

In this article, we use SPEED datasets [19] to train and test our pose estimation model. SPEED dataset contains more than 10000 space images with the ground truth pose of the "Tango" satellite [20], which provided by the Kelvins Pose Estimation Challenge [21] of ESA. Six "Tango" image examples of SPPED, which have significant variations in view, lighting condition, image background, and object size, as shown in Fig.2. Each image in the SPEED dataset has only one known object (i.e., Tango).

The contributions of this article include the following three aspects. First, we propose a box reliability judgment model and append it to the spacecraft detection sub-net for improving the detection accuracy. Second, we construct a keypoints existence judgment model and add it to the keypoints regression sub-net to accelerate the network convergence and improve the regression accuracy. Third, the spacecraft's 6D pose is estimated by the 2D-3D correspondences produced by keypoints regression network and PnP with RANSAC. And the 6D pose is further refined by geometric optimizer. The keypoints regression network architecture of our approach, as shown in Fig.1.

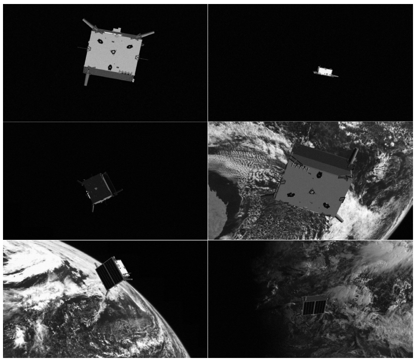


FIGURE 2. Six satellite image examples of SPEED dataset with significant variation in view, lighting condition, background, and object size.

In the rest of the paper, we review the related works and explain our algorithm in details in section II and III, respectively. Section IV and V are experiments and conclusions, respectively.

II. RELATED WORKS

We now review and summarize the existing works on 6D monocular vision-based spacecraft pose estimation.

A. TRADITIONAL METHODS

Most traditional methods use features extracted from the 2D image, feature matching, and Perspective-n-Points (PnP)

[22] to estimate the spacecraft's 6DOF pose. The features can be divided into local keypoints, corners, and edges, etc. The features and descriptors usually produced by handcrafted detectors, e.g., SIFT [23], SURF [24], BRIEF [25], Canny edge detector [26], and Hough Transform [27], etc. Such techniques produce the features correspondences and then use the geometric methods (e.g., PnP, EPnP [28], etc.) to generate the 6DOF pose. Shi *et al.* [29] combine the SIFT and BRIEF methods to extract the target interest points in the image, and the EPnP is used to obtain the initial pose. Rondao and Aouf [30] leverage the FREAK descriptor [31] in combination with the EDLines detector [32] to extract keypoints, corners, and edges to find the correspondence between features, and a EPnP solver is utilized to generate the initial pose. Sharma *et al.* [3] use Weak Gradient Elimination to alleviate the effect of image background on the accuracy of pose estimation and the Sobel operators [33] and the Hough Transform are used to extract the features. Meanwhile, they prove that compared with the PnP method, the EPnP method has better performance in terms of both pose accuracy and runtime. However, due to the small number of image samples used, it is impossible to be sure that this method is robust to any scenario, nor can it be evaluated whether it is suitable for variable illumination conditions in typical space environment. Furthermore, Pesce *et al.* [34] use the GFTT algorithm [35] to extract the feature points of the target 3D model, and adopt a combination of PCA [36] and RANSAC [37] to obtain the correspondences between the target image and the 3D model. Finally, the EPnP solver is used to initialize the pose. Such methods are fast and invariant to perspective, scale and illumination changes, but still sensitive to extreme illumination scenarios, occlusion and scene clutter. They only reliably handle textured objects in high-resolution images [38]. And they will failed in the case of significant variation in light conditions, object size, and image background, i.e., the pose estimation performance relies on the quality of feature correspondences. Thus, it is necessary to explore a new method to solve the above problems. Nevertheless, earlier researches indicate that if we have reasonable 2D-3D keypoints correspondences, the geometric methods are able to estimate the 6D pose well. Therefore, we use the geometric algorithms in our pose estimation approach.

B. MACHINE-LEARNING-BASED METHODS

In recent years, with the success of machine learning especially the deep learning in object recognition and object detection, machine learning has been gradually applied to object 6D pose estimation.

1) NON-NETWORK-BASED METHODS

Shi *et al.* [39] proposed a PCA-based method. The PCA algorithm matches the object from the camera image to a stored matrix of images that has been transformed to its eigenspaces. Although the method can reduce the dimension of the training dataset by the eigenvectors, but this method is only valid under the conditions that each image contains one object only, the object is non-occulted, and the target object is viewed by a weak perspective.

Capuano *et al.* [40] proposed an EKF-SLAM based pose estimation approach that does not require any knowledge of the target. This method uses Harris corner detector [41] to obtain image features, and a single beam LIDAR measurement is also assumed to measure the distance of the extracted features to recover the scale of the reconstructed map. This method estimates its angular velocity from optical flow by EKF (Extended Kalman Filter). However, the attainable performance of this method might worsen when processing images of an orbiting target in unfavorable illumination conditions.

2) CNN-BASED METHODS

CNNs over traditional algorithms is an increase in the robustness for adverse illumination condition, as well as a reduction in the computational complexity [42].

Multiple end-to-end pose regression CNN-based neural networks [10]–[13] have been proposed to map RGB images to 6D poses directly. Although end-to-end poses regression methods are simple, it is not clear whether such end-to-end algorithms have learned enough feature representations for pose estimation. Therefore, the generalization performance of the above models is not robust, and such models are suitable for specified cases. To solve the above problems, several CNN-based keypoints regression methods were proposed.

Instead of handcrafted descriptors, CNN-based keypoints regression methods produce the 2D-3D keypoints by deep learning. Such methods usually use CNN-based networks to regress 2D-3D keypoints and then use geometric algorithms such as PnP to generate a 6D pose [14]–[18]. The object detection model is a critical part of 6D pose estimation in the CNN based keypoints regression methods. Object detection networks such as Faster-RCNN [43] and YOLO [44], etc., are usually used for object detection in the keypoints regression model first, and keypoints regression is performed later. References [14] and [38] perform semantic segmentation on the image and then predict the 3D bounding box of the object. These two methods may not need accurate 3D models. To pursue high prediction accuracy, the common method is to build large deep neural networks. But it will result in slow calculation speed. With the lightweight networks proposed, such as MobileNets [45], ShuffleNets [46], and tiny-yolov3 [47], we can achieve better speed-accuracy trade-off. In our approach, we construct a lightweight network based on tiny-YOLOv3 architecture and the attractive convolution pattern in lightweight CNN networks for predicting the 2D projected locations of 3D keypoints in each image.

In recent years, spacecraft monocular vision-based pose estimation has usually exploited CNN-based techniques. Sharma *et al.* [48] use the AlexNet [49] as the baseline, and the three-dimensional texture model of the space object is also leveraged to calculate the relative pose. Reference [50] proposes a deep learning framework for pose estimation based on orientation soft classification, which allows modeling orientation ambiguity as a mixture of Gaussians. References [51] and [52] simplify the pose estimation problem into a simple regression problem. Reference [51] uses a deep CNN to regress the three-axis stable satellite pose parameters. In [52], a VGG-19 [53] based architecture is used to complete the pose regression task. The Spacecraft Pose Network (SPN) [19], [54] is the seminal work on the SPEED. SPN is constructed based on the region generation network (RPN) of Faster-RCNN and uses a hybrid of classification, regression neural networks and Gauss-Newton iteration method for the pose estimation problem. References [55] and [56] detect the object by regressing a 2D bounding box, and then a separate location regression network is used to predict the 2D locations of the known surface keypoints. The extracted 2D keypoints can be used in conjunction with corresponding 3D model coordinates to compute relative pose via the PnP. Neural Style Transfer (NST) is applied to randomize the texture of the spacecraft in synthetically rendered images in [56]. In [57], a simplified stacked hourglass architecture [15] is constructed for feature detection. A Covariant Efficient Procrustes Perspective-n-Points (CEPPnP) [58] solver combined with an EKF method to enable robust monocular pose estimation for close-proximity operations around an uncooperative spacecraft.

III. METHODOLOGY

Our approach aims to implement spacecraft 6DOF pose estimation in real-time via the learnable method and geometric algorithm. We first reconstruct the spacecraft's 3D wireframe with a few 3D keypoints, which more closely related to the object features from 12 manually multiview images (high image quality and different object orientation). The 3D structure of the spacecraft is solved by the explicit annotation of the 2D locations of the image corresponding to the known 3D keypoints and the triangulation method. And we then construct a new lightweight spacecraft detection sub-net model inspired by tiny-YOLOv3 to recognize the spacecraft and predict the 2D bounding box of the input image. 2D keypoint locations regression is implemented using the input image and predicted 2D bounding box via keypoints regression sub-net later. Lastly, we use the 2D-3D keypoints correspondences, PnP, and a geometric optimization algorithm to estimate and refine spacecraft 6D pose.

A. SPACECRAFT 3D RECONSTRUCTION

Since the SPEED dataset does not provide the spacecraft's 3D model, we need to reconstruct the target's 3D model from the images and poses given in the dataset. And the 3D model will be utilized in subsequent research. It should be noted that this step is in general not required if a 3D model is available. We reconstruct the spacecraft's 3D wireframe model with a few 3D keypoints from 2D monocular vision-based images. We select a small number of 3D keypoints, $\{x_k\}_{k=1}^{M}$, which are more closely related to the object features of the known 3D model prior [20] to depict 3D object model. Note that for satellite, it is best to select these keypoints to be the corner points or endpoints of the satellite solar panels, the corner points of the satellite body, and the endpoints of distinctive antennas, etc. In this article, we select $M = 11$ keypoints to represent satellite, including eight corners, and three antenna endpoints of the satellite. From the training set of the SPEED dataset, we manually selected $N = 12$ images with different object orientation, object size, and high quality, and we carefully marked the corresponding 2D keypoints locations of M 3D keypoints on the N images. Let p_k^i denotes the k^{th} keypoint on the i^{th} image, the 3D keypoints $\{x_k\}_{k=1}^{M}$ could be reconstructed by solving the following objective:

$$min \sum_{i,k} \left\| \boldsymbol{p}_k^i - \boldsymbol{K} \left[\boldsymbol{R}_i | \boldsymbol{t}_i \right] \boldsymbol{x}_k^i \right\|_2^2, \tag{1}$$

where $\boldsymbol{E} = [\boldsymbol{R}_i|\boldsymbol{t}_i] \in \boldsymbol{SE}(3)$, $\boldsymbol{K}$ refers to the camera intrinsic matrix, $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$ are the ground truth rotation matrix and translation matrix of the i^{th} image, respectively. We solve the (1) by CVX solver of MATLAB.[1] 3D keypoints, 3D object wireframe, and the 2D locations of keypoints projected in the image, as shown in Fig.3.

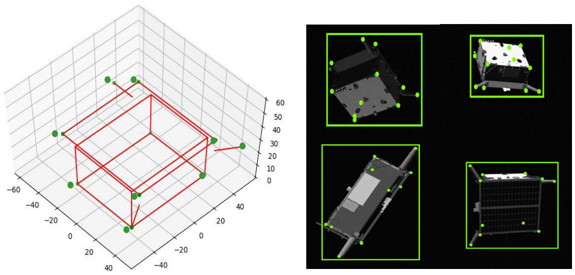


FIGURE 3. 3D reconstruction wireframe with 11 keypoints and 4 examples of ground truth 2D bounding box enclosed 11 2D projected keypoints.

B. SPACECRAFT DETECTION WITH BOX RELIABILITY JUDGMENT

1) BOX RELIABILITY JUDGMENT MODELING

We first establish a spacecraft detection sub-net based on tiny-YOLOv3 to predict the 2D bounding box and category of the object of the input image. Tiny-YOLOv3 attracts a lot of attention for its fast computing speed and therefore, can be applied to mobile devices.

To ensure the speed of calculation and improve the accuracy of space image recognition, we use SE attention mechanism model of SENets [59] after the backbone of tiny-YOLOv3. As mentioned above, the Tiny-YOLOv3 can generate the 2D bounding box (center coordinate (c_x, c_y), and the height h, width w of boxes), the objectness score $Conf_{obj}$ and the category of objects. The $Conf_{obj}$ describes the possibility of containing objects for each grid cell of image. Hence, we can regard $Conf_{obj}$ as the evaluation of the reliability of the central coordinate (c_x, c_y) of the bounding box.

The tiny-YOLOv3 could predict the reliability of the center coordinate and the deterministic coordinate values of bounding box, but it does not predict the reliability of the bounding

[1] The CVX solver is a toolkit of Matlab. It can be downloaded from http://cvxr.com/cvx/.

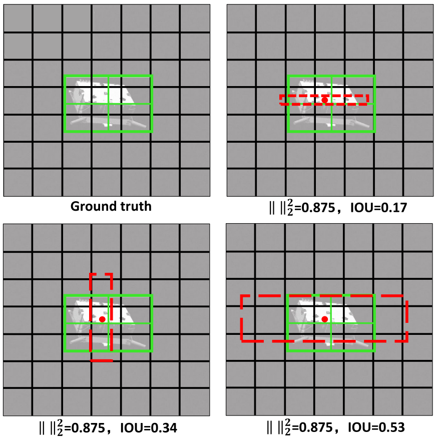

Ground truth
$\| \|_2^2$=0.875, IOU=0.17
$\| \|_2^2$=0.875, IOU=0.34
$\| \|_2^2$=0.875, IOU=0.53

FIGURE 4. Four predicted bounding boxes with the same L2-norm but different IoUs.

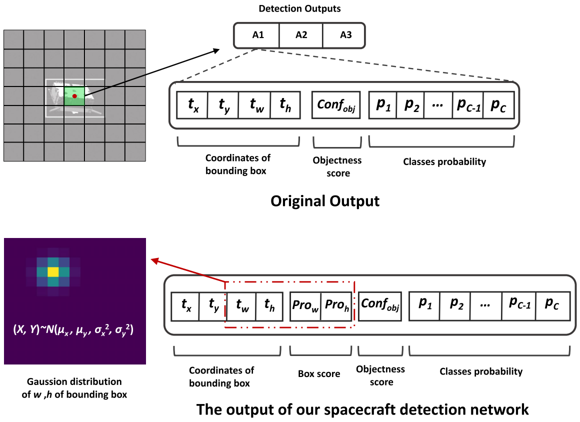

Detection Outputs
A1
A2
A3
t_x t_y t_w t_h
$Conf_{obj}$
p_1 p_2 ... p_{C-1} p_C
Coordinates of bounding box
Objectness score
Classes probability
Original Output
$(X, Y) \sim N(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2)$
t_x t_y t_w t_h Pro_w Pro_h $Conf_{obj}$ p_1 p_2 ... p_{C-1} p_C
Gaussian distribution of w ,h of bounding box
Coordinates of bounding box
Box score
Objectness score
Classes probability
The output of our spacecraft detection network

FIGURE 5. The outputs of the tiny-YOLOv3 and the spacecraft detection network of our approach.

box, i.e., the reliability of w and h, which will lead the network unable to distinguish bounding boxes with different quality. Tiny-YOLOv3 is sensitive to the scale and uses Mean Squared Error (MSE) to regress the coordinates of box. Thus, predicted boxes of different quality may have the same L2-norms (or L1-norms) but different IoUs (Intersection-over-Union). Fig.4 describes that problem (the green rectangle represents ground truth, and the red rectangle represents prediction). In Fig.4, the L2-norms between each predicted bbox and ground truth are the same, but of different IOUs and different quality. Obviously, although L2-norms between three predicted boxes and the ground truth bbox are the same, and the center point of the predicted bbox is also in the same cell, the quality of the three bounding boxes is different. Different predicted boxes with the same box regression loss will result in the inability to train the model well and inaccurate object positioning, and the regression accuracy of 2D keypoints will also be affected.

To solve the above problem, we change the output of the tiny-YOLOv3. The new network output is shown in Fig.5. In addition to the original four box coordinates, objectness score and category, we have increased the reliability score of width w and height h of box, which are denoted by Pro_w and Pro_h, respectively. We model w, h, Pro_w and Pro_h as two-dimensional Gaussian distribution, where (w, h) of bounding box is the mean of Gaussian distribution, and the corresponding reliability score (Pro_w, Pro_h) is the standard deviation. The 2-dimensional Gaussian model will be used to construct the box reliability loss function. By predicting the reliability of w and h, and defining a new regression strategy, the accuracy of the box prediction is improved. Compared with [60], our output is less, but can achieve the same purpose.

2) LOSS FUNCTION OF BOX RELIABILITY

Let p_w and p_h denote the width and height of anchor prior; t_w and t_h denote the width offset and height offset of predicted 2D bounding box, respectively. We define the loss function of box reliability score as (2).

$$\begin{aligned} L_{br} &= \sum_{i}^{W} \sum_{j}^{H} \sum_{k}^{A} \mathbf{1}_k^{truth} \lambda_{i,j,k} \left(truth_{br} - \hat{\boldsymbol{H}} \right)^2, \\ \hat{\boldsymbol{H}} &\sim N \Big(w \left(g_{i,j,k} \right), h \left(g_{i,j,k} \right), Pro_w^2 \left(g_{i,j,k} \right), \\ &\qquad Pro_h^2 \left(g_{i,j,k} \right) | g_{i,j,k} \in G \Big), \end{aligned} \tag{2}$$

where $\hat{\boldsymbol{H}}$ denotes predicted box reliability 2D Gaussian distribution, λ denotes loss weight, and g denotes the grid cell of image grids. We generate the ground truth box reliability score $truth_{br}$ as 2D Gaussian distribution with the means equal to the width and height of the ground truth box, and the standard deviations of 1.0.

Let L_{OriDec} denotes the loss function of original tiny-YOLOv3, the new loss function of our spacecraft detection sub-net is defined as:

$$L_{ObjDec} = \lambda_{OriDec} L_{OriDec} + \lambda_{br} L_{br}. \tag{3}$$

According to the results of multiple trainings, we set λ_{OriDec} and λ_{br} to 1.0 and 10.0.

C. SPACECRAFT KEYPOINTS REGRESSION WITH KEYPOINTS EXISTENCE JUDGMENT

We then construct the keypoints regression sub-net (as shown in Fig.1) to regress the 2D projected location of the 3D keypoints. The end of the keypoints regression sub-net abandoned the fully connected layers, but use convolutional layers to generate predicted heatmaps for each keypoint. We use the sigmoid function to map the value of heatmap to [0, 1], and search for the location of the maximum in the heatmap to obtain predicted coordinates of keypoints, $\left\{ \left(\hat{x_k}, \hat{y_k} \right) \right\}_{k=1}^{M}$. MSE is used to calculate the error of image heatmap:

$$L_{heatmap} = \lambda_{heatmap} \frac{1}{M} \sum_{k=1}^{M} \mathbf{1}_k^{visible} \left(\hat{\boldsymbol{h}}_{p_k} - truth_k^h \right)^2. \tag{4}$$

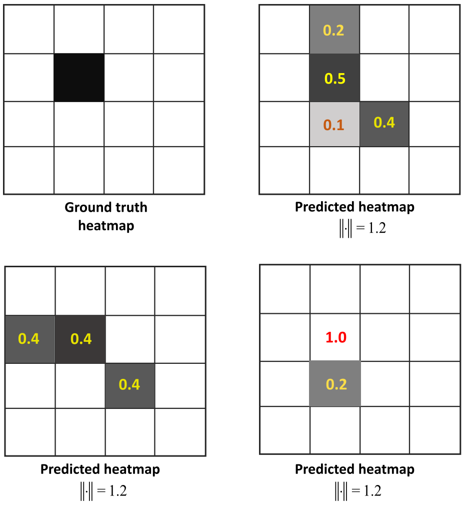


FIGURE 6. The ground truth heatmap and 3 predicted heatmap examples of one keypoint.

In (4), $\lambda_{heatmap}$ refers to loss weight, M denotes the number of keypoints, $\mathbf{1}_k^{visible}$ denotes the ground truth keypoint p_k presents in the image, $\hat{\boldsymbol{h}}_{p_k}$ is predicted heatmap, and $truth_k^h$ denotes the ground truth heatmap of each keypoint. Ground truth heatmap in this article is defined as 2D Gaussian distribution with means equal to coordinates of keypoints and standard deviations equal to 1.0:

$$truth^h \sim N\left(x_k, y_k, 1, 1\right)$$

$$f\left(x, y\right) = exp\left[-log\left(2\right) \cdot \left(\left(x - x_k\right)^2 + \left(y - y_k m\right)^2\right)\right], \quad (5)$$

where (x_k, y_k) is the coordinate of ground truth keypoint p_k. If the keypoints regression network is trained only by minimizing (4), there is a problem that the L2-norms or L1-norms of predicted heatmaps with different qualities and the ground truth heatmap are the same. The three predicted heatmaps with the same L2-norm or L1-norm are shown in Fig.6. The top right and bottom left heatmaps can produce the correct keypoint coordinate, but the bottom right heatmap cannot correctly predict the coordinate of the keypoint.

For solving the above problem, and to improve the accuracy of keypoints regression, accelerate the speed of network convergence, we use logistic regression to assign an existence score to the location with keypoint in the heatmap, i.e., the probability that the grid cell in the heatmap has the keypoint. we derive an effective loss formula based on two expectations: i). the area with keypoints of predicted heatmap $\hat{\boldsymbol{h}}_{p_k}$ should be converged to the area with keypoints of its ground truth $\boldsymbol{h}_{p_k}$. ii). the difference between predicted $\hat{\boldsymbol{h}}_{p_k}$ and its ground truth $\boldsymbol{h}_{p_k}$ at non-keypoints area should converge to zero. Therefore, the keypoints existence loss is formulated below:

$$L_{kp} = \lambda_{kp} \sum_k^M \sum_i^W \sum_j^H \mathbf{1}_k^{vis} \left[\mathbf{1}_{k,i,j}^{co} log\left(\hat{\boldsymbol{h}}_{p_k}\left(g_{k,i,j}\right)\right)\right], \quad (6)$$

$$L_{nokp} = \lambda_{nokp} \sum_k^M \sum_i^W \sum_j^H \mathbf{1}_k^{vis} * \left[\mathbf{1}_{k,i,j}^{noco}\left(1 - log\left(\hat{\boldsymbol{h}}_{p_k}\left(g_{k,i,j}\right)\right)\right)\right]. \quad (7)$$

The λ is the loss weight; W and H are the width and height of heatmap, respectively; $\mathbf{1}_{k,i,j}^{co}$ represents the grid cell $g_{k,i,j}$ has keypoint. The grid cell in ground truth heatmap where the keypoint is located is responsible for determining the value. $\hat{\boldsymbol{h}}_{p_k}\left(g_{k,i,j}\right)$ denotes the value for grid cell $g_{k,i,j}$ in the predicted heatmap of keypoint p_k; $\mathbf{1}_{k,i,j}^{noco}$ represents that there is no keypoint in the grid cell $g_{k,i,j}$. The grid cell in the ground truth heatmap other than the location of the ground truth keypoint is responsible for determining that value.

The coordinates of predicted keypoints, $\left\{\left(\hat{x}_k, \hat{y}_k\right)\right\}_{k=1}^M$, can be generated by searching the locations of the maximum in predicted heatmaps. However, the problem is that the predicted keypoints produced by predicted heatmaps are not differentiable. We cannot optimize the network parameters by minimizing the MSE of predicted keypoints and ground truth. Therefore, we use MSE to estimate the error of the values of the predicted heatmap grid cell where the keypoint locates and the corresponding ground truth, which denoted as L_{co}. And we also estimate the error of the remaining grid cells values in the predicted heatmap and the relevant ground truth, which denoted as L_{noco}. These two losses will help us to improve the accuracy of keypoints prediction and accelerate the convergence speed of the network. L_{co} and L_{noco} are defined as:

$$L_{co} = \lambda_{co} \sum_k^M \sum_i^W \sum_j^H \mathbf{1}_k^{vis} \left[\mathbf{1}_{k,i,j}^{co}\left(\hat{\boldsymbol{h}}_{p_k}\left(g_{k,i,j}\right) - truth_{i,j,k}^h\right)\right]^2, \quad (8)$$

$$L_{noco} = \lambda_{noco} \sum_k^M \sum_i^W \sum_j^H \mathbf{1}_k^{vis} \left[\mathbf{1}_{k,i,j}^{noco}\left(\hat{\boldsymbol{h}}_{p_k}\left(g_{k,i,j}\right) - truth_{i,j,k}^h\right)\right]^2, \quad (9)$$

where $truth_{i,j,k}^h$ denotes the value of grid cell in ground truth heatmap in which the keypoint p_k locates. The loss of keypoints regression network is:

$$L_{KR} = L_{heatmap} + L_{kp} + L_{nokp} + L_{co} + L_{noco}. \quad (10)$$

After several training experiments, we set $\lambda_{heatmap}$ to 50.0, λ_{kp} and λ_{co} to 1.0, and λ_{nokp}, λ_{noco} to 10.0.

We train the model (spacecraft detection sub-net and keypoints regression network) by minimizing the following loss:

$$L = L_{ObjDet} + L_{KR}. \quad (11)$$

Loss function L is defined based on single image. For a mini-batch, the loss is averaged. Our training strategy is "first stage training-later merge training", i.e., we train the spacecraft detection sub-net first and then train with the keypoints regression network.

TABLE 1. The parameters of the camera used to capture the speed.

Parameter	Description	Value
f_x	Horizontal focal length	17.6 mm
f_y	Vertical focal length	17.6 mm
n_u	Number of horizontal pixels	1920
n_v	Number of vertical pixels	1200
d_u	Horizontal pixel length	5.86e-3 mm
d_v	Vertical pixel length	5.86e-3 mm

D. POSE ESTIMATION

Finally, the 6D pose of spacecraft is solved by 2D-3D keypoints correspondences, PnP, and geometric optimizer. Although the proposed network can produce accurate 3D-2D correspondences, it cannot guarantee that all correspondences against to each image are correct. Since the PnP method requires more than three pairs of accurate 3D-2D correspondences, it is better to use RANSAC to improve the robustness of the PnP method when the network detects some 2D keypoints of an image incorrectly (the outliers exist).

First, we use PnP with RANSAC methods[2] to generate the initial value of the transformation matrix $\hat{\boldsymbol{E}}_0 = \left[\hat{\boldsymbol{R}}_0 | \hat{\boldsymbol{t}}_0\right]$ and then perform bundle adjustment to optimize the pose. We optimize the 6D pose $\hat{\boldsymbol{E}}$ by solving the following formula 6D pose:

$$min_{\boldsymbol{E}} \sum_k log\left(cosh\left(\hat{\boldsymbol{p}}_k - \boldsymbol{K}\hat{\boldsymbol{E}} \cdot \boldsymbol{x}_k\right)\right), \tag{12}$$

where $\boldsymbol{K}$ denotes the internal matrix of the camera, $\hat{\boldsymbol{p}}_k$ refers to the estimated 2D keypoints, $\boldsymbol{x}_k$ denotes the coordinate of 3D keypoints of the object in the world coordinate system. (12) is the Log-cosh loss function. We use Levenberg-Marquardt (LM) [61] to solve (12).

IV. EXPERIMENTS

In this section, we evaluate our method for predicting the 6D pose of spacecraft on the SPEED dataset aforementioned. We then compare our method against previous monocular vision-based RGB based state-of-the-art approaches that do not use of depth information for accurate spacecraft 6D pose estimation on SPEED dataset.

A. TRAINING AND TEST DATASET

There are 12000 synthetic images and 5 real images with ground truth pose in the training set, 2998 synthetic images and 300 real images in the test set of the SPEED dataset [19]. The size of each image in the SPEED dataset is 1920 × 1200 px. The parameters of the camera used to capture the SPEED images, as shown in Table 1. The camera parameters determine the internal matrix of the camera, and we ignore the camera distortion in this article.

The labels of the test set in SPEED are not provided, so we cannot conduct the evaluation based on the test set. Therefore, we randomly select 80% of the synthetic images from the training set of SPEED dataset as our training set. And the remaining images, including the 5 real images, are used as the test set. Fig.7 and Fig.8 show the distribution of $\boldsymbol{t}$ and $\boldsymbol{q}$ in our test images, respectively. $\boldsymbol{t}$ denotes the relative position of the target body frame with respect to the camera frame. $\boldsymbol{q}$ represents the quaternion of the target. In Fig.8, the quaternion is parameterized as Euler angles.

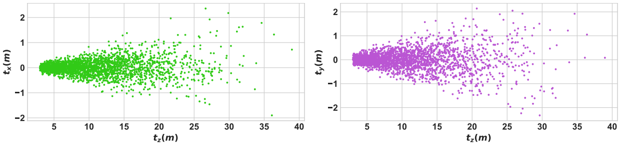


FIGURE 7. The distribution of the relative position, t, in our test images.

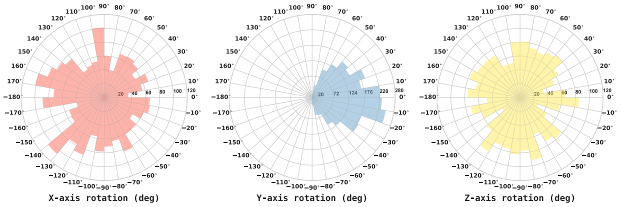


FIGURE 8. The distribution of the relative attitude in our test images.

B. EVALUATION METRICS

The spacecraft (object) detection is reported as the Intersection-Over-Union (IoU) score, which is defined as the intersection area (A_I) divided by the union area (A_U) of the predicted 2D bounding box and the ground truth 2D bounding box:

$$IoU = \frac{A_I}{A_U}. \tag{13}$$

The pose estimation performance is reported as the rotation error ξ_R and the translation error ξ_T [21]. We define the rotation error as the angle of the rotation, that aligns the estimated and ground truth orientations. Let the $\hat{q}$ denotes the rotation quaternion estimation, and q denotes the ground truth of an image. The rotation error ξ_R is defined as:

$$\xi_R = 2 \cdot cos^{-1}\left(|z_r|\right), \tag{14}$$

where z_r denotes real part of $\hat{q} \cdot conj\,(q)$, and $conj\,(\cdot)$ means conjugate.

The translation error for each image is simply the L2-norm of the estimated and ground truth translation vectors. Let $\hat{t}$ and t denote the predicted translation vectors of an image and the ground truth. The translation error ξ_T is defined as:

$$\xi_T = \left\|\hat{t} - t\right\|_2. \tag{15}$$

C. TRAINING DETAILS

Our network is trained on NVIDIA Titan X. The network is optimized by SGD with a moment of 0.9, a weight decay regularization of 0.0001. The batch size is 5 images. Training starts with weights from the backbone of tiny-YOLOv3

[2] We used the routine solvePnPRansac in Python-opencv.

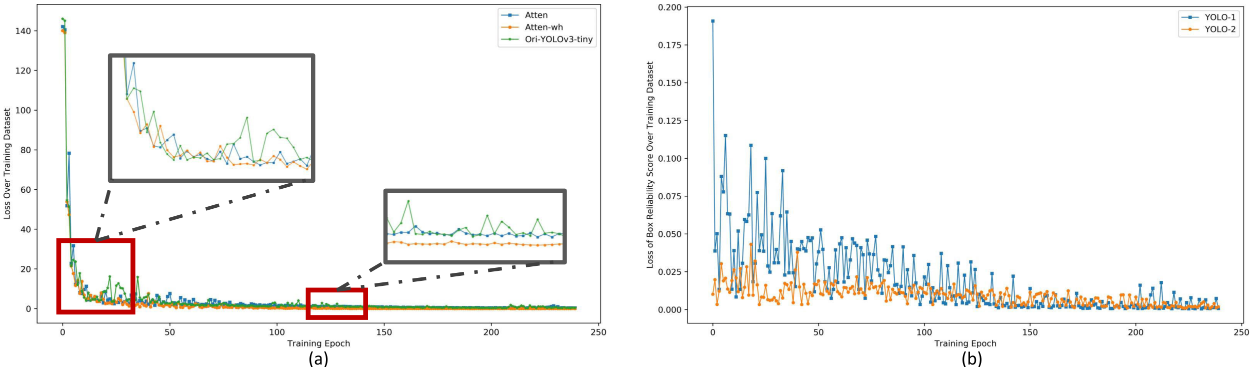


FIGURE 9. The training loss curves vs. training epochs: (a) The training loss of proposed model, model with attention mechanism, model without attention mechanism and box reliability judgment (the three models were named Atten-wh, Atten and Ori-YOLOv3-tiny, respectively in figure); (b) The loss of box reliability of two YOLO layers (YOLO-1, YOLO-2).

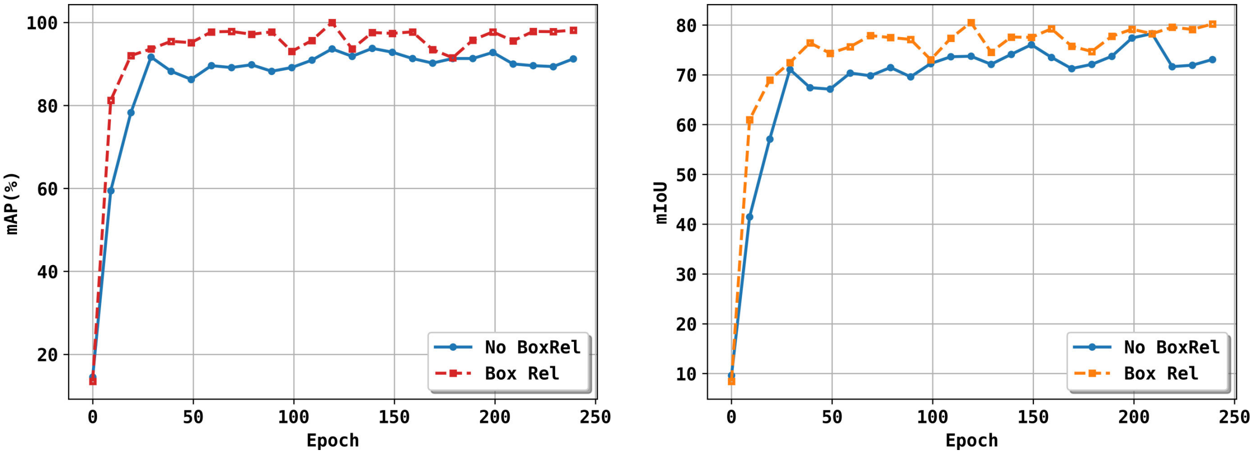


FIGURE 10. The comparison results of mAP and mIoU of Box Rel and No BoxRel models on validation set.

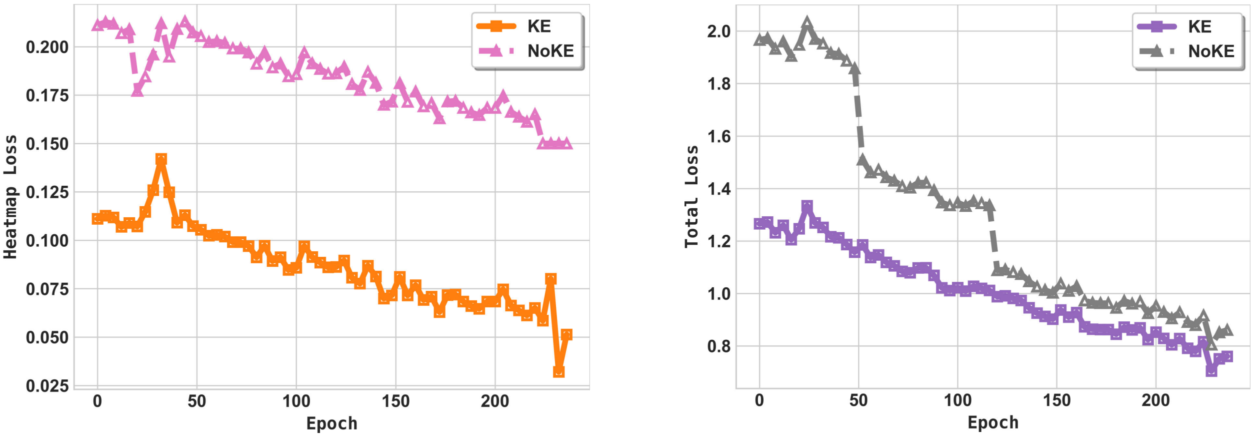
trained on COCO dataset. The initial learning rate set to 0.001 and decayed 2 times every 50 epochs. We use K-means to determine the width and height of six anchor priors (width, height): (30, 47), (42, 88), (55, 59), (73, 105), (103, 169), (172, 254). We construct the proposed model based on the Pytorch architecture.

D. RESULTS

1) POSE ESTIMATION

The training loss using the proposed model converges to 0.015 after 240 epochs. In order to illustrate the advantages of using attention mechanism and the loss of box reliability, we compared the proposed model with a model using only attention mechanism and a model without attention mechanism and box reliability judgment. After 240 iterations, the training loss of the model using only the attention mechanism converged to 0.7, while the training loss of the model without attention mechanism and box reliability judgment converged to 1.315. Training loss of three models above is shown in the Fig.9. From the Fig.9 (a), we find that when the attention mechanism is used, it can converge faster than the model without attention mechanism and box reliability judgment. The output of the keypoints regression is an 11 heatmaps, so each channel of the convolutional layer output is very important for the regression results. By adding a channel attention mechanism, we can control the characteristics of each channel and accelerate the convergence speed of the regression model.

In order to further illustrate the effectiveness of the proposed box reliability judgment model, we randomly selected 10% of the data from the test set as the validation set, and evaluate the detection performance of the spacecraft detection sub-net with and without the box reliability judgement model on the validation set. The validation Mean Average Precision (mAP) and mean IoU (mIoU) in the training process are shown in Fig.10 ("No BoxRel" and "Box Rel" denote the spacecraft detection sub-net with and without the bbox reliability judgement model, respectively). It can be seen from the results that we can improve the accuracy of spacecraft detection and speed up the convergence of the model by exploiting the box reliability judgement model. Since keypoints regression requires the use of bounding box information, it is necessary to increase the reliability of the



(a)

(b)

FIGURE 11. The comparison results of heatmap loss and total loss of KE and NoKE models. (a) The heatmap regression loss; (b) The total loss.

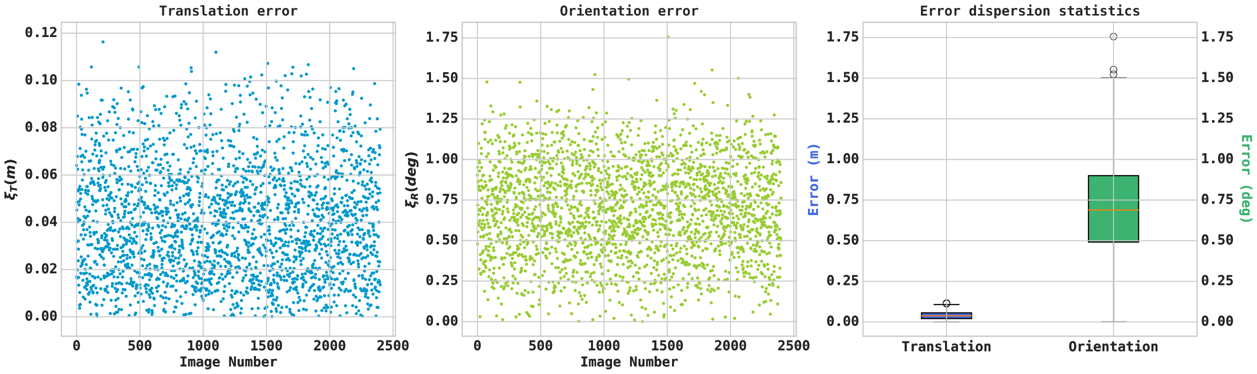


FIGURE 12. The ξ_T (left), ξ_R (middle) and error dispersion (right) on the test set.

bbox to the output of spacecraft detection. The results prove that our proposed model is effective.

And then, we verified the impact of the keypoints existence judgment model on the keypoints regression. First, we established a comparison model, i.e. the keypoints regression sub-net with no keypoints existence judgment model, and named it ''NoKE model''. Then, the keypoints regression sub-net with the keypoints existence judgment model (referred to as the ''KE'') and the NoKE model were trained respectively. The experimental environment and data are the same as those of the training spacecraft detection sub-net. Finally, the heatmap regression loss and the total loss of the keypoints regression sub-net were obtained. The comparison results of heatmap loss and total loss of KE and NoKE models are shown in the Fig.11.

It can be seen from the results that the keypoints regression sub-net using the keypoints existence judgment model can quickly converge. The accuracy of the keypoints regression is also high. For the keypoints regression sub-net that do not use the keypoints existence judgment, the network convergence speed is slow, and the keypoints regression accuracy is low.

Furthermore, we calculated the translation estimation error and orientation estimation error of the proposed network for each image in the test set, and calculated the translation error and rotation error of 1 standard deviation (1σ). The 1σ of translation error in prediction is 0.025m and the 1σ of orientation error is 0.29325°. Fig.12 and Fig.14 show the translation estimation errors, ξ_T, and the orientation estimation errors, ξ_R, on the test set. We also counted the dispersion of the errors as shown in Fig.12. The results show that all ξ_T are below 0.12m, and most of ξ_R are below 1.5 degrees.

The ξ_T in Fig.14 is parameterized as the errors on x-axis, y-axis and z-axis. The x-axis and y-axis are aligned with the image plane axes. The z-axis is aligned with the camera boresight direction. Generally, the x-axis and y-axis of the plane axes point to the right and down along the plane, respectively. The ticks of the x-axis in Fig.14 are the image indices sorted according to the ξ_T and ξ_R. As shown in Fig.14, the proposed method in our paper has good performance in estimating the relative position of x-axis and y-axis, and the errors are below 0.016m, while the estimation errors of the relative position of z-axis are below 0.1m. Moreover, most of

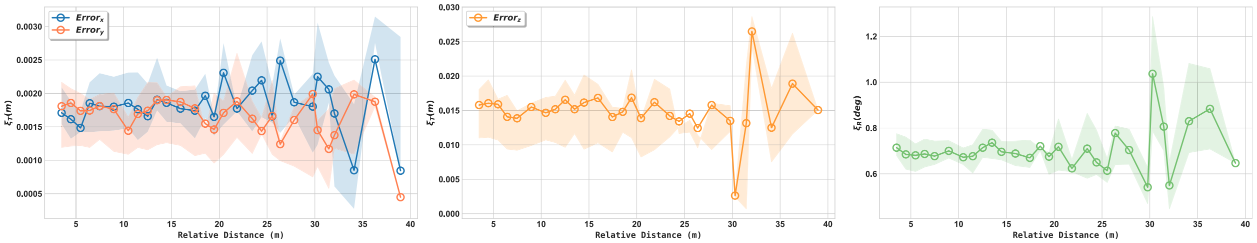


FIGURE 13. The mean ξ_T and mean ξ_R computed against to the mean relative distance. The shaded region shows the 25 and 75 percentile values.

TABLE 2. Pose estimation comparison between the SPN, HRNET-PE, URSONet and the proposed method.

Method	Mean IOU	Median IOU	Mean ξ_R $(degree)$	Median ξ_R $(degree)$	Mean ξ_T (m)	Median ξ_T (m)
SPN [19], [54]	0.8582	0.8908	8.4254	7.0689	0.2937	0.1803
HRNet-PE [55]	0.9534	0.9634	0.7277	0.5214	0.0359	0.0147
URSONet [50]	-	-	3.1036	2.6205	2.1809	1.2718
Proposed	0.9610	0.9727	0.6812	0.5027	0.0320	0.0144

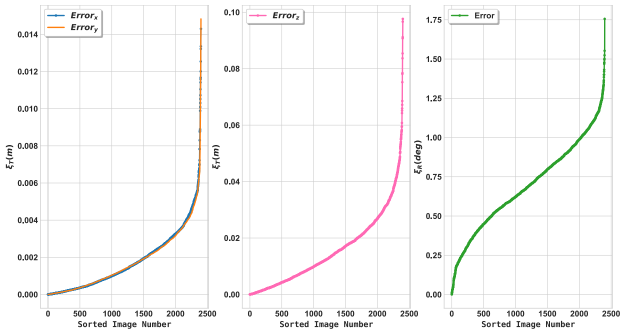


FIGURE 14. The ξ_T and ξ_R for sorted test images.

the orientation errors are mainly concentrated between 0.25 and 1.25 degrees.

Fig.13 shows the mean ξ_T and mean ξ_R computed against to the mean relative distance. According to the curves in the Fig.13, the errors of the z-axis is 10 times the errors in the x-axis and y-axis directions. Since the predicted bbox directly affects the estimation accuracy of relative position in the x-axis and y-axis directions, it turns out that the proposed spacecraft detection sub-net can better locate the position of the target in the image. For most relative distances, the mean ξ_R is between 0.6 and 0.8 degrees. The experimental results above prove that the method proposed in the paper has good performance in the pose estimation of space objects.

In addition, we compared our method against SPN [19], [54], HRNet-PE [55] (named only in this article) and URSONet [50]. Table 2 reports the performance results. Our proposed method achieves competitive performances in both spacecraft detection and pose estimation. The rotational error is smaller than 1°, and the translation error is smaller than 1 meter.

Reference [19], [54] has experimented on the test dataset. Since the [19], [54] did not release the program code, we used the experimental results in the paper for comparison. Reference [55] used a training cross-validation dataset to train proposed network for improving the accuracy of pose estimation. Our training method does not use the training mechanism, but still has competitive performance.

Fig.15 and 16 show the spacecraft detection, keypoints regression and pose estimation results on a sample of the test images. Due to the limitation of the length of the paper, we only show the heatmap of 3 keypoints. In Fig.15, the images are img007510, img001058, img013511, img000068, and img013135 in the training set of the SPEED dataset. In Fig.16, the images we show are img010873, img007926, img012856, img007898, img007343, img007628, img007396, img007517, img007582 in the training set of the SPEED dataset.

2) PERFORMANCE COMPARISON

In order to further evaluate the effectiveness of the keypoints regression network proposed, we used 5 mainstream object detection network such as YOLO-v2 [62], YOLO-v3 [44], SSD [63], RetinaNet [64], Faster R-CNN [43], to replace the spacecraft detection sub-net in the keypoints regression network. The training set and test set described above are leveraged to train the 5 network models, and obtain the predicted 2D bounding boxes and 6DOF pose. Because the difference between the comparison network models and the keypoints regression network proposed is the difference of the spacecraft detection sub-net. Therefore, we only need to evaluate the detection performance to verify effectiveness of our approach. All the models were implemented based on Pytorch framework. The backbones of Feature Pyramid Network (FPN) include ResNet-50 [65], ResNet-101 [65], ResNeXt-101 [66].

Table 3 reports the performance of these state-of-the-art works and proposed model (100 training epochs). Experiment results demonstrate that our proposed network achieves competitive performance with other popular object detection

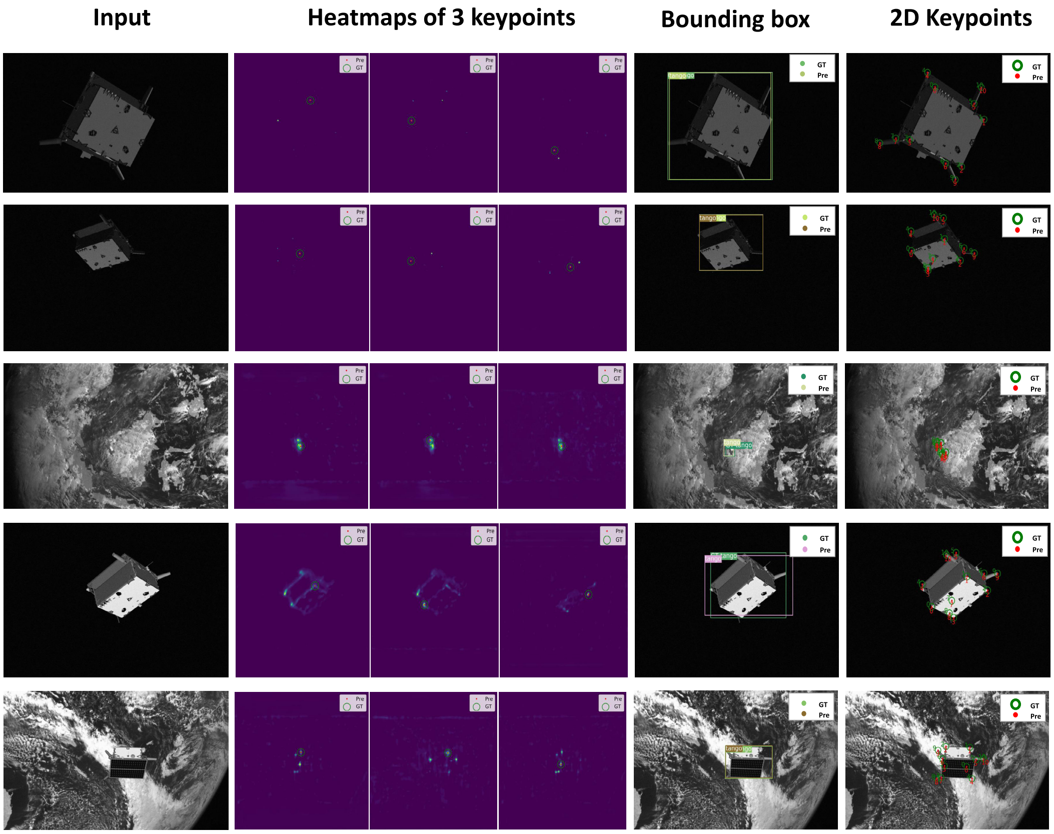


FIGURE 15. Visualization of bounding boxes detection and keypoints regression for SPEED images. The bounding boxes are presented in different colors randomly. (GT denotes ground truth; Pre denotes prediction).

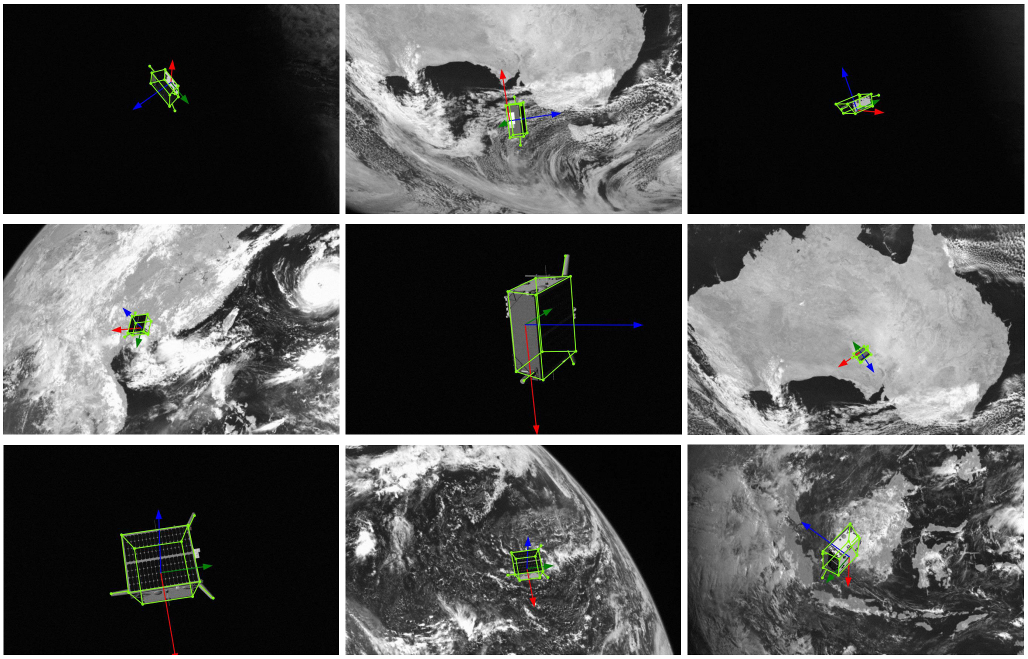


FIGURE 16. Visualization of SPEED images with the predicted poses shown as green wireframes and axes.

frameworks. The following metrics were used in Table 3: average precision (AP), average recall (AR), and parameters amount (Params). If the IoU threshold is γ, the predicted bounding box is regarded as a true positive (TP) only if the IoU between the predicted bounding box and its ground truth is greater than γ; otherwise, it is a false negative (FN).

TABLE 3. Comparison of spacecraft detection performance with other state-of-the-arts.

Dection Model	Backbone	AP	$AP^{.50}$	$AP^{.75}$	AP^{S}	AP^{M}	AP^{L}
YOLO v2	DarkNet-19	60.3	89.3	55.3	61.2	60.7	63.4
YOLO v3	DarkNet-35	71.5	93.2	62.1	69.6	75.0	51.9
SSD@ 300 × 300	VGG-16	68.1	90.0	70.3	71.5	74.6	73.5
RetinaNet	ResNet-50+FPN	67.1	83.3	74.4	57.1	82.3	50.0
	ResNet-101+FPN	68.8	90.6	75.5	80.4	82.5	68.7
	ResNeXt-101+64 ×4d+FPN	69.3	91.6	79.7	81.3	83.9	72.0
Faster R-CNN	ResNet-50+FPN	72.6	94.2	79.0	72.7	84.0	76.1
	ResNet-101+FPN	72.6	94.2	82.0	72.4	84.2	76.4
	ResNeXt-101+64 ×4d+FPN	74.9	94.5	84.8	73.7	85.2	77.6
Proposed	DarkNet-19+Channel Attention	73.2	92.7	83.0	77.3	83.7	74.3

Dection Model	Backbone	AR	AR^{S}	AR^{M}	AR^{L}	$Params(MB)$
YOLO v2	DarkNet-19	64.1	60.0	71.8	66.8	60.58
YOLO v3	DarkNet-35	71.7	69.5	74.3	58.0	102.49
SSD@ 300 × 300	VGG-16	76.1	70.3	82.0	75.9	62.67
RetinaNet	ResNet-50+FPN	76.7	70.8	84.0	69.8	66.45
	ResNet-101+FPN	77.3	73.5	84.3	75.1	99.73
	ResNeXt-101+64 ×4d+FPN	78.0	73.3	84.3	79.0	127.79
Faster R-CNN	ResNet-50+FPN	78.0	73.9	86.1	73.9	78.59
	ResNet-101+FPN	78.5	74.6	86.5	74.0	117.53
	ResNeXt-101+64 ×4d+FPN	80.0	76.2	87.1	77.1	124.38
Proposed	DarkNet-19+Channel Attention	77.6	74.0	84.8	76.6	34.04

The precision, P, and recall, R, at IoU threshold γ are defined below, respectively:

$$P(\gamma) = TP/(TP+FP),$$
$$R(\gamma) = TP/(TP+FN), \tag{16}$$

where FP implies false positive, and $TP + FP$ represents the total number of bounding boxes recognized by the detection sub-net. $TP + FN$ is the total amount of ground truth bounding boxes. Precision refers to the percentage of detected spacecraft instances that are relevant and recall refers to the percentage of total relevant spacecraft instances correctly gathered by the detection sub-net. Since each image in the SPEED dataset corresponds to only one true-value bounding box and only one object, $TP + FN$ is the total number of images in test set.

AP is defined over multiple γ:

$$AP = \frac{1}{|\Upsilon|}\sum_{\gamma\in\Upsilon} P(\gamma), \tag{17}$$

where $\Upsilon = [0.5, 0.55, 0.60, \ldots, 0.95]$ represent a set of different IoU thresholds, and $|\Upsilon|$ denotes the length of Υ. In this article, AR is defined as the average of the recall R over different IoU thresholds, which can be defined as:

$$AR = \frac{1}{|\Upsilon|}\sum_{\gamma\in\Upsilon} R(\gamma). \tag{18}$$

In our experiments, the AP at $\gamma = 0.5$ ($AP^{.50}$) and $\gamma = 0.75$ ($AP^{.50}$) were also calculated and reported.

At the same time, according to the size of the object in the test image, we defined six different indicators about AP and AR, including AP^{S}, AP^{M}, AP^{L}, AR^{S}, AR^{M}, AM^{L}. In this article, according to the area of the ground truth 2D bounding boxes $\Re_{gt}$ in the images, we divided the all test images into small size ($\Re_{gt} < 38^2$), medium size ($38^2 < \Re_{gt} < 101^2$) and large size ($\Re_{gt} > 101^2$). AP^{S}, AR^{S}, AP^{M}, AR^{M}, AP^{L}, AM^{L} represent the AP and AR for small size instances, medium size instances, and large size instances, respectively.

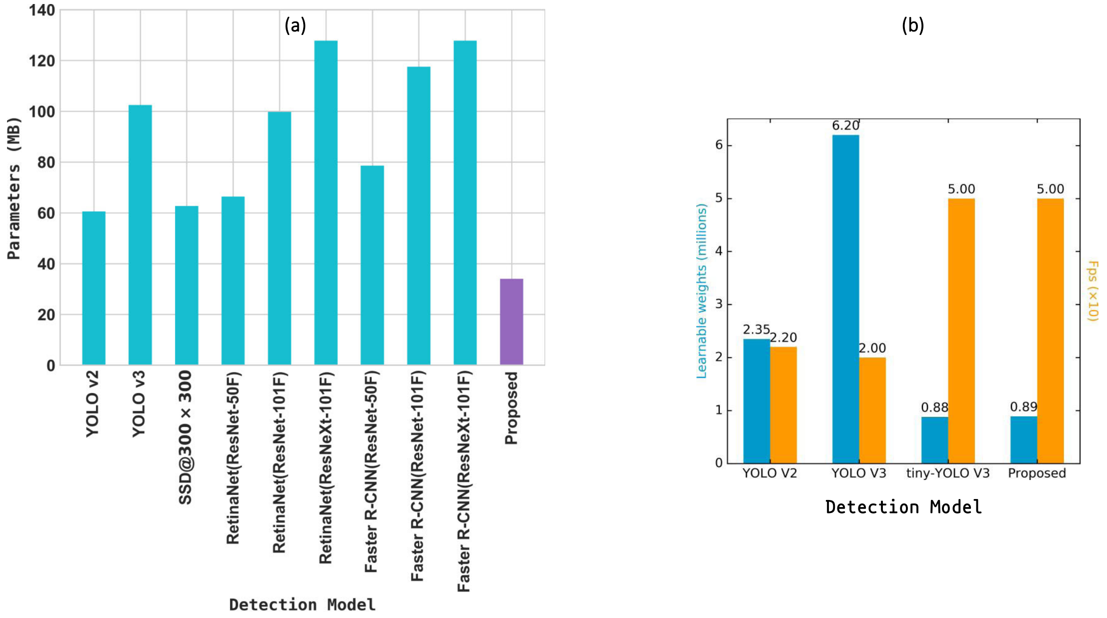
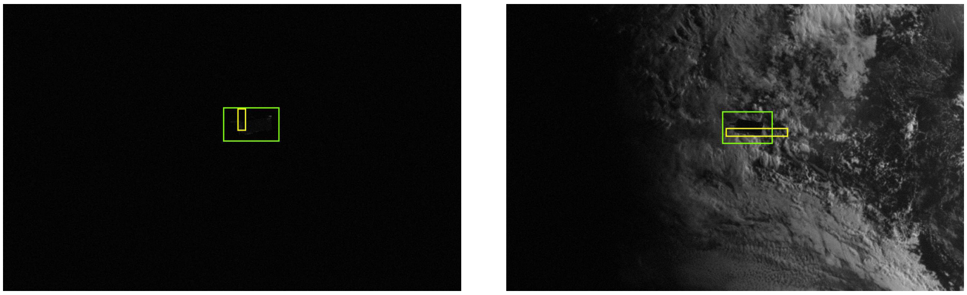


FIGURE 17. The amount of parameters of different backbones and the inference time cost of YOLO series frameworks. (a) The amount of parameters of different backbones. (b) The learnable weights statistic and frame per second (×10) of YOLO series networke.



(a)

(b)

FIGURE 18. Failed cases in spacecraft detection: (a) Poor light condition and small object; (b) Earth background and small object. (Green bounding box denotes ground truth; yellow bounding box denotes wrong box.)

In the experiment, the lightweight design of the network is evaluated as well. Fig.17 illustrates the amount of parameters of different backbones and the inference time cost of YOLO series frameworks based on NVIDIA Titan X. Compared with other frameworks (YOLOv2, YOLOv3, tiny-YOLOv3 described before), our proposed network, with $\sim$ 0.89 million learnable weights in total, only cost $\sim$ 20.0 milliseconds to infer a 1920 × 1200 SPEED image, i.e., the fps (frames per second) is $\sim$ 50. Therefore, even though it performs a little bit inferior than other state-of-the-art frameworks in detection and pose estimation performance, the proposed spacecraft detection sub-net with channel attention and box reliability judgment, and the keypoints regression sub-net with keypoint existence judgment are absolutely a much more suitable solution for space object pose estimation in high-speed and low-cost scenarios.

Although the proposed model can correctly estimate the pose of the object in most images, there are still failed cases. The failure cases are mainly caused by spacecraft detection fails. The two unsuccessful cases are shown in the Fig.18. The object in Fig.18 (a) is too small and the light condition is poor. In Fig.18 (b), the difference in gray level between the earth background and the object is ambiguity. The features extracted by the spacecraft detection sub-net are not sufficient to describe the object, so the detection result is wrong. Therefore, in future work, we will further explore how to improve the accuracy of spacecraft detection in the above scenarios.

V. CONCLUSION

In this article, we propose a novel lightweight tiny-YOLOv3-based framework to estimate the 6DOF pose of a known spacecraft from a single space imagery in real-time. The box reliability judgment model and keypoints existence judgment model are proposed for improving the spacecraft detection and keypoints regression accuracy, and accelerating the network convergence. The spacecraft's 6D pose is estimated by the 2D-3D correspondences produced by proposed keypoints regression network and PnP with RANSAC. And we use the Log-cosh and LM to remove the wrong and inaccurate predictions for pose refinement. Experimental results show that the mean rotational error is 0.6812°, and the mean translation error is 0.0320m. The proposed approach achieves very competitive detection and pose estimation performance, and the proposed network in this article is extreme lightweight ($\sim$ 0.89 million learnable weights in total). Our proposed method is of low-cost and carries slight quantity of learnable weights. It achieves encouraging performance in both pose estimation and real-time capacity on SPEED dataset. Future work should consider how to improve the spacecraft detection accuracy in earth background and extreme poor light condition. And we will study how to reduce the size of the network model while ensuring high keypoints regression accuracy, so that our model is more applicable for on-orbit processing.

REFERENCES

[1] C. English, S. Zhu, C. Smith, S. Ruel, and I. Christie, "Tridar: A hybrid sensor for exploiting the complimentary nature of triangulation and lidar technologies," in *Proc. 8th Int. Symp. Artif. Intell., Robot. Automat. Space*, vol. 1, Sep. 2005, pp. 1–9.

[2] S. Ruel, T. Luu, and A. Berube, "Space shuttle testing of the TriDAR 3D rendezvous and docking sensor," *J. Field Robot.*, vol. 29, no. 4, pp. 535–553, Jul. 2012.

[3] S. Sharma, J. Ventura, and S. D'Amico, "Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous," *J. Spacecraft Rockets*, vol. 55, no. 6, pp. 1414–1429, Nov. 2018.

[4] E. Marchand, F. Chaumette, T. Chabot, K. Kanani, and A. Pollini, "RemoveDebris vision-based navigation preliminary results," in *Proc. 70th Int. Astron. Congr. (IAC)*, Washington, DC, USA, Oct. 2019, pp. 1–10. [Online]. Available: https://hal.inria.fr/hal-02315122

[5] A. A. Grompone, "Vision-based 3D motion estimation for on-orbit proximity satellite tracking and navigation," Naval Postgraduate School, Monterey, CA, USA, Tech. Rep., 2015. [Online]. Available: https://core.ac.uk/download/pdf/36737973.pdf

[6] L. Zhang, H. Yang, S. Zhang, H. Cai, and S. Qian, "Kalman filtering for relative spacecraft attitude and position estimation: A revisit," *J. Guid., Control, Dyn.*, vol. 37, no. 5, pp. 1706–1711, Sep. 2014.

[7] H. Benninghoff, T. Boge, and T. Tzschichholz, "Hardware-in-the-loop rendezvous simulation involving an autonomous guidance, navigation and control system," *Adv. Astron. Sci.*, vol. 145, no. 2012, pp. 953–972, 2012.

[8] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *Int. J. Comput. Vis.*, vol. 66, no. 3, pp. 231–259, Mar. 2006.

[9] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Pose tracking from natural features on mobile phones," in *Proc. 7th IEEE/ACM Int. Symp. Mixed Augmented Reality*, Sep. 2008, pp. 125–134.

[10] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2686–2694.

[11] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1521–1529.

[12] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," 2017, *arXiv:1711.00199*. [Online]. Available: http://arxiv.org/abs/1711.00199

[13] M. Bui, S. Zakharov, S. Albarqouni, S. Ilic, and N. Navab, "When regression meets manifold learning for object recognition and pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–7.

[14] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3828–3836.

[15] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2011–2018.

[16] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 699–715.

[17] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6D object pose and size estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2642–2651.

[18] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4561–4570.

[19] S. Sharma and S. D'Amico, "Pose estimation for non-cooperative rendezvous using neural networks," 2019, *arXiv:1906.09868*. [Online]. Available: http://arxiv.org/abs/1906.09868

[20] S. D'Amico, M. Benn, and J. L. Jorgensen, "Pose estimation of an uncooperative spacecraft from actual space imagery," *Int. J. Space Sci. Eng.*, vol. 2, no. 2, pp. 171–189, 2014.

[21] (2019). *ESA*. [Online]. Available: https://kelvins.esa.int/satellite-pose-estimation-challenge/

[22] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[24] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Computer Vision—ECCV*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 404–417.

[25] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Computer Vision—ECCV*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Germany: Springer, 2010, pp. 778–792.

[26] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[27] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, Jan. 1981.

[28] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the PNP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, p. 155, 2009.

[29] J.-F. Shi, S. Ulrich, and S. Ruel, "Spacecraft pose estimation using a monocular camera," in *Proc. 67th Int. Astron. Congr.*, Guadalajara, Mexico, Jan. 2016, pp. 1–15.

[30] D. Rondao and N. Aouf, "Multi-view monocular pose estimation for spacecraft relative navigation," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2018, p. 2100.

[31] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 510–517.

[32] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, Oct. 2011.

[33] I. Sobel and G. Feldman, "A 3×3 isotropic gradient operator for image processing," *Pattern Classification Scene Anal.*, pp. 271–272, Jan. 1973.

[34] V. Pesce, R. Opromolla, S. Sarno, M. Lavagna, and M. Grassi, "Autonomous relative navigation around uncooperative spacecraft based on a single camera," *Aerosp. Sci. Technol.*, vol. 84, pp. 1070–1080, Jan. 2019.

[35] J. Shi and Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, Jun. 1994, pp. 593–600.

[36] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.

[37] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[38] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 292–301.

[39] J.-F. Shi, S. Ulrich, and S. Ruel, "Spacecraft pose estimation using principal component analysis and a monocular camera," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2017, p. 1034.

[40] V. Capuano, K. Kim, A. Harvard, and S.-J. Chung, "Monocular-based pose determination of uncooperative space objects," *Acta Astronautica*, vol. 166, pp. 493–506, Jan. 2020.

[41] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Procedings Alvey Vis. Conf.*, 1988, p. 5244.

[42] L. Pasqualetto Cassinis, R. Fonod, and E. Gill, "Review of the robustness and applicability of monocular pose estimation systems for relative navigation with an uncooperative spacecraft," *Prog. Aerosp. Sci.*, vol. 110, Oct. 2019, Art. no. 100548.

[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[44] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[45] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

[46] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[47] R. Joseph. (2018). *YOLO Series*. [Online]. Available: https://pjreddie.com/darknet/yolo/

[48] S. Sharma, C. Beierle, and S. D'Amico, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," in *Proc. IEEE Aerosp. Conf.*, Mar. 2018, pp. 1–12.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[50] P. F. Proenca and Y. Gao, "Deep learning for spacecraft pose estimation from photorealistic rendering," 2019, *arXiv:1907.04298*. [Online]. Available: http://arxiv.org/abs/1907.04298

[51] R. Arakawa, Y. Matsushita, T. Hanada, Y. Yoshimura, and S. Nagasaki, "Attitude estimation of space objects using imaging observations and deep learning," in *Proc. AMOS*, 2019, p. 21.

[52] R. Alimo, D. Jeong, and K. Man, "Explainable non-cooperative spacecraft pose estimation using convolutional neural networks," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 2096.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[54] S. Sharma and S. Damico, "Neural network-based pose estimation for noncooperative spacecraft rendezvous," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Jun. 2, 2020, doi: 10.1109/TAES.2020.2999148.

[55] B. Chen, J. Cao, A. Parra, and T.-J. Chin, "Satellite pose estimation with deep landmark regression and nonlinear pose refinement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2816–2824.

[56] T. Ha Park, S. Sharma, and S. D'Amico, "Towards robust learning-based pose estimation of noncooperative spacecraft," 2019, *arXiv:1909.00392*. [Online]. Available: http://arxiv.org/abs/1909.00392

[57] L. P. Cassinis, R. Fonod, E. Gill, I. Ahrns, and J. G. Fernandez, "CNN-based pose estimation system for close-proximity operations around uncooperative spacecraft," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 1457.

[58] L. Ferraz, X. Binefa, and F. Moreno-Noguer, "Leveraging feature uncertainty in the PnP problem," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–13.

[59] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[60] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[61] M. I. A. Lourakis, "A brief description of the Levenberg-Marquardt algorithm implemented by Levmar," *Inst. Comput. Sci., Found. Res. Technol.*, vol. 4, no. 1, pp. 1–6, 2005.

[62] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[63] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision—ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37.

[64] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[66] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

YURONG HUO (Member, IEEE) received the B.Eng. degree in software engineering from Sun Yat-sen University, in 2014, and the master's degree in aeronautical and astronautical science and technology from Space Engineering University, China, where she is currently pursuing the Ph.D. degree. Her research interests include optics curve/image processing, image pattern recognition, and deep learning.

ZHI LI was born in 1973. He received the Ph.D. degree from China Earthquake Administration, in 2003. He is currently a Professor and a Ph.D. Supervisor with Space Engineering University. His main research interests include space system application and SSA.

FENG ZHANG received the master's degree from Space Engineering University (SEU), Beijing, China, in 2018. He is currently an Engineer with No. 32032 of the Chinese People's Liberation Army. His research interests include spatial object characteristics, optics in computing, and spatial situational awareness and control.