Acta Astronautica 238 (2026) 612–629

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research Paper

Hybrid deep learning based monocular pose estimation for autonomous space docking operations

Safinaz Khalil *, Ziwei Wang, Nabil Aouf

Department of Engineering, City St George's, University of London, EC1V 0HB London, United Kingdom

ARTICLE INFO

Keywords:
Deep learning
Hybrid pose estimation
Computer vision
Autonomous docking
Space robotics
Guidance, navigation and control

ABSTRACT

The growing necessity for autonomous space operations has intensified due to the proliferation of on-orbit servicing missions and the critical need to mitigate space debris accumulation, highlighting the essential role of precise and reliable autonomous docking systems. In response to these challenges, this paper presents and validates a novel hybrid methodology for autonomous spacecraft docking that integrates Convolutional Neural Networks (CNNs) with Perspective-n-Point (PnP) algorithms for monocular pose estimation. The proposed hybrid framework synergistically combines CNN-based keypoint detection with PnP geometric reconstruction and RANSAC-based outlier rejection to achieve robust and accurate pose estimation under diverse operational conditions, including variable illumination, viewing geometries, and approach trajectories. A comprehensive evaluation of CNN backbone architectures was conducted using both synthetic and real-world datasets to optimize performance characteristics, encompassing ResNet50, MobileNet, EfficientNet, and HRNet architectures. Experimental validation was performed in a controlled facility utilizing robotic hardware and specialized illumination systems designed to replicate space environmental conditions. The system demonstrated exceptional performance, maintaining translational errors below 0.30% and rotational errors below 1.14° during simulated docking scenarios. Comparative analysis with other direct pose estimation methodologies confirms that the proposed hybrid approach achieves superior translational accuracy while preserving high rotational precision, establishing its viability for autonomous spacecraft operations.

1. Introduction

As On-Orbit Servicing (OOS) operations and Active Debris Removal (ADR) initiatives gain momentum, the demand for precise autonomous space operations has intensified [1,2]. Recent technological advancements have catalyzed the development of the OOS capabilities, marking a pivotal shift in space mission architecture and sustainability [3]. Autonomous systems capable of executing complex maneuvers without human intervention are essential for enabling routine docking operations, particularly given communication delays and limited bandwidth between Earth and orbital assets [4].

While traditional Guidance, Navigation, and Control (GNC) systems have established reliable foundations for space operations, they often struggle with adaptive responses to uncooperative targets or dynamic space environments [5]. The integration of Artificial Intelligence (AI) solutions, particularly Deep Learning (DL), offers promising enhancements to relative navigation capabilities by learning from diverse environmental conditions and adapting to unforeseen scenarios [6]. DL approaches offer particularly compelling advantages for space navigation through their ability to leverage low-cost passive sensors such as monocular and stereo cameras, thereby eliminating the dependency on power-intensive active sensors like LIDAR or radar systems. This passive sensing integration not only reduces power consumption but also minimizes mass and volume requirements while eliminating moving components that could compromise long-term reliability [7,8]. Furthermore, camera-based systems provide rich contextual information that DL algorithms can process to handle challenging conditions, including orbital lighting variations, occlusions, and spacecraft with unknown or altered configurations. As demonstrated by Phisannupawong et al. [9], monocular vision-based navigation systems enhanced by deep learning can achieve centimeter-level positioning accuracy with minimal computational overhead, establishing their viability as alternatives to traditional sensor suites for smaller satellites and extended missions.

While recent end-to-end deep learning approaches in spacecraft pose estimation directly regress relative position and attitude parameters within a single network, such methods present several limitations for safety-critical space applications. AI-based spacecraft navigation systems exhibit significant vulnerability to adversarial attacks that can induce critical navigation errors without being readily detectable by

* Corresponding author.
E-mail addresses: safinaz.khalil.2@citystgeorges.ac.uk (S. Khalil), ziwei.wang.3@citystgeorges.ac.uk (Z. Wang), nabil.aouf@citystgeorges.ac.uk (N. Aouf).

https://doi.org/10.1016/j.actaastro.2025.10.010
Received 7 April 2025; Received in revised form 30 September 2025; Accepted 5 October 2025
Available online 23 October 2025

human operators [10]. Even subtle perturbations to input imagery can cause substantial errors in DL-based pose estimation systems, potentially resulting in mission failures during critical operations such as autonomous docking. Additionally, the black-box nature of end-to-end systems limits interpretability, making it difficult to diagnose failure modes or validate intermediate results during critical docking operations.

This research addresses these limitations by developing a hybrid methodology that combines the interpretability and computational efficiency of classical computer vision algorithms with the adaptability and pattern recognition capabilities of DL. The proposed framework utilizes Convolutional Neural Networks (CNNs) for robust keypoint detection followed by Perspective-n-Point (PnP) algorithms enhanced with RANSAC outlier rejection to determine precise 6-degree-of-freedom (6-DOF) pose estimation in terms of relative position and attitude. The proposed two-stage approach offers several key advantages over end-to-end methods: (1) enhanced training stability through well-established geometric constraints provided by PnP solvers; (2) reduced sensitivity to annotation noise, as keypoint detection tolerates labeling uncertainties better than direct pose regression; and (3) improved interpretability through intermediate keypoint outputs that enable visual verification and failure analysis. The proposed methodology has been comprehensively evaluated using both synthetic datasets generated from high-fidelity International Space Station (ISS) models and real-world experimental data captured under simulated space lighting conditions. The evaluation methodology includes rigorous assessment of various CNN backbone architectures, including ResNet50 [11], MobileNet [12], EfficientNet [13], and HRNet [14], to identify optimal configurations balancing keypoint detection accuracy and computational efficiency. Furthermore, this work introduces a novel "soft dataset" approach that enhances model generalization by selectively curating training examples to emphasize the most informative segments of docking sequences.

The principal contributions of this work are as follows:

- Development of a novel hybrid pose estimation framework that integrates CNN-based keypoint detection with PnP algorithms and RANSAC outlier rejection for robust relative pose estimation in autonomous spacecraft docking scenarios.
- Introduction of a "soft dataset" regularization technique that strategically excludes temporally proximate frames to enhance model generalization capability across diverse docking scenarios and operational conditions.
- Comprehensive evaluation of multiple CNN backbone architectures across varying docking scenarios, illumination conditions, and approach trajectories, establishing quantitative performance benchmarks for autonomous docking systems in space environments.
- Rigorous validation of the proposed system using both synthetic and real-world experimental datasets, demonstrating both practical applicability and robustness under simulated space conditions.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of relevant literature and establishes the theoretical foundation for the proposed approach. Section 3 presents a detailed exposition of the methodology and design specifications of the hybrid pose estimation system. Section 4 presents comprehensive experimental results and performance analysis using both synthetic and real-world datasets. Finally, Section 5 concludes with a discussion of the research implications and identifies promising directions for future investigation.

2. Background and related works

The incorporation of deep learning methodologies in spacecraft docking and refueling applications has yielded substantial advancements, particularly in non-cooperative rendezvous (NCRV) scenarios where accurate and real-time pose estimation is critical for mission success. Existing research spans a spectrum of approaches, ranging from SLAM-based methods that extend traditional navigation pipelines, to direct regression networks that infer pose end-to-end, and two-stage techniques that integrate learned keypoint detection with classical Perspective-n-Point solvers.

2.1. SLAM-based methods

Recent advances in SLAM-based navigation highlight the value of combining complementary sensing modalities. One such approach is presented by Du et al. [15] developed an angles-only navigation algorithm incorporating multisensor data fusion for spacecraft non-cooperative rendezvous operations. Their methodology combines optical measurements with range and range-rate data from ground-based radar systems using a Square-Root Unscented Kalman Filter (SRUKF). The approach addresses practical operational constraints where multiple targets can only be simultaneously tracked by a single radar system. Semi-physical simulation validation confirmed that optical navigation cameras combined with inertial measurement units provide sufficient accuracy for non-cooperative spacecraft rendezvous scenarios.

Building on the theme of radar-assisted navigation but seeking to reduce reliance on multiple stations, Zhang et al. [16] introduced a hybrid real-time maneuver detection scheme that combines Input Detection and Estimation Extended Kalman Filter (IEEKF) with weighted nonlinear least squares methodologies. Their approach utilizes temporal observation series from a single radar station, eliminating the requirement for multiple ground stations and addressing significant practical limitations of existing methods. Simulation results demonstrated robust performance for impulse magnitudes ranging from 1.0 to 100.0 m/s, with particularly strong performance above 5.0 m/s thresholds. For smaller maneuvers, an iterative refinement methodology was developed to enhance maneuver time estimation accuracy.

While radar-based methods demonstrate strong utility, alternative sensing modalities such as LiDAR have also been explored to improve relative motion estimation in non-cooperative scenarios. Kechagias-Stamatis et al. [17] introduced DeepLO, a deep learning-based LiDAR odometry system for spacecraft relative motion estimation that converts 3D point cloud data into 2D depth image representations for CNN-based feature extraction. These 2D projections are subsequently processed by CNN architectures for feature extraction and then fed into Recurrent Neural Networks (RNNs). This hybrid CNN-RNN architecture learns temporal dependencies for pose estimation, demonstrating superior performance compared to traditional Iterative Closest Point (ICP) algorithms, achieving translation errors below 1% of relative range and angular errors averaging 0.29 degrees. DeepLO's multimodal sensor fusion approach integrates LiDAR, Inertial Measurement Unit (IMU), and vision-based sensors to maintain accuracy under challenging space conditions, including debris presence and occlusions. The system achieves real-time performance with 60-millisecond processing latency per LiDAR frame, establishing its suitability for autonomous docking and active debris removal missions. However, DeepLO's generalization capabilities across diverse docking scenarios remains limited without mission-specific reconfiguration.

To address the limitations of traditional LiDAR odometry systems, which rely on loosely coupled sensor fusion and suffer from drift accumulation in large-scale environments, Shan et al. [18] proposed LIO-SAM, a tightly coupled LiDAR Inertial Odometry framework based on factor graph optimization. By integrating IMU preintegration, LiDAR scan-matching, GPS measurements, and loop closure constraints within a unified graph structure, LIO-SAM achieves highly accurate real-time trajectory estimation. The framework supports data playback at rates up to 13x real-time and achieves Root Mean Square Error (RMSE) below 1 m in GPS-referenced evaluations. These performance metrics have been validated across five datasets and three platform configurations, including handheld, ground-based, and marine vehicles. LIO-SAM's modular architecture and precision characteristics establish it as a compelling solution for autonomous navigation in challenging, GPS-denied, or dynamic environments, including space operations.

2.2. Direct regression-based methods

A significant contribution in this domain is ChiNet [19], which employs a Deep Recurrent Convolutional Neural Network (DRCNN) architecture for spacecraft relative pose estimation. The system leverages multimodal data fusion by integrating imagery from both visible spectrum and Long-Wavelength Infrared (LWIR) cameras. This multimodal approach enables robust performance under environmental variability, including illumination fluctuations and conditions that typically degrade the performance of conventional vision-based methods. The architecture synergistically combines Convolutional Neural Networks (CNNs) for spatial feature extraction with Long Short-Term Memory (LSTM) units to capture temporal dependencies in sequential image data. The system processes Red–Green–Blue-Thermal (RGBT) image sequences, incorporating both visible and infrared modalities. Thermal imaging proves particularly advantageous in space docking scenarios where shadows, solar glare, or low-light conditions significantly compromise visible-spectrum sensors. Through LWIR data integration, ChiNet demonstrates high accuracy across diverse environmental contexts.

Building upon the demonstrated effectiveness of temporal dependencies in spacecraft pose estimation, Yang et al. [20] proposed PVSPE, a pyramid vision multitask transformer network that addresses inherent limitations of traditional CNN methodologies in spacecraft pose estimation. The approach combines an enhanced pyramid vision transformer backbone with a specialized feature pyramid network for robust feature extraction and incorporates Matrix Fisher and multivariate Gaussian distributions for comprehensive uncertainty modeling. Experimental validation demonstrated degree-level attitude accuracy and centimeter-level translation precision under challenging illumination conditions. This transformer-based methodology significantly enhances robustness for on-orbit servicing missions. Accurate pose estimation for uncooperative spacecraft remains a critical challenge for autonomous rendezvous and docking.

Proença and Gao [21] addressed this by proposing a deep learning framework trained on URSO, a custom built photorealistic rendering simulator based on Unreal Engine 4, which generates labeled spacecraft imagery under realistic Low Earth Orbit (LEO) conditions. Their approach reformulates orientation estimation as a probabilistic soft classification problem, modeling ambiguity through a Gaussian mixture over discretized Euler angles. Compared to direct quaternion regression, this formulation significantly improves generalization and robustness. Their best model achieved 0.17 m translation error and 4.0° orientation error on the real test set of ESA's pose estimation challenge earning second place among all submissions—and demonstrated strong sim-to-real transferability using just five real images for domain adaptation. Experiments on synthetic and real datasets showed that orientation soft classification outperforms regression by over 5°, and training with simulated camera perturbations and contrast augmentations further reduced orientation error by more than 11.5°. These results highlight the efficacy of combining synthetic data generation with uncertainty aware deep models for robust 6-DOF pose estimation in space based applications. However, the framework's dependency on synthetic datasets restricts its effectiveness when adapting to real mission environment variability.

The research of Duarte et al. [22] introduced a recent study concerning monocular pose estimation systems for autonomous space refueling. They developed a machine learning-based image-driven navigation framework to offer low-cost pose estimation capabilities using single camera setups rather than expensive active sensing systems such as LIDAR. For enhanced prediction accuracy during diverse docking situations scientists trained their CNN with synthetic data derived from high-detail CAD spacecraft models and analysis of shape variations in different lighting conditions. Position errors remained below 1% of relative range with orientation errors under 1 degree during testing which demonstrates compliance with industry docking standards. Their development leverages dropout layers together with data augmentation to extend its generalization capabilities to new conditions which have not been seen before. Through experimentation with a robotic arm in a simulated laboratory setting, they confirmed the model's ability to perform effectively in real time under space docking conditions. Their results show that although theoretical designs demonstrate exceptional performance with synthetic-based training, the effectiveness must still be validated in real-world environments. The inclusion of real mission data into these datasets will help researchers solve the difficult problems presented by space environments.

2.3. Two-stage methods (keypoint detection + PnP)

Standard vision-based docking approaches suffer in Low Earth Orbit (LEO) due to intense lighting variation, reflections, and saturation. Munasinghe et al. [23] addressed these limitations by developing a photometrically accurate LEO simulation testbed and introducing a robust event-based vision pipeline for docking port detection. The setup includes a robotic arm with a satellite mock-up, realistic illumination using a 130 $\mathrm{klm/m^2}$ artificial sun, and Earthshine simulation to replicate orbital lighting. A Dynamic Vision Sensor (DVS) event camera is used to collect asynchronous brightness changes, allowing visual perception under conditions where RGB cameras fail. The proposed detection pipeline accumulates 20,000 events into histograms, applies a CNN-based ring filter, and performs ellipse fitting via RANSAC to estimate the pose of a reflective docking port. The system achieved a mean localization error of 8.58 pixels, with maximum errors up to 39 pixels, corresponding to 2.48% and 3.30% of the image width and height respectively. Notably, this was accomplished even when RGB images exhibited over 30% pixel saturation, highlighting the resilience of event cameras in extreme lighting. Furthermore, the pipeline was trained in under an hour on a mobile GPU using only 20 min of data and generalized well across physical augmentations of the satellite texture. These results demonstrate the potential of event cameras to enable reliable, low-power, and high speed visual sensing for future autonomous satellite docking systems operating in dynamic orbital environments.

The machine vision system for spacecraft docking navigation presented by Chien and Baker [24] analyzes RGB image data for real-time adjustments during docking. The navigation machinery recognized high-contrast geometric features from the target spacecraft which enabled accurate pose determination. The system demonstrated successful docking navigation capabilities because simulated docking scenarios produced position RMS errors below 5 cm and attitude errors under 0.5 degrees. Related studies, including the work by Kisantal et al. demonstrate that high-resolution synthetic datasets can significantly enhance the training of systems for relative pose estimation [25]. The detection-based approach which this system uses can suffer from decreased performance when working in low-light conditions or environments with strong reflections.

The survey conducted by Song et al. [7] gives a good insight about deep learning-based methods for spacecraft relative navigation. The survey includes different deep learning architecture models such as CNNs and RNNs and discusses on the possibility of expanding the accuracy of the pose of estimations and robustness. It brings into focus various training strategies, and the use of virtual and actual environments datasets and the problems related to the deployment of these models in the space. The survey emphasizes the benefits of deep learning in combination with the use other techniques to traditional techniques to complement the deep learning, to increase the reliability of docking and rendezvous spacecraft missions. The survey also reveals the current challenges, successes, trends, and further development of the field, which outlines the further prospective for research.

The work by Kiruki and Asami [26] makes a significant contribution by addressing the challenge of deploying deep learning-based

spacecraft pose estimation algorithms on resource-constrained platforms such as nanosatellites. The authors focus on implementing the inference stage of CNN-based landmark localization directly on Field Programmable Gate Arrays (FPGAs), specifically utilizing the Xilinx Zynq UltraScale+ MPSoC device.

Three different approaches for landmark localization were evaluated: (i) direct regression using a ResNet-50 model, (ii) detection-based heatmap estimation using a U-Net, and (iii) a hybrid detection approach combining spacecraft detection via YOLOv3 with cropped input for landmark detection using ResNet34–U-Net. Results demonstrated that detection-based methods significantly outperform direct regression, with the ResNet34–U-Net achieving an average RMS error of 1.98 pixels compared to 64.5 pixels for regression methods. Furthermore, incorporating spacecraft detection and cropping before landmark localization improved robustness under challenging illumination conditions.

Kiruki and Asami [26] address the challenge of deploying CNN-based spacecraft pose estimation on resource-constrained nanosatellites by proposing an onboard inference framework using a Xilinx Zynq UltraScale+ MPSoC device. Their study evaluates three approaches for landmark localization: (i) direct regression with a ResNet-50 backbone, (ii) heatmap-based detection using U-Net, and (iii) a hybrid pipeline combining spacecraft detection via YOLOv3 with cropped landmark detection using ResNet34–U-Net. Results show that detection-based methods substantially outperform regression approaches, with the ResNet34–U-Net achieving an average RMS error of 1.98 pixels compared to 64.5 pixels for direct regression. Furthermore, preprocessing through spacecraft detection and cropping significantly enhances robustness under challenging illumination. A key contribution is the demonstration that FPGA-based inference with 8-bit quantization achieves comparable accuracy to PC-based floating-point implementations, with an average RMS error difference of less than 0.55. The proposed onboard solution also operates at a low power budget of approximately 3.5 W, confirming its suitability for autonomous, power-limited spacecraft engaged in on-orbit servicing and debris removal missions.

Ma et al. [27] propose GKNet, a graph-based keypoints network for monocular pose estimation of non-cooperative spacecraft. Unlike conventional hybrid methods that treat keypoints as isolated features, GKNet explicitly leverages the geometric constraints of a keypoint graph to reason about spatial relationships. This design enhances robustness against structural symmetry and partial occlusion, two major challenges in spacecraft pose estimation. The architecture employs a dual-branch decoder, consisting of an upsampling-based branch and a graph-convolutional branch, whose outputs are fused to predict accurate keypoint heatmaps.

To support rigorous evaluation, the authors introduce the Spacecraft Keypoints Dataset (SKD), comprising 90,000 simulated images with precise annotations for three different spacecraft models. Experimental results demonstrate that GKNet consistently outperforms state-of-the-art keypoint detectors such as HRNet and ResUNet. For instance, on Satellite 02, GKNet reduced RMSE to 29.1 pixels compared to 74.7 for HRNet, while also improving pose accuracy when combined with a standard PnP solver. Ablation studies further confirm the contribution of the graph-convolutional branch, showing significant degradation when it is removed. These results highlight that incorporating structural context into keypoint detection substantially improves both detection and downstream pose estimation accuracy for non-cooperative spacecraft in challenging orbital conditions.

Chen et al. [28] propose a monocular pose estimation framework that combines deep landmark regression with nonlinear pose refinement for space-borne satellites. Their approach begins by reconstructing a sparse 3D model of the target spacecraft through multi-view triangulation, selecting 11 visually distinctive landmarks such as corners and antenna endpoints. A deep network based on High-Resolution Net (HRNet) is then trained to regress the 2D image coordinates of these predefined landmarks from bounding-box-cropped satellite images. By maintaining high-resolution representations, HRNet achieves superior landmark localization accuracy compared to lower-resolution backbones.

The predicted 2D landmarks are associated with their 3D counterparts, and a Perspective-n-Point (PnP) solver followed by a robust nonlinear least-squares optimization refines the estimated pose. To further enhance robustness, the authors introduce a Simulated Annealing–Levenberg–Marquardt Pose Estimator (SA-LMPE), which adaptively removes outlier correspondences during optimization. Evaluated on the SPEED dataset from the Kelvins Pose Estimation Challenge (KPEC), their method achieved a cross-validation orientation error of 0.73° and a translation error of 0.036 m, ranking first in the competition with an overall score of 0.0094. This work demonstrates that combining deep landmark regression with geometric optimization provides state-of-the-art accuracy for spacecraft pose estimation, significantly outperforming prior methods such as the Spacecraft Pose Network (SPN).

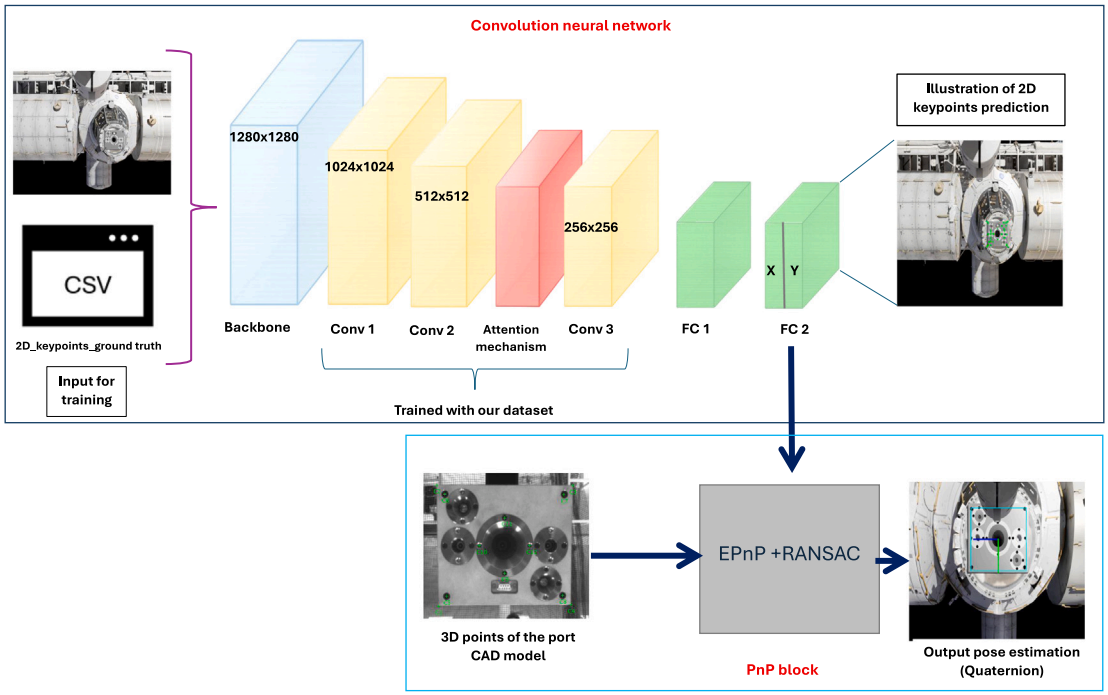
2.4. Critical analysis of related works

While SLAM-based methods (e.g., Du et al. [15], Zhang et al. [16], Kechagias-Stamatis et al. [17], Shan et al. [18]) demonstrate strong multisensor fusion capabilities, they rely heavily on radar or LiDAR inputs and ground-based infrastructure, which limits scalability for purely onboard, vision-based navigation in deep space. Direct regression approaches such as ChiNet [19], PVSPE [20], or synthetic-data driven frameworks like Proença and Gao [21] and Duarte et al. [29] offer end-to-end learning but suffer from limited interpretability, sensitivity to label noise, and poor generalization across illumination and background variations critical factors in docking scenarios. Two-stage pipelines, including event-based docking [23] or RGB feature extraction [24], improve robustness but remain tailored to specific sensor modalities, making them less versatile for passive monocular systems. FPGA-based studies (Kiruki and Asami [26]) address onboard efficiency but do not explicitly handle temporal redundancy or generalization across docking sequences. Recent keypoint driven architectures such as HRNet-based landmark regression [28] or graph based networks like GKNet [27] achieve high accuracy on benchmark datasets, but often assume cooperative targets, high quality synthetic training, or extensive landmark visibility, which does not reflect operational constraints in low-light or cluttered orbital conditions.

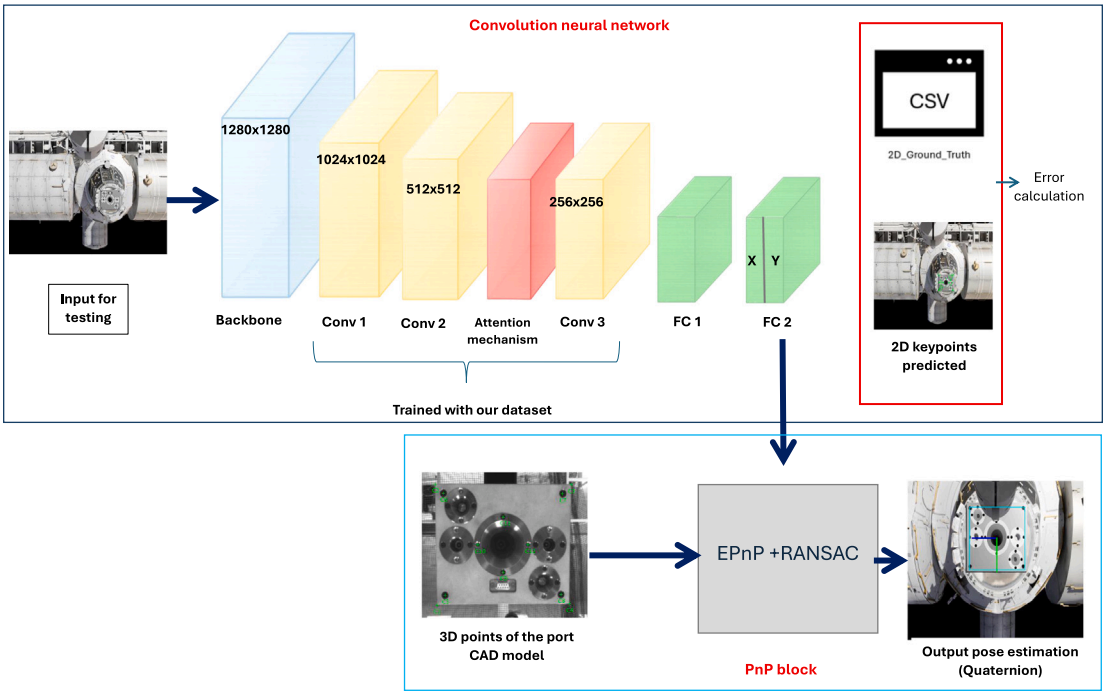
In contrast, our proposed methodology deliberately integrates CNN-based keypoint detection with geometric PnP [30] and RANSAC [31], ensuring interpretability, robustness to annotation noise, and geometric consistency. The introduction of the soft dataset addresses temporal redundancy and overfitting issues largely ignored in prior works thereby enhancing generalization across both synthetic and real-world docking sequences. Furthermore, by validating across multiple lightweight backbones (ResNet [11], MobileNet [12,32], EfficientNet [13], HRNet [14]) and deploying on space grade hardware [33,34], our framework balances accuracy, efficiency, and reliability, offering a more practical solution for autonomous docking than existing SLAM-based, regression based, or sensor-specific methods.

3. Methodology

The proposed pose estimation methodology employed in this work utilizes an indirect, hybrid approach: keypoint features are first detected in 2D imagery through a deep learning network, followed by recovery of the ISS's 6-DOF pose through robust PnP problem formulation. Although the ISS is traditionally classified as a cooperative spacecraft equipped with retroreflectors, GPS transponders, and docking target fiducials to assist chaser vehicle navigation [35], our proposed monocular pipeline operates without active beacons or artificial markers. In this case, the ISS is treated as a non-cooperative target, where pose estimation must be inferred exclusively from passive imagery



(a) Model training pipeline



(b) Model testing pipeline

Fig. 1. Overall workflow of the proposed CNN-based pose estimation framework.

under complex Earth-background clutter and unfiltered solar illumination conditions. As noted by Shi et al. [36], logistical vehicles can only approach the ISS through a constrained zenith-ward trajectory, resulting in the target spacecraft being consistently observed against Earth's backdrop, thereby creating particularly challenging conditions for non-cooperative operations.

The proposed framework integrates PnP algorithms with Convolutional Neural Network (CNN) architectures to process the target's

datasets. The CNN predicts 2D keypoint locations within RGB imagery while simultaneously extracting spatial features critical for high-precision pose estimation. Based on these extracted features, the PnP algorithm computes the spatial position and orientation of the ISS, which is essential for autonomous docking operations.

The subsequent sections provide detailed exposition of the CNN architecture design rationale for 2D keypoint prediction, along with comprehensive analysis of various backbone architectures that enhance the robustness and accuracy of the complete pose estimation pipeline.

3.1. Methodology overview

A tailored CNN-based architecture (Fig. 1) is developed for accurate and efficient 2D keypoints prediction in RGB imagery. This architecture addresses the computational efficiency and environmental robustness requirements critical for space applications. The network processes RGB inputs through a hierarchical series of convolutional layers that progressively capture and refine spatial features relevant to autonomous docking operations. The initial convolutional layers focus on detecting low-level patterns that establish a foundational representation of the input imagery. These foundational layers enable the network to consistently identify critical structural elements within the target docking region, facilitating robust feature learning in subsequent network stages.

Intermediate convolutional layers capture spatial relationships essential for precise keypoint localization as data propagates through the network hierarchy. These layers are designed to learn mid-level spatial patterns, including corners, junctions, and other docking-specific landmarks. Given the high-precision requirements of docking scenarios, this processing stage is particularly critical, as minor errors in keypoint prediction can propagate into significant pose estimation inaccuracies. To enhance training stability and model generalization, batch normalization is applied following each convolutional layer to standardize input distributions and maintain gradient flow throughout the architecture. ReLU activations introduce non-linearity, enabling the network to learn complex spatial relationships within the data.

The architecture subsequently employs deeper convolutional layers specialized for extracting higher-level, abstract features. These layers operate with expanded receptive fields and integrate broader spatial contexts, enabling the network to distinguish between critical docking landmarks and irrelevant background structures. These deeper layers are critical for maintaining model robustness in space environments characterized by various lighting conditions, specular reflections, and dynamic shadow patterns. Furthermore, the deeper architecture components are optimized to reduce model sensitivity to noise and minor input image variations, ensuring consistent keypoint detection performance.

Experimentally, an attention mechanism was integrated into the network to direct model focus toward relevant regions within input imagery. This mechanism enables selective attention to keypoint-containing areas while suppressing irrelevant background details, particularly beneficial in scenarios involving distracting backgrounds or noisy environments. Specifically, we instantiate this mechanism as a Squeeze-and-Excitation (SE) channel-attention module placed immediately before the final convolution of the prediction head. Let the backbone output be $\mathbf{X}^{(0)} \in \mathbb{R}^{B\times 1280\times H_0\times W_0}$; two subsequent convolutions, $\mathrm{Conv}_{1280\to 1024}$ and $\mathrm{Conv}_{1024\to 512}$, yield $\mathbf{X} \in \mathbb{R}^{B\times 512\times H\times W}$.

$$\mathbf{X}^{(0)} \in \mathbb{R}^{B\times 1280\times H_0\times W_0}. \tag{1}$$

$$\mathbf{X} = \mathrm{Conv}_{1024\to 512}\big(\mathrm{Conv}_{1280\to 1024}(\mathbf{X}^{(0)})\big) \in \mathbb{R}^{B\times 512\times H\times W}. \tag{2}$$

The SE block first compresses the spatial dimensions by global average pooling to produce a 2-D channel descriptor:

$$\mathbf{z} = \mathrm{GAP}_{(H,W)}(\mathbf{X}) \in \mathbb{R}^{B\times 512}, \tag{3}$$

$$z_{b,c} = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} X_{b,c,h,w}. \tag{4}$$

This 4-D→2-D reduction preserves channel semantics while removing spatial variability solely for the attention computation. Channel gates are then computed with a two-layer MLP (reduction ratio r) using a ReLU nonlinearity,

$$\mathbf{s} = \sigma\big(\mathbf{W}_2\,\mathrm{ReLU}(\mathbf{W}_1\,\mathbf{z})\big) \in \mathbb{R}^{B\times 512}, \tag{5}$$

where $\sigma(\cdot)$ denotes the sigmoid, $\mathbf{W}_1 \in \mathbb{R}^{\frac{512}{r}\times 512}$ and $\mathbf{W}_2 \in \mathbb{R}^{512\times\frac{512}{r}}$. The resulting 2-D gate vector $\mathbf{s}$ is then broadcast back over (H, W) and applied channel-wise to the original 4-D tensor:

$$\tilde{\mathbf{X}}_{b,c,h,w} = s_{b,c}\,X_{b,c,h,w}, \tag{6}$$

thereby restoring the 4-D shape while reweighting channels uniformly across spatial locations. This preserves spatial topology but amplifies keypoint-informative responses and attenuates distractors before the final convolution $\mathrm{Conv}_{512\to 256}$.

Following feature extraction, the refined feature maps are flattened and processed through fully-connected layers responsible for 2D keypoint coordinate predictions. These layers map the high-level spatial information extracted by convolutional stages to precise 2D keypoint locations (x, y coordinates). The architecture is designed to ensure that these fully connected layers efficiently translate spatial relationships into accurate keypoint predictions, providing structured input for subsequent pose estimation phases. The 3D pose of the ISS relative to the camera coordinate system is computed using the PnP algorithm, with the predicted keypoints serving as its input parameters.

The CNN's structure, illustrated in Fig. 1, employs a balanced approach between depth and computational efficiency to ensure effective computation. This architecture establishes a robust AI-based framework for high-accuracy pose estimation in space docking applications through the strategic integration of convolutional layers optimized for spatial relationship extraction, attention mechanisms for selective feature focus, and fully connected layers trained for precise 2D keypoint regression.

We note that in the synthetic dataset generation, the ECI frame was set to coincide with the virtual camera frame in Blender, which simplifies the transformation chain. This assumption was only applied in simulation and does not affect the real-world experiments, where the complete frame mapping is preserved.

3.2. RANSAC based PnP algorithm for pose estimation

Once the CNN has detected 2D keypoints on the target (e.g., the ISS), each detected pixel coordinate [37]:

$$x_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} \in \mathbb{R}^2 \tag{7}$$

is associated with a known 3D landmark in the target's coordinate frame.

$$X_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \in \mathbb{R}^3 \tag{8}$$

To recover the camera's pose relative to the target, we solve the Perspective-n-Point problem [38]. Throughout, let n denote the total number of detected 2D–3D correspondences; thus, for $i = 1, \ldots, n$, we have (X_i, x_i).

The PnP problem seeks a rotation matrix $R \in \mathbb{R}^{3\times 3}$ (satisfying orthonormality and unit-determinant, Eq. (11)), and a translation $t \in \mathbb{R}^3$ that minimize the reprojection error:

$$E_{\mathrm{reproj}}(R,t) = \sum_{i=1}^{n} \left\| x_i - \pi\big(K\,[\,R\,X_i + t\,]\big) \right\|_2^2 \tag{9}$$

Here, $K \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix (see Eq. (14)); $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ denotes the perspective-division mapping

$$\pi\left([X_c,\, Y_c,\, Z_c]^\top\right) = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix}, \quad Z_c > 0 \tag{10}$$

and $R\,X_i + t$ transforms the 3D landmark X_i from the object frame into the camera frame. The vector $x_i = [\,u_i, v_i\,]^\top$ is the observed 2D pixel coordinate. At least four non-coplanar correspondences are required to solve for the six degrees of freedom in (R, t) [30,39].

$$R\,R^\top = I_3, \quad \det(R) = 1 \tag{11}$$

In practice, we first compute a closed-form estimate $\left(R_{\text{init}}, t_{\text{init}}\right)$ using the EPnP solver [30], and then refine (R, t) by minimizing E_{reproj} over all n points via Levenberg–Marquardt [38].

The rotation matrix R is parametrized by a 3-vector $r = [\,r_x, r_y, r_z\,]^\top$ (Rodrigues parameters [40]). Specifically,

$$R(r) = \exp\left([r]_\times\right) = I_3 + \frac{[r]_\times}{\|r\|}\,\sin\|r\| + \frac{[r]_\times^2}{\|r\|^2}\left(1 - \cos\|r\|\right) \tag{12}$$

In Eq. (12), $\|r\| = \sqrt{r_x^2 + r_y^2 + r_z^2}$ is the rotation angle in radians, and

$$[r]_\times = \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} \tag{13}$$

is the skew-symmetric matrix corresponding to r. After obtaining R_{init} from EPnP, we initialize the nonlinear stage using the inverse Rodrigues transform: $r_{\text{init}} = \text{Rodrigues}^{-1}(R_{\text{init}})$.

Assuming zero skew and square pixels, the camera intrinsic matrix [37] takes the form

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

In Eq. (14), f_x and f_y are the focal lengths (in pixels) along the x- and y-axes, respectively; c_x and c_y denote the principal point coordinates, typically near the image center. The product $K\left[R\,X_i + t\right]$ yields the camera-frame coordinates $\left[X_{c,i},\, Y_{c,i},\, Z_{c,i}\right]^\top$ before projection.

During both the EPnP initialization and the full nonlinear refinement [30], any 3D point X_i projects to the image plane as

$$\hat{x}_i = \pi\left(K\,[\,R(r)\,X_i + t\,]\right) \tag{15}$$

In Eq. (15), $\hat{x}_i = [\,\hat{u}_i, \hat{v}_i\,]^\top \in \mathbb{R}^2$ is the predicted 2D projection in pixel coordinates, and $R(r)\,X_i + t = [\,X_{c,i},\, Y_{c,i},\, Z_{c,i}\,]^\top$ are the coordinates of X_i in the camera frame. The operator π (See Eq. (10)) is simply the homogeneous-division mapping that takes a 3D point in camera coordinates and returns its 2D pixel projection.

To handle outliers in the CNN-detected correspondences, we embed EPnP within a RANSAC loop [31]. Let τ be the inlier threshold (measured in pixels) and $N_{\max}$ be the maximum number of RANSAC iterations (e.g., 1000). At each iteration $j = 1, \ldots, N_{\max}$, four correspondences $\{(X_{i_k}, x_{i_k})\}_{k=1}^4$ are randomly selected, ensuring when possible that $\{X_{i_k}\}$ are not coplanar. EPnP is then applied to these four pairs to obtain $\left(R_{\text{init}}^{(j)}, t_{\text{init}}^{(j)}\right)$, and $r_{\text{init}}^{(j)} = \text{Rodrigues}^{-1}\left(R_{\text{init}}^{(j)}\right)$. Starting from $\left(r_{\text{init}}^{(j)}, t_{\text{init}}^{(j)}\right)$, Levenberg–Marquardt [41] is run on those four points to produce $\left(r^{(j)}, t^{(j)}\right)$. For each correspondence $i = 1, \ldots, n$, the predicted projection is

$$\hat{x}_i^{(j)} = \pi\left(K\,[\,R(r^{(j)})\,X_i + t^{(j)}\,]\right), \quad e_i^{(j)} = \left\| x_i - \hat{x}_i^{(j)} \right\|_2 \tag{16}$$

Scaramuzza et al. [42]

A correspondence i is classified as an inlier if $e_i^{(j)} < \tau$. Let $S^{(j)}$ denote the set of all inliers in iteration j, with cardinality $|S^{(j)}|$. If $|S^{(j)}|$ exceeds the current maximum, the iteration's pose parameters are recorded as

$$r_{\text{best}} = r^{(j)}, \quad t_{\text{best}} = t^{(j)}, \quad S_{\text{best}} = S^{(j)} \tag{17}$$

After $N_{\max}$ iterations, a final Levenberg–Marquardt optimization is performed over all correspondences in S_{best} to minimize

$$\sum_{i \in S_{\text{best}}} \left\| x_i - \pi\left(K\,[\,R(r)\,X_i + t\,]\right) \right\|_2^2 \tag{18}$$

yielding the final pose $\left(r_{\text{est}}, t_{\text{est}}\right)$.

In our implementation, the inlier threshold τ (measured in pixels) was empirically selected by analyzing the reprojection error (see Eq. (9)) distribution on a held-out calibration set comprising both synthetic and real images; we chose $\tau = 4$ px to represent approximately two standard deviations of the keypoint localization error distribution observed during validation [38]. The maximum iteration count $N_{\max} = 1000$ was chosen to ensure 99.9% confidence of finding a consensus set with at least 70% inliers, following the standard RANSAC failure-probability formula [31]. These parameter choices consistently delivered accurate pose estimates across a wide range of test conditions, underscoring the reliability and robustness of our PnP+RANSAC pipeline even in the presence of moderate keypoint noise.

In Eq. (18), r_{best} and t_{best} are the pose parameters from the iteration with the largest inlier set, and S_{best} is the corresponding set of inlier indices. The resulting $\left(r_{\text{est}}, t_{\text{est}}\right)$ minimizes the reprojection cost over all inliers in S_{best}.

After obtaining $\left(r_{\text{est}}, t_{\text{est}}\right)$, we compare it to the ground-truth pose $\left(r_{\text{gt}}, t_{\text{gt}}\right)$, provided by simulator logs or a motion-capture system, using two error metrics. First, the normalized position error is defined as [43]

$$\delta t_r = \frac{\left\| t_{\text{est}} - t_{\text{gt}} \right\|_2}{\left\| t_{\text{gt}} \right\|_2} \tag{19}$$

In Eq. (19), t_{est} and t_{gt} are the estimated and ground-truth translation vectors, $\|t_{\text{est}} - t_{\text{gt}}\|_2$ is their Euclidean distance, and $\|t_{\text{gt}}\|_2$ normalizes the error. Second, the attitude error is computed using unit quaternions. A rotation vector $r \in \mathbb{R}^3$ corresponds to a unit quaternion

$$q = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}, \quad q_w = \cos\left(\|r\|/2\right) \tag{20}$$

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \sin\left(\|r\|/2\right)\frac{r}{\|r\|}, \quad \|q\| = 1 \tag{21}$$

Let q_{est} and q_{gt} be the quaternions corresponding to r_{est} and r_{gt}. The attitude error is then

$$\delta_q = 2\,\arccos\left(\left|\, q_{\text{est}} \cdot q_{\text{gt}} \,\right|\right) \tag{22}$$

In Eq. (22), q_{est} and q_{gt} are unit quaternions in $\mathbb{R}^4$ [42]. The dot product $q_{\text{est}} \cdot q_{\text{gt}}$ computes the cosine of half the angle between the two rotations; taking the absolute value inside arccos ensures the smallest angle between equivalent quaternion representations $(q, -q)$. Consequently, $\delta_q \in [0, \pi]$ measures the angular discrepancy in radians [43].

3.3. Synthetic data generation

We generate synthetic RGB images of the ISS and corresponding 2D keypoint annotations by first simulating orbital motion in MATLAB/Simulink (10 Hz) and then rendering in Blender (see Table 1 and Figs. 6 and 7). Below are the detailed steps, equations, and variable definitions.

In MATLAB/Simulink, the ISS orbit is defined by six classical Keplerian elements $\{a, e, i, \Omega, \omega, \nu\}$ [44]:

- $a = R_E + 408\,\text{km}$, where $R_E = 6378\,\text{km}$ is Earth's radius and 408 km is the ISS altitude.
- $e \approx 0.0001$ is eccentricity.
- $i = 51.6°$ is inclination.

- Ω represents the right ascension of the ascending node (RAAN).
- ω is the argument of perigee.
- ν is the true anomaly (angle from perigee to current position).

At each simulation time t:

$$r(t) = \frac{a\left(1-e^2\right)}{1+e\cos\left(\nu(t)\right)} \tag{23}$$

where:

- $r(t)$ is the distance from Earth's center to the ISS at time t,
- a is the semi-major axis,
- e is eccentricity,
- $\nu(t)$ is the true anomaly at time t.

The coordinates in the orbital plane are [44]:

$$x_{\text{orb}}(t) = r(t)\,\cos\left(\nu(t)\right), \qquad y_{\text{orb}}(t) = r(t)\,\sin\left(\nu(t)\right) \tag{24}$$

Here:

- $[\,x_{\text{orb}}(t),\, y_{\text{orb}}(t)\,]^\top$ are the ISS coordinates in its orbital plane at time t.

To transform into Earth-Centered Inertial (ECI) coordinates, apply the rotation matrix

$$R_{\text{ECI}} = R_3\left(\Omega(t)\right) R_1\left(i\right) R_3\left(\omega(t)\right) \tag{25}$$

with

$$R_3(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{26}$$

Here:

- $\Omega(t)$ is RAAN at time t,
- i is inclination,
- $\omega(t)$ is argument of perigee at time t.

Thus, the ECI position vector [43] (denoted $\bar{R}(t)$) is

$$\bar{R}(t) = R_{\text{ECI}} \begin{bmatrix} x_{\text{orb}}(t) \\ y_{\text{orb}}(t) \\ 0 \end{bmatrix} \tag{27}$$

Here:

- $\bar{R}(t) \in \mathbb{R}^3$ is the ISS ECI position at time t.

The ECI velocity vector $\bar{V}(t)$ is computed by integrating two-body dynamics (standard Keplerian differential equations). MATLAB/Simulink directly outputs $\bar{R}(t)$ and $\bar{V}(t)$ at 10 Hz. We refer to these time-series as "V-bar" and "R-bar":

$$\text{V-bar} := \bar{V}(t) \in \mathbb{R}^3, \qquad \text{R-bar} := \bar{R}(t) \in \mathbb{R}^3 \tag{28}$$

For rendering purposes in Blender, spacecraft orientation was parametrized using a Rodrigues axis–angle vector $\mathbf{r}(t)$, which is derived from the quaternion representation of orbital motion. We emphasize that this $\mathbf{r}(t)$ is distinct from the translational velocity vector $\mathbf{V}(t)$: the latter strictly represents orbital dynamics, while $\mathbf{r}(t)$ is introduced solely as an orientation parameter for pose generation.

At each time step t, Simulink provides: V-bar = $\bar{V}(t)$ (expressed as an axis–angle rotation vector), R-bar = $\bar{R}(t)$ (translation). We convert V-bar = $\bar{V}(t)$ (axis–angle) into a unit quaternion:

$$q(t) = \begin{bmatrix} q_w(t) \\ q_x(t) \\ q_y(t) \\ q_z(t) \end{bmatrix} \tag{29}$$

where

$$q_w(t) = \cos\left(\|\bar{V}(t)\|/2\right) \tag{30}$$

$$\left[q_x(t),\, q_y(t),\, q_z(t)\right]^\top = \sin\left(\|\bar{V}(t)\|/2\right) \frac{\bar{V}(t)}{\|\bar{V}(t)\|} \tag{31}$$

Here:

- $\|\bar{V}(t)\|$ is the magnitude of the axis–angle vector at time t,
- $q_w(t)$ is the scalar (real) part of the quaternion,
- $\left[q_x(t),\, q_y(t),\, q_z(t)\right]^\top$ are the vector (imaginary) components.

We then convert $q(t)$ into the 3×3 rotation matrix $R(t)$ via:

$$R(t) = \begin{bmatrix} 1-2q_y^2-2q_z^2 & 2q_xq_y-2q_zq_w & 2q_xq_z+2q_yq_w \\ 2q_xq_y+2q_zq_w & 1-2q_x^2-2q_z^2 & 2q_yq_z-2q_xq_w \\ 2q_xq_z-2q_yq_w & 2q_yq_z+2q_xq_w & 1-2q_x^2-2q_y^2 \end{bmatrix} \tag{32}$$

where the time dependence of q_w, q_x, q_y, q_z is implied. The Simulink translation is simply

$$t(t) = \bar{R}(t) \in \mathbb{R}^3 \tag{33}$$

the ECI position at time t. Thus, at each t we have a full 6-DOF pose $\left(R(t),\, t(t)\right)$.

For each pose $\left(R(t),\, t(t)\right)$:

- A Python script sets Blender's virtual camera orientation to $R(t)$ and position to $t(t)$.
- The ISS CAD model, whose body frame 3D landmarks $\{X_i\}_{i=1}^{28} \subset \mathbb{R}^3$ (IDSS interface corners in Fig. 5) are known, is rendered into an RGB frame RGB_t.
- Each landmark X_i (in the ISS body frame) is transformed into the camera frame by

$$X_i^c(t) = R(t)\,X_i + t(t) \tag{34}$$

  where:

  - $X_i^c(t) = [X_{c,i}(t),\, Y_{c,i}(t),\, Z_{c,i}(t)]^\top \in \mathbb{R}^3$ is the ith landmark in camera coordinates,
  - $R(t)$ and $t(t)$ come from Eqs. (32)–(33).

- Store the pair $\left(\text{RGB}_t, \{X_i^c(t)\}_{i=1}^{28}\right)$ for later projection.

Each transformed 3D landmark [38]

$$X_i^c(t) = \begin{bmatrix} X_{c,i}(t) \\ Y_{c,i}(t) \\ Z_{c,i}(t) \end{bmatrix} \tag{35}$$

where $\mathbf{R}(t)$ and $\mathbf{t}(t)$ are obtained from Eqs. (26)–(27). It should be noted that in the synthetic rendering pipeline the Earth-Centered Inertial (ECI) frame was deliberately aligned with the Blender camera frame. As a result, the usual composition

$$F_b \;\rightarrow\; F_{\text{ECI}} \;\rightarrow\; F_c$$

collapses into a single transform by construction. In contrast, for the real-world dataset (Section 3.4), the full extrinsic mapping between body, OptiTrack world, and camera frames was explicitly estimated.

The 3D point is then projected using the camera intrinsic matrix K (Eq. (14)) and perspective division:

$$\hat{x}_i(t) = \pi\left(K\,X_i^c(t)\right) \tag{36}$$

and π denotes homogeneous normalization:

$$\pi\left(\begin{bmatrix} u \\ v \\ w \end{bmatrix}\right) = \begin{bmatrix} u/w \\ v/w \end{bmatrix} \tag{37}$$

Finally, for each time step t:

- Rendered RGB image: RGB_t.

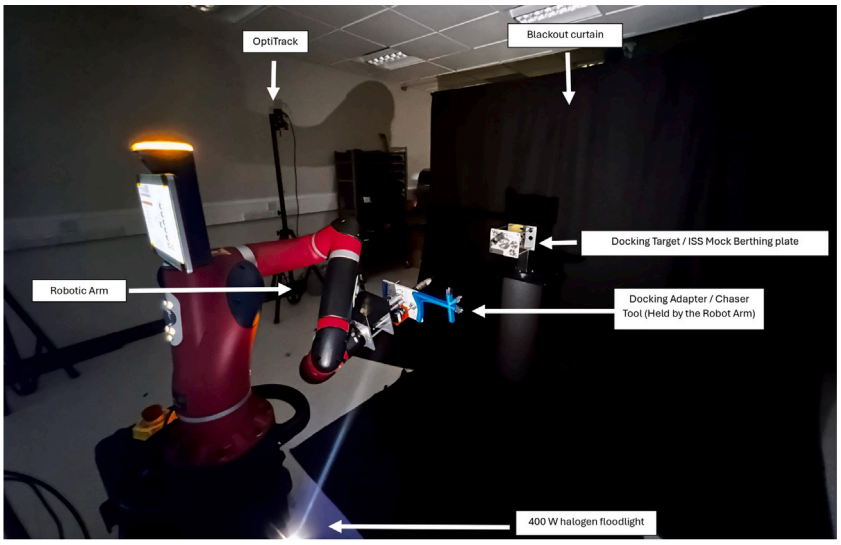


Fig. 2. Integration validation setup at City, St George's University of London's ASMIL laboratory.

- 2D keypoint set: $\{\hat{x}_i(t)\}_{i=1}^{28}$.

We save each $\left(\text{RGB}_t, \{\hat{x}_i(t)\}_{i=1}^{28}\right)$ as one annotated example. Repeating this for all t in each V-bar and R-bar sequence yields a complete synthetic dataset for CNN training/testing on 28 keypoints. By executing these steps, we produce a synthetic dataset of Blender-rendered RGB frames with accurate 2D keypoint annotations for all 28 IDSS landmarks.

3.4. Real-world dataset generation

Real-world data were collected in the ASMIL lab at City, St George's University of London (Fig. 2 and 3). A robotic arm executes docking maneuvers while a Visual-Based System (VBS) captures RGB images at 10 Hz. A blackout curtain and a 400 W halogen floodlight (60° beam spread) simulate deep-space lighting. The lamp's solid angle is

$$\Omega = 2\pi\left(1 - \cos 30^\circ\right) = 0.8418\ \text{sr} \tag{38}$$

[45] where Ω is the beam's steradian measure. To achieve irradiance

$$E = 1361\ \text{W/m}^2 \tag{39}$$

$$E = \frac{P}{\Omega\, r^2} \implies r \approx 0.6\,\text{m} \tag{40}$$

[46] where:

- $P = 400\,\text{W}$ is the lamp power.
- r is the lamp-to-target distance (m).
- Ω is from Eq. (38).

Ground-truth 6-DOF poses of the docking target are obtained via an OptiTrack system [47] (six PrimeX 13 cameras, 240 Hz, 1280×1024px, $\leq$0.2 mm positional error, $\leq$ 0.5° rotational error). A DFK22BUC03 CMOS camera (744×480px, 3.5 mm focal length) serves as the VBS sensor; its intrinsic matrix K was defined in Eq. (14). Table 2 summarizes the VBS camera parameters.

OptiTrack measures the poses of two marker clusters as elements of $SE(3)$:

$$T_{oc}(t) \quad \text{and} \quad T_{ob}(t) \tag{41}$$

[48] where:

- $T_{oc}(t) \in SE(3)$ is the pose of the camera-housing marker frame F_c in the world frame F_o at time t.
- $T_{ob}(t) \in SE(3)$ is the pose of the target-body marker frame F_b in F_o at time t.

We note that $SE(3)$, the Special Euclidean group in three dimensions, comprises all 3D rigid-body transforms. Each element of $SE(3)$ can be written as a 4×4 homogeneous matrix

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \tag{42}$$

where $R \in SO(3)$ is a 3×3 rotation matrix, $t \in \mathbb{R}^3$ is a translation vector, and the bottom row [0001] enforces homogeneous coordinates. Applying $T \in SE(3)$ to a homogeneous point $[X_x, X_y, X_z, 1]^\top$ produces [38]:

$$T \begin{bmatrix} X_x \\ X_y \\ X_z \\ 1 \end{bmatrix} = \begin{bmatrix} R\,[X_x, X_y, X_z]^\top + t \\ 1 \end{bmatrix} \tag{43}$$

However, the VBS navigation algorithm requires the target's pose in the camera optical-center frame F_i. We therefore estimate two static transforms in $SE(3)$:

$$T_{ic} : F_c \longrightarrow F_i, \qquad T_{sb} : F_b \longrightarrow F_s \tag{44}$$

where:

- F_i is the camera's optical-center frame (pinhole center),
- F_s is a scene frame rigidly attached to a known calibration target on the ISS mock-up.

To estimate T_{ic} and T_{sb}, we place the calibration target in view of both OptiTrack and the VBS camera. Each known 3D calibration point $X_s \in \mathbb{R}^3$ in the scene frame F_s projects to measured pixel coordinates $x_{\text{meas}}(t)$. Using the intrinsic matrix K (Eq. (14)) and the projection function π (Eq. (10)), its predicted pixel location is

$$\hat{x}(t) = \pi\Big(\, K\left[T_{ic}\, T_{oc}(t)^{-1}\, T_{ob}(t)\, T_{sb}^{-1}\, X_s\right]\Big) \tag{45}$$

where:

- $T_{oc}(t)^{-1} T_{ob}(t)$ maps F_b to F_c at time t,
- $T_{sb}^{-1} X_s$ transforms the 3D point X_s from F_s to F_b,
- T_{ic} then maps from F_c to F_i,
- K forms camera-frame homogeneous coordinates.

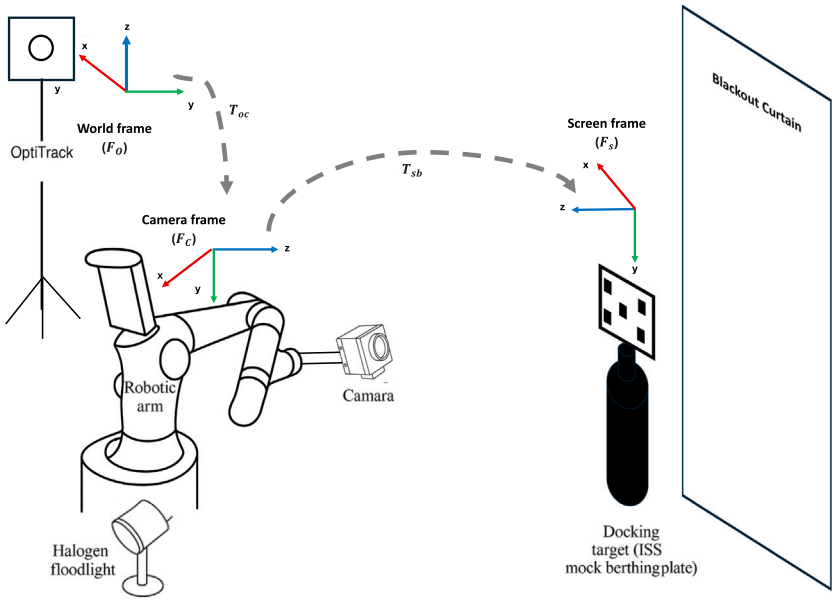


Fig. 3. Schematic representation of the integration validation setup, illustrating the world, camera, and screen frames.

We minimize the reprojection error over all calibration points s and times t:

$$\min_{T_{ic},T_{sb}} \sum_{t,s} \left\| x_{\text{meas}}(t) - \hat{x}(t) \right\|_2^2 \tag{46}$$

Once T_{ic} and T_{sb} are known, the target-to-camera relative transform at time t is

$$T_{bc}(t) = T_{sb} T_{ob}(t)^{-1} T_{oc}(t) T_{ic}^{-1} \tag{47}$$

[38] where:

- $T_{ob}(t)^{-1} T_{oc}(t)$ maps F_c to F_b via F_o,
- Multiplying by T_{sb} sends F_b to the scene frame F_s,
- Finally, T_{ic}^{-1} maps F_i back to F_c, yielding the target in F_i.

Decomposing $T_{bc}(t) \in SE(3)$ yields:

$$R(t) \in SO(3) \qquad t(t) \in \mathbb{R}^3 \tag{48}$$

the rotation matrix and translation vector of the target in the camera optical-center frame at time t. These are then projected back to 2D annotation keypoints and stored as the annotation for the frame.

Twelve docking trajectories were recorded with alternating "port" and "starboard" lighting angles (Fig. 4). Each sequence lasts 319–358s. The first ten sequences are used for CNN training/validation on real-world images, and the last two for final testing. In half of the sequences, a static pose misalignment is introduced during translation and corrected before the final docking phase, simulating unplanned attitude disturbances.

This laboratory configuration captures the essential visual characteristics that define space docking environments. The controlled setup deliberately replicates the three fundamental challenges present in orbital scenarios: (1) the space-representative illumination contrast without atmospheric diffusion, achieved through our directional 400 W halogen source that creates the sharp shadow boundaries typical of unfiltered solar illumination; (2) absence of terrestrial reference features, enforced by the blackout background that forces reliance solely on spacecraft-specific visual cues—the primary information source available during actual space rendezvous; and (3) specular surface interactions on metallic spacecraft components under directional lighting, which the ISS mock-up materials authentically reproduce.

The laboratory emulation setup specifically validates performance across the 10-meter to contact operational range, representing the most critical and highest-risk segment of autonomous docking missions where precision requirements are most stringent and pose estimation errors can directly impact mission success. While orbital environments

Fig. 4. Real ground-truth keypoints (12) on the ISS mock target.

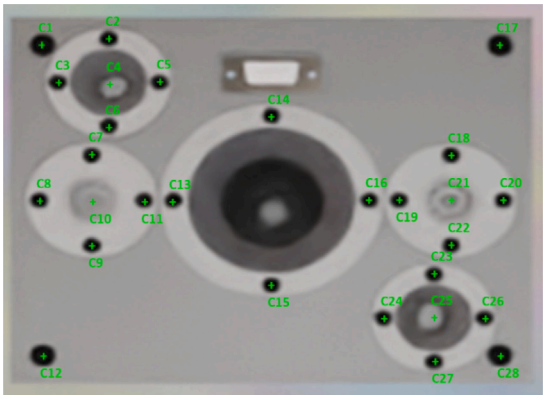


Fig. 5. Synthetic ground-truth keypoints (28) on the ISS mock target.

Table 1
Synthetic dataset characteristics used for training and testing. A '+' denotes motion along the positive axis and a '–' along the negative axis.

Seq.	Docking port	V-bar	R-bar	Sun elevation (°)	ISS	Perlin	Duration (s)	Split
1	1	+		37	×	×	332	Test
2	1	+		75	×	×	319	Train
3	2		–	56		×	358	Train
4	3		–	146		×	336	Train
5	3	–		127	×		348	Train
6	4	–		165	×		327	Train
7	4		+	56		×	333	Train
8	5		–	146		×	329	Train
9	5	+		56	×		333	Train
10	6	+		146	×		342	Train
11	6		–	56		×	323	Train
12	6	+		146	×		355	Test

Table 2
Technical specifications of the DFK 22BUC03 VBS camera.

Parameter	Units	Value
Resolution	px	744 × 480
Maximum Frame Rate	Hz	76
Focal Length	mm	3.5
Horizontal FOV	°	65.6
Vertical FOV	°	44.7

introduce additional complexities such as dynamic backgrounds and varying solar angles, these factors typically provide supplementary visual information rather than fundamental algorithmic challenges. Our laboratory approach therefore captures the core computer vision problems inherent to space docking while establishing a controlled baseline for performance validation during the most demanding operational phase. The high-precision OptiTrack [47] ground truth system enables algorithm validation at accuracy levels that exceed operational requirements, ensuring that laboratory-validated performance will translate reliably to space applications where the fundamental visual challenges remain consistent but may be supplemented by additional orbital context information.

3.5. Training and validation of CNN models

To train and validate the CNN models, the synthetic dataset described in Table 1 is carefully partitioned into training, validation, and testing subsets in case 1 and into training and testing only for case 2. Sequences 1 and 8 were only used for testing for both cases to check the ability of the model to predict the docking of other scenarios that were used neither in training nor in validation. These particular sequences were chosen because they illustrate diverse docking scenarios and scenarios such as different docking ports, approach axes, and sun elevation angles which help in assessing the model's efficiency.

The remaining sequences (2–7 and 9–12) are used for training and validation purposes in case 1 and only for training in case 2. To ensure an unbiased division for case 1, these respective sequences were split according to an 80%/20% ratio, with 80% allocated to the training set and 20% to the validation set.

Due to the fact that the data contains long temporal sequences, which may contain hundreds of frames, the sequences are divided into smaller temporal segments in order to ensure that the desired 80%/20% split can be realized without bias. These smaller segments are obtained by splitting each original sequence into batches of 32. In this manner, the training and validation datasets were made to have equal samples.

During training, the CNN model is optimized to minimize the Mean Squared Error (MSE) loss function, which measures the accuracy of predicted keypoints against their ground truth positions. The model is trained using different gradient optimizers, and an exponential decay learning rate starting at 0.001, tuned using the validation set through variations applied either to a single layer or to multiple layers, in order to ensure that the best convergence behavior is achieved. Early stopping is applied based on the validation loss to prevent overfitting, halting the training process if the validation performance does not improve for a set number of epochs. The complete set of training hyperparameters is summarized in Table 3

The Mean Squared Error (MSE) estimates the average squared deviation of the predicted and ground truth keypoints. For every keypoint, it calculates the squared Euclidean distance between the coordinates of the predicted and actual location. It is defined as:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right) \tag{49}$$

The MSE is used as the main evaluation criterion for this keypoint detection task as it is more sensitive to large errors which is important for accurate keypoint positioning. Importantly, the cost function introduced in Eq. (9) refers to the reprojection error used during the PnP based pose estimation stage, which differs fundamentally from the MSE loss applied here for keypoint regression.

And finally, a regular and soft dataset are employed to assess their impact on enhancing model stability and improving predictive accuracy in new docking scenarios. The regular dataset includes every frame from each approach sequence, even those captured at very close ranges where the docking-port features become ambiguous. In contrast, the soft data set deliberately omits the final frames of each sequence: those in which the chaser is so close that the distinguishing markers of the port are no longer clearly visible.

3.5.1. Case study 1: Comparison of CNN backbones with the regular dataset

The first case study investigates the performance of CNNs with different backbone architectures trained on the regular dataset. It examines lightweight models, including EfficientNet, MobileNet, ResNet50, and HRNet. The regular dataset in this study consists of the full set of training images. This step aims to compare these backbone architectures and determine which achieves the best performance when trained on the complete dataset.

3.5.2. Case study 2: Comparison of CNN backbones with the soft dataset

The second case study extends the first by evaluating CNN backbone architectures on the soft dataset. This task aims to examine how training on the soft dataset influences model performance compared to the regular dataset.

3.5.3. Case study 3: Training and testing the real dataset

This case study utilizes only the real dataset for both training and testing, providing deeper insight into the performance of the pipeline when applied to data from the same domain. Unlike the synthetic dataset, which employs 28 keypoints for detection, the real dataset is simplified to 12 keypoints to align with experimental requirements and make detection feasible in scenarios where dense keypoint labeling is impractical (Fig. 4). This approach focuses on training the model on real-world images to evaluate its capability in processing and interpreting docking environment data.

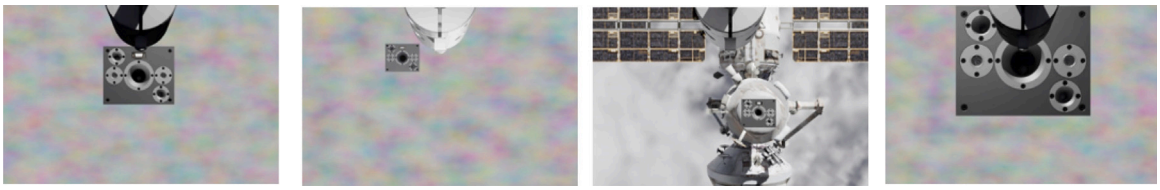


Fig. 6. Perlin noise background samples for synthetic data.

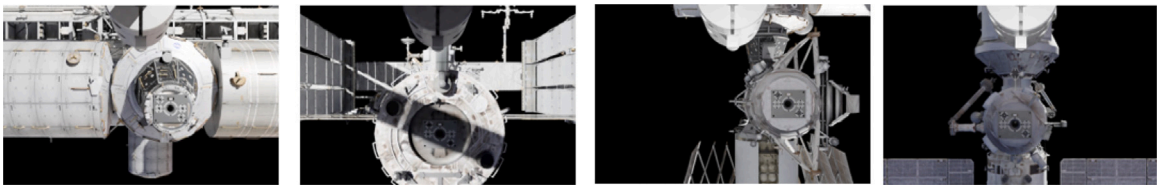


Fig. 7. ISS background samples for synthetic data.

Table 3
Summary of training hyperparameters used across all experimental cases.

Category	Hyperparameters
Optimizer	Adam
Learning rate	1×10^{-3} (StepLR: step=30, $\gamma = 0.1$)
Batch size	32
Epochs	100
Weight initialization	Xavier uniform
Loss function	MSE on 2D keypoints
Backbones	ResNet-50, MobileNet, EfficientNet-B0, HRNet
Dropout rate	0.5 (FC layers)
Data augmentation	Flips, rotations, color jitter, brightness adj.
Pose solver	EPnP + RANSAC ($\tau = 4$ px, 1000 iters)
Covariance (EKF)	diag=$10^{-3}, 10^{-2}, 10^{-1}$

4. Experimental results and analysis

This section presents the experiments and a comprehensive analysis of the proposed method by applying to both synthetic data and real-world data collected from a representative laboratory environment.

4.1. Software training setup: Data processing and augmentation

During training, we preprocess each image by resizing to a fixed resolution and normalizing pixel intensities. We then apply a suite of data augmentation operations detailed in Table 4, including random rotations, translations, brightness shifts, and other perturbations. These augmentations expose the CNN to diverse visual conditions, improving its ability to generalize to new scenes. During inference, predicted keypoint coordinates are passed through a Gaussian smoothing filter to suppress spurious noise, yielding more accurate and stable inputs for the final pose estimation stage.

4.2. Backbone comparison

Fig. 8(a) showcases the performance of the ResNet50, MobileNet, EfficientNet, and HRNet backbone models across the acquisition, approach, and final docking phases, highlighting distinct characteristics in the way each model handles position and attitude errors.

The ResNet50 model maintains steady performance throughout every evaluated phase. The position error briefly surges during the acquisition phase to about 0.20% range-normalized for sequence `dp000, s0`, but reaches a slightly higher value for sequence `dp003, s1`. The initial spike shows rapid stabilization, which allows position and attitude errors to maintain their integrity within satisfactory ranges despite the early variations. The acquisition phase raises attitude error, yet keeps it under 15 degrees demonstrating that ResNet50 achieves reliable rotational precision. ResNet50 maintains effective operational performance during the approach phase because both position and attitude errors show a systematic reduction. Throughout the sequence progression, neither the position error exceeds 0.10% nor does the attitude error exceed 5 degrees (Fig. 9a). The ResNet50 model shows small error rates and stable performance through the final docking sequence which proves that it successfully manages positional and rotational accuracy demands in moving docking environments. During the acquisition stage, MobileNet demonstrates marginally increased initial positional deviations which settle at 0.3% across both data channels. The performance of MobileNet strengthens notably through both approach and docking stages reaching exceptional lows with position error falling below 0.2% during dp000 sequence s0. During acquisition the orientation error reaches its maximum at 5 degrees before stabilization throughout subsequent phases. The performance evaluation shows that MobileNet functions as a lightweight solution for resource-limited scenarios while maintaining positional accuracy but faces initial orientation challenges.

During its acquisition phase, EfficientNet's orientation error experiences high initial spikes until reaching over 10 degrees on models such as `dp003, s1`. EfficientNet demonstrates trending stability during approach and docking phases following the initial spikes but maintains higher error rate variability when measured against other models. EfficientNet starts with difficulties in position and orientation management but achieves smoother transitions during later phases of the sequence.

HRNet maintains task consistency across metrics for position and orientation yet reveals more orientation deviations within `dp003, s1`. The position error of HRNet maintains no variability between phases while orientation error reveals substantial difficulties when faced with sequences that demonstrate high variability. While HRNet demonstrates good effectiveness its performance drops during sequences that demand fast orientation changes.

4.3. Computational efficiency of backbones

To evaluate suitability for onboard deployment, we benchmarked each backbone's parameter count, theoretical compute, and projected inference latency on the S-A1760 Venus™, which features an NVIDIA® Jetson™ TX2i SoM with 256 CUDA cores delivering up to 1 TFLOPS at high energy efficiency, optimized for short-duration spaceflight, NEO, and LEO satellite applications. The choice of the S-A1760 Venus™ platform was guided by its widespread adoption by American space agencies and its proven reliability in similar space applications [33]. The results, adjusted based on hardware benchmarks specific to the Jetson TX2i platform (batch size 1, 224 × 224 inputs), are summarized in Table 5.

Table 4
Image augmentation parameters used during CNN training.

Transformation	Parameter range	Unit	Description
Channel Shift	−20 to 20	–	Pixel intensity shift
Gaussian Blur	7 to 13	px	Kernel size
Gaussian Noise	3×10^{-3}–1×10^{-2}	–	Variance
JPEG Compression	2 to 8	–	Compression level
Median Blur	7 to 13	px	Kernel size
Patch Dropout	10%	%	Proportion of image area masked
Patch Size	3% to 5%	%	Relative patch size
Brightness Adjustment	−0.2 to 0.2	–	Intensity adjustment
Contrast Adjustment	0.8 to 1.2	–	Intensity adjustment
CLAHE	2 to 6	–	Number of CLAHE tiles
Gamma Correction	0.35 to 1.50	–	Intensity correction factor
Camera Rotation	−5° to 5°	deg	Rotation magnitude per axis
In-plane Image Rotation	−5° to 5°	deg	Overall image rotation
Image Translation	−150 to 150	px	Translation magnitude

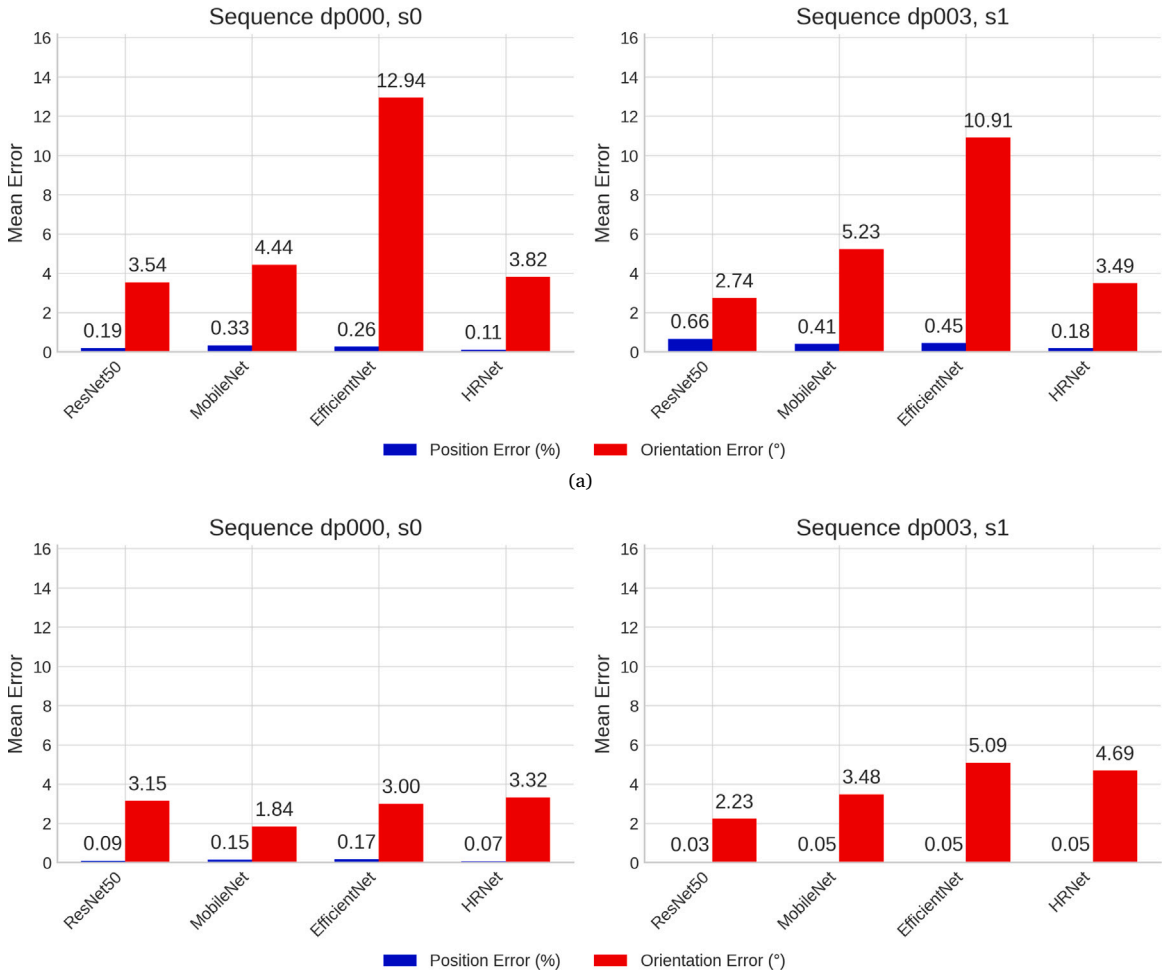


Fig. 8. Comparison of pose estimation performance metrics using regular and soft datasets across different backbones for ISS test sequences. (a) Comparison of Mean Position (%) and Mean Attitude Errors (deg) for Sequences dp000, s0 and dp003, s1 using the regular dataset. (b) Comparison of Mean Position (%) and Mean Attitude Errors (deg) across Backbones for Sequences dp000, s0 and dp003, s1 using the soft dataset.

Table 5
Model size, theoretical FLOPs, and projected inference latency on the S-A1760 Venus™ platform (mean ± std estimated over 100 simulated runs).

Backbone	Params (M)	FLOPs (G)	GPU Latency (ms)	CPU Latency (ms)
ResNet50	25.6 [11]	4.1 [11]	220 ± 15 [34]	650 ± 30 [34]
EfficientNet	5.3 [13]	0.39 [13]	110 ± 8 [34]	340 ± 20 [34]
MobileNet	3.5 [32]	0.30 [32]	80 ± 6 [34]	240 ± 15 [34]
HRNet	9.3 [14]	4.0 [14]	240 ± 18 [34]	750 ± 40 [34]

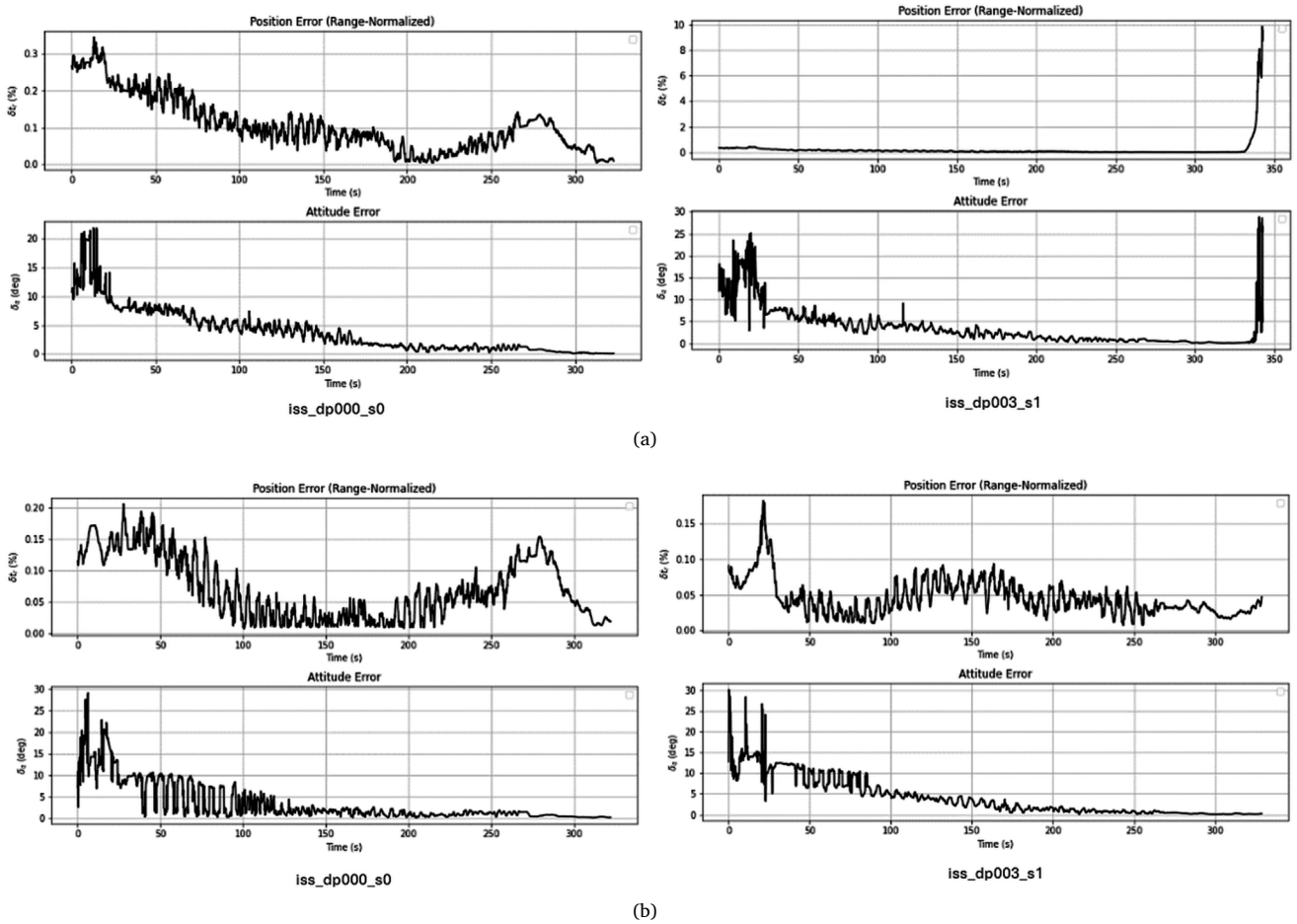


Fig. 9. (a) Position and Attitude Errors for ISS dp000, s0 and ISS dp003, s1 (ResNet50 backbone/Regular Dataset). (b) Position and Attitude Errors for ISS dp000, s0 and ISS dp003, s1 (ResNet50 backbone/Soft Dataset).

Computational efficiency is a critical concern for space-based systems, where power, thermal, and real-time constraints significantly restrict onboard processing budgets. Although ResNet50 is neither the smallest (25.6 M params) nor the lowest compute (4.1 GFLOPs) backbone, it consistently delivers the highest keypoint localization accuracy in our experiments (Fig. 8). Its GPU latency of approximately 220 ms/frame remains practical for a 4–5 Hz inference pipeline on the S-A1760 Venus™ hardware, making it a viable choice when precision is paramount. Lighter models such as MobileNet (3.5 M params, 0.3 GFLOPs, ~80 ms/frame) and EfficientNet (5.3 M params, 0.39 GFLOPs, ~110 ms/frame) offer approximately 2×–3×speedups at a modest accuracy penalty of 3%–5%, potentially making them preferable for missions with tighter power or latency constraints. However, for proximity operations in challenging lighting or complex backgrounds – scenarios where maximal pose precision directly impacts mission safety – ResNet50's superior representational capability justifies its higher computational cost, making it our recommended backbone.

4.4. Effect of dataset regularization on model performance

The introduction of the soft dataset significantly enhances model generalization, as observed in Fig. 8. The approach achieves reduced position and orientation errors spikes with faster initial convergence and maintains uniform accuracy during the approach docking phase.

For ResNet50 specifically, Figs. 9(a) and 9(b) illustrate how the use of a soft dataset reduces variability which results in stable position and orientation error measurements during all docking phases. In sequence dp000, s0, the soft dataset reduces mean position error from 0.19% to 0.09% (a 52.6% improvement) and decreases orientation error from 3.54° to 3.15° (an 11.0% reduction), yielding noticeably less variability throughout the approach. In sequence dp003, s1, it drives position error down from 0.66% to 0.03% (a 95.4% improvement) and stabilizes orientation error at 2.23° (an 18.6% reduction).

The rotational accuracy performance of MobileNet sees noticeable improvement due to dataset regularization from soft examples. During experiment `dp000, s0` MobileNet achieved a 60.5% reduction in position error while shrinking from 0.38% to 0.15% and experienced a 58.6% loss in orientation error leading from 4.44° down to 1.84°. On sequence `dp003, s1`, MobileNet demonstrates decreased positional inaccuracy by 87.8% (from 0.41% to 0.05%) together with a 33.5% decrease in orientation error levels from 5.23° to 3.48°. The study reveals MobileNet's successful adaptation to soft dataset regularization which aids in diminishing variability together with accelerated error convergence during the system acquisition.

The most substantial orientation error improvement is demonstrated by EfficientNet. For sequence `dp000, s0` the position error declined by 34.6% (from a starting point of 0.26% to 0.17%) while orientation error diminished by an extraordinary rate of 76.8% (moving from 12.94° to 3.00°). The data reveals that while position error was cut dramatically by 88.9% (from 0.45% to 0.05% opening to close), orientation error decreased by 53.3% (from 10.91° to 5.09°) in sequence `dp003, s1`. Analytical results reveal that soft dataset usage reduces initial phase fluctuations and rotational dynamics which strengthens EfficientNet's performance in metrics for position and orientation.

Table 6
Comparison of Position and attitude errors for dp000, s0 and dp003, s1 using ResNet50 backbone.

Dataset	Error type	dp000, s0			dp003, s1		
		Mean	Median	Std. Dev.	Mean	Median	Std. Dev.
Regular Dataset	Position Error(%)	0.11	0.09	0.07	0.18	0.06	0.75
	Attitude Error(deg)	3.83	2.32	3.89	3.49	2.15	4.25
Soft Dataset	Position Error(%)	0.07	0.06	0.05	0.05	0.04	0.02
	Attitude Error(deg)	3.32	1.34	4.47	4.09	2.07	4.58

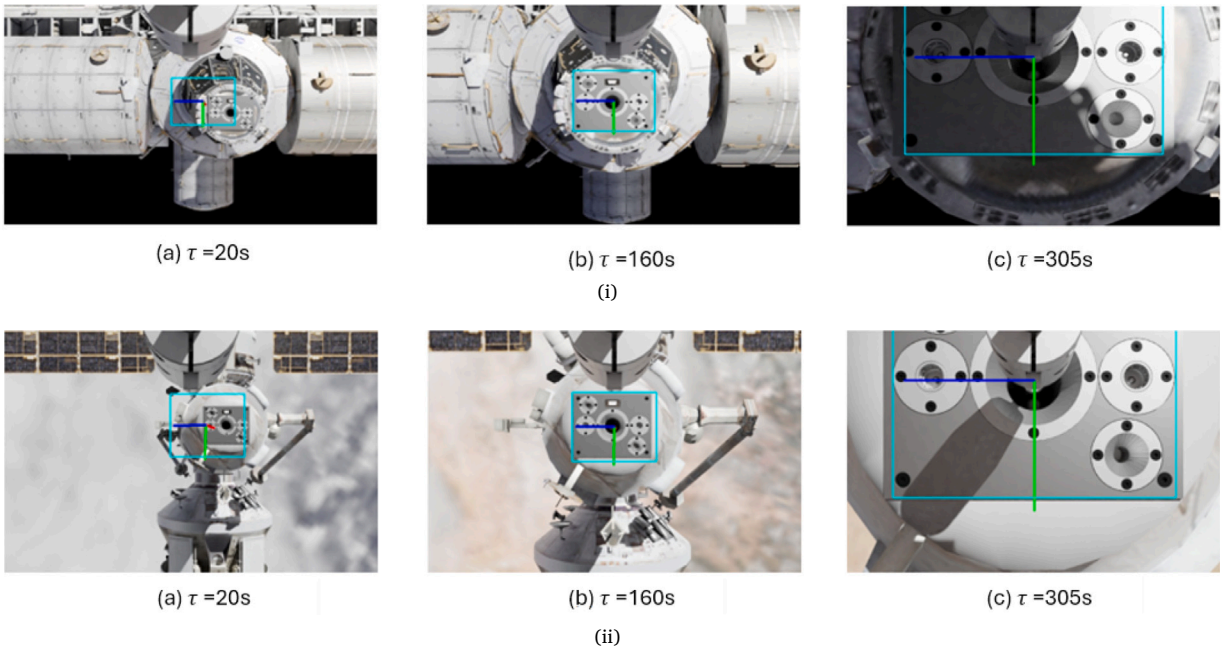


Fig. 10. (i) Qualitative pose estimation performance on the synthetic test dataset dp000, s0. (ii) Qualitative pose estimation performance on the ISS dp003, s1 test sequence.

The soft dataset shows enhanced positional accuracy benefits for HRNet. During the `dp000, s0` sequence HRNet achieves decreased position and orientation errors by 36.4% (0.11% down to 0.07%) and 13.1% (3.82° to 3.32°) respectively. HRNet's performance in sequence `dp003, s1` shows significant positional error improvement with a reduction of 72.2% (from 0.18% to 0.05%) while orientation error registers minimal yet direct degradation (rising from 3.49° to 4.69°). HRNet demonstrates effective positional accuracy improvements through the regularization of the soft dataset but shows performance difficulties during strong rotational changes.

In summary, Fig. 8 shows how the soft dataset leads to decreased error instances while maintaining stable results throughout different backbone architectures. Through this regularization effect, models demonstrate increased resilience because dynamic docking operations benefit from improved stationary and rotational predictability. Among all tested backbones, ResNet50 consistently achieves the best results, both relative position and orientation metrics (Figs. 8(b) and Table 6). We advanced ResNet50 trained on the soft dataset to continue the validation process within this pipeline. The qualitative pose estimation results using ResNet50 trained on the soft dataset and tested on synthetic test sequences are presented in Figs. 10(i) and 10(ii).

4.5. Evaluation pipeline performance on real dataset with real dataset training

This section evaluates the performance of the full pipeline model described in Fig. 1, tested on the real dataset. The results show strong accuracy in both position and attitude estimation, underscoring the backbone's effectiveness in supporting accurate predictions in real docking scenarios (Table 7).

A mean position error of 0.28% along with a median of 0.18% and a standard deviation of 0.16% is present for dataset 'experimental/11' whereas dataset 'experimental/12' shows a mean position error of 0.30% combined with a median of 0.18% and a standard deviation of 0.17%. Because position error rates remain remarkably low, the model demonstrates precise functionality which stems from training in real-world conditions paired with testing that follows those standards.

Position data aside, this model proves similarly reliable when handling attitude error. The 'experimental/11' group exhibits a mean attitude error of 1.08 degrees with a standard deviation of 1.02 degrees but 'experimental/12' displays mean and standard deviation values of 1.14 degrees and 1.06 degrees, respectively. This model retains its accuracy of orientation interpretation within real-world docking scenarios by using the ResNet50 backbone even when the number of keypoints is low.

Comparing the results obtained in Table 7 and it is qualitative pose estimation illustration in Fig. 11 to OIBAR's direct approach in [29], which achieved position errors of approximately 1.02% for 'experimental/11' and 1.17% for 'experimental/12', and attitude errors of 1.65 degrees for 'experimental/11' and 0.86 degrees for 'experimental/12', we observe that the hybrid approach with the ResNet50 backbone achieves comparable or superior results in position estimation. Through special optimization for docking assignments OIBAR's direct technique achieves minimal attitude errors which reveals its efficiency to maintain precision during orientation assessment especially shown for 'experimental/12'. Through its combined feature learning

Table 7
Position and attitude errors for experimental/11 and experimental/12.

Error type	Experimental/11			Experimental/12		
	Mean	Median	Std. Dev.	Mean	Median	Std. Dev.
Position Error (%)	0.28	0.18	0.16	0.30	0.18	0.17
Attitude Error (deg)	1.08	1.09	1.02	1.14	1.12	1.06

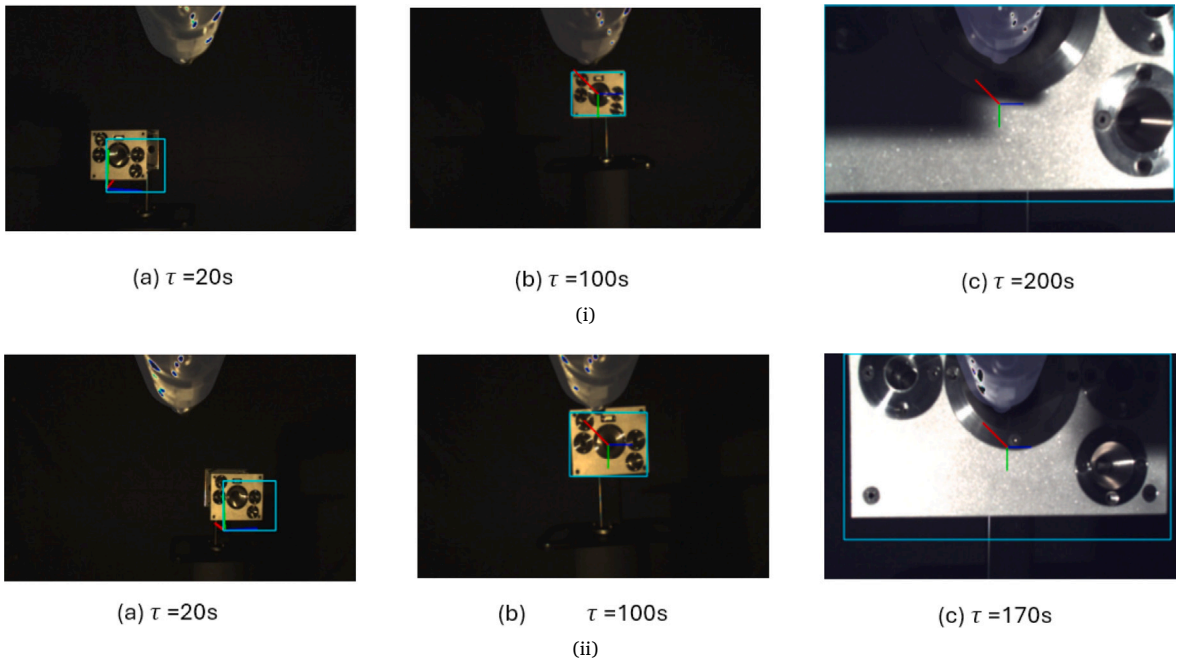


Fig. 11. (a) Qualitative pose estimation performance on the experimental/11 test sequence. (b) Qualitative pose estimation performance on the experimental/12 test sequence.

and keypoint detection capabilities this hybrid approach becomes a formidable choice in diverse real-world scenarios.

Although the real-world dataset used in Case 3 comprised fewer frames (each trajectory spanning ∼200 s compared to ∼300 s in the synthetic cases), the resulting model achieved superior attitude accuracy. This apparent contradiction can be explained by the richer information content of the real sequences. The synthetic trajectories, while longer, were generated under clean orbital backgrounds and controlled rendering settings. In contrast, the real dataset introduced far more challenging visual conditions, including the presence of a blackout curtain with strong directional illumination, specular reflections from metallic surfaces, sensor-induced noise, and realistic optical clutter. Moreover, the real docking maneuvers involved shorter but more dynamic approach trajectories, which exposed the network to higher-frequency viewpoint changes and natural keypoint occlusions. These factors collectively made the real data more information-dense, enabling the CNN+PnP+RANSAC pipeline to extract stronger geometric and appearance cues for rotational estimation. As a result, the real-trained model outperformed the synthetic-trained cases in attitude accuracy despite the reduced dataset size, demonstrating the robustness of the proposed hybrid methodology under operationally realistic conditions.

These findings reinforce that the strength of our approach is not only in performance metrics but also in the methodological choices tailored for docking conditions. Unlike generic combinations of CNN-based keypoint detectors with PnP solvers, the proposed framework incorporates several domain-specific innovations that explain the robustness observed across all three cases. First, the PnP+RANSAC stage is statistically configured based on reprojection-error distributions to ensure reliability under sensor noise and docking dynamics. Second, the CNN keypoint detector integrates an attention mechanism to suppress clutter and specular highlights, both prevalent in orbital imagery. Third, a "soft dataset" regularization strategy reduces temporal redundancy in long docking sequences, improving generalization across backbones and datasets. Fourth, validation under space-representative conditions – using a robot-in-the-loop setup with calibrated halogen lighting, OptiTrack ground truth, and a VBS camera – ensures fidelity to real mission challenges. Finally, benchmarking on flight-grade compute hardware (S-A1760 Venus, Jetson TX2i) and explicitly treating the ISS as a non-cooperative target further underline the operational relevance of the framework. Collectively, these elements establish a tailored and safety-motivated hybrid design that balances robustness, interpretability, and on-board feasibility for mission-critical docking operations.

5. Conclusion

This research introduced a hybrid monocular pose estimation framework for autonomous space docking systems, resolving high-accuracy position and rotation estimation requirements for On-Orbit Servicing and Active Debris Removal. The method delivers scalable efficiency through lightweight CNNs with PnP and RANSAC.

Analysis of CNN models such as ResNet50, MobileNet, EfficientNet, and HRNet on synthetic and real datasets showed ResNet50 as the best backbone across both settings. In multiple scenarios, ResNet50 recorded minimal positional and attitude errors and showed enhancements with the soft dataset. For sequence `dp000, s0`, the soft dataset reduced mean position error by 52.6% (0.19% → 0.09%) and orientation error by 11.0% (3.54° → 3.15°). For sequence `dp003, s1`, it achieved a 95.4% position error reduction (0.66% → 0.03%) and 18.6%

orientation error reduction (2.74° → 2.23°). These results confirm ResNet50's generalization and accuracy throughout dynamic docking sequences.

Soft datasets improved generalization by reducing variability from outliers in positional and rotational errors across all backbones. MobileNet and EfficientNet also improved through dataset regularization, minimizing errors and demonstrating its role in stability and reliability enhancement.

Real-world datasets validated robustness. Controlled laboratory environments gave favorable training, but real-world data introduced generalization problems, especially with reduced keypoints and changing viewpoints. Position errors were 0.28% for experimental/11 and 0.30% for experimental/12, with attitude errors of 1.08° and 1.14° for real-world-tuned ResNet50. Optimal performance for space applications depends heavily on domain-specific training data.

Findings show the hybrid framework generated position estimates matching or outperforming direct methods, which showed 1.02% and 1.17% position errors for experimental/11 and /12 and attitude errors of 1.65° and 0.86°. Despite direct methods reducing attitude errors in some cases, the hybrid approach demonstrates higher adaptability and robustness via feature learning and keypoint detection.

Although performance is strong under controlled conditions, future work will address limitations through environmental and algorithmic enhancements. Environmental improvements will add dynamic orbital backgrounds, multi-source illumination including Earth albedo and solar angles, and celestial bodies producing non-uniform conditions. Algorithmic robustness will be enhanced with adaptive RANSAC and uncertainty quantification for systematic outliers and decision making. Further strategies include: adaptive thresholding (τ set dynamically, e.g. 90th percentile of reprojection error histogram); Sequential Probability Ratio Tests (SPRT) in the RANSAC loop to reject unlikely poses early, reducing overhead; and geometry-based priors from spacecraft kinematics and docking-port geometry (bounds on angular velocities, lateral offsets) to filter implausible poses. Together, these strategies allow PnP+RANSAC to adapt to data quality, filter outliers, enforce physical constraints, and preserve real-time performance for deployment.

CRediT authorship contribution statement

Safinaz Khalil: Writing – original draft. **Ziwei Wang:** Writing – review & editing. **Nabil Aouf:** Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J.-C. Liou, N.L. Johnson, A sensitivity study of the effectiveness of active debris removal in leo, Acta Astronaut. 64 (2009) 236–243.

[2] B.B. Reed, R.C. Smith, B.J. Naasz, J.F. Pellegrino, C.E. Bacon, The restore-l servicing mission, in: AIAA Space 2016, 2016, p. 5478.

[3] A. Flores-Abad, O. Ma, K. Pham, S. Ulrich, A review of space robotics technologies for on-orbit servicing, Prog. Aerosp. Sci. 68 (2014) 1–26.

[4] M. Delpech, J.-C. Berges, T. Karlsson, F. Malbet, Results of prisma/ffiord extended mission and applicability to future formation flying and active debris removal missions, Int. J. Space Sci. Eng. 5 (1) (2013) 382–409.

[5] F. Zhang, G. Duan, Robust adaptive integrated translation and rotation control of a rigid spacecraft with control saturation and actuator misalignment, Acta Astronaut. 86 (2013) 167–187.

[6] S. Sharma, S. D'Amico, Reduced-dynamics pose estimation for non-cooperative spacecraft rendezvous using monocular vision, in: 38th AAS Guidance and Control Conference, Vol. 2, Breckenridge, Colorado, 2017.

[7] J. Song, D. Rondao, N. Aouf, Deep learning-based spacecraft relative navigation methods: A survey, Acta Astronaut. 191 (2022) 22–40.

[8] R. Opromolla, M.Z. Di Fraia, G. Fasano, G. Rufino, M. Grassi, Laboratory test of pose determination algorithms for uncooperative spacecraft, in: 2017 IEEE International Workshop on Metrology for AeroSpace, MetroAeroSpace, IEEE, 2017, pp. 169–174.

[9] T. Phisannupawong, P. Kamsing, P. Torteeka, S. Channumsin, U. Sawangwit, W. Hematulin, T. Jarawan, T. Somjit, S. Yooyen, D. Delahaye, et al., Vision-based spacecraft pose estimation via a deep convolutional neural network for noncooperative docking operations, Aerospace 7 (2020) 126.

[10] Z. Wang, N. Aouf, J. Pizarro, C. Honvault, Robust adversarial attacks detection for deep learning based relative pose estimation for space rendezvous, Adv. Space Res. 75 (2025) 560–575.

[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[12] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv preprint arXiv:1704.04861.

[13] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[14] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.

[15] R. Du, W. Liao, X. Zhang, Feasibility analysis of angles-only navigation algorithm with multisensor data fusion for spacecraft noncooperative rendezvous, Astrodynamics 7 (2023) 179–196.

[16] P. Zhang, D. Wu, H. Baoyin, Real-time hybrid method for maneuver detection and estimation of non-cooperative space targets, Astrodynamics 8 (2024) 437–453.

[17] O. Kechagias-Stamatis, N. Aouf, V. Dubanchet, M.A. Richardson, Deeplo: Multi-projection deep lidar odometry for space orbital robotics rendezvous relative navigation, Acta Astronaut. 177 (2020) 270–285.

[18] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, D. Rus, Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2020, pp. 5135–5142.

[19] D. Rondao, N. Aouf, M.A. Richardson, Chinet: Deep recurrent convolutional learning for multimodal spacecraft pose estimation, IEEE Trans. Aerosp. Electron. Syst. 59 (2022) 937–949.

[20] H. Yang, X. Xiao, M. Yao, Y. Xiong, H. Cui, Y. Fu, Pvspe: A pyramid vision multitask transformer network for spacecraft pose estimation, Adv. Space Res. 74 (2024) 1327–1342.

[21] P.F. Proença, Y. Gao, Deep learning for spacecraft pose estimation from photorealistic rendering, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 6007–6013.

[22] L. He, D. Rondao, N. Aouf, A novel mechanism for orbital AI-based autonomous refuelling, in: AIAA SCITECH 2023 Forum, 2023, p. 2211.

[23] N. Munasinghe, C. Le Gentil, J. Naylor, M. Asavkin, D.G. Dansereau, T. Vidal-Calleja, Towards event-based satellite docking: A photometrically accurate low-earth orbit hardware simulation, in: HERMES2 Workshop at ICRA2024, 2024.

[24] C.-H. Chien, K. Baker, Machine Vision for Relative Spacecraft Navigation During Approach to Docking, Technical Report, 2011.

[25] M. Kisantal, S. Sharma, T.H. Park, D. Izzo, M. Märtens, S. D'Amico, Satellite pose estimation challenge: Dataset, competition design, and results, IEEE Trans. Aerosp. Electron. Syst. 56 (2020) 4083–4098.

[26] K. Cosmas, A. Kenichi, Utilization of fpga for onboard inference of landmark localization in cnn-based spacecraft pose estimation, Aerospace 7 (2020).

[27] W. Ma, D. Zhou, Y. Hu, Z. He, Gknet: Graph-based keypoints network for monocular pose estimation of non-cooperative spacecraft, 2025, arXiv:2507.11077.

[28] B. Chen, J. Cao, A. Parra, T.-J. Chin, Satellite pose estimation with deep landmark regression and nonlinear pose refinement, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop, ICCVW, 2019, pp. 2816–2824, http://dx.doi.org/10.1109/ICCVW.2019.00343.

[29] D. Rondao, L. He, N. Aouf, AI-based monocular pose estimation for autonomous space refuelling, Acta Astronaut. 220 (2024) 126–140.

[30] V. Lepetit, F. Moreno-Noguer, P. Fua, Epnp: An accurate $\mathcal{O}(n)$ solution to the pnp problem, Int. J. Comput. Vis. 81 (2009) 155–166.

[31] M.A. Fischler, R.C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 4510–4520, http://dx.doi.org/10.1109/CVPR.2018.00474.

[33] Aitech, S-a1760 venus™ space AI gpgpu, 2023, https://www.aitechsystems.com/products/s-a1760-venus-space-ai-gpgpu/.

[34] NVIDIA Corporation, Nvidia jetson tx2i technical specifications and performance, 2023, https://developer.nvidia.com/embedded/jetson-tx2i.

[35] J.R. Forbes, Fundamentals of spacecraft attitude determination and control [bookshelf], IEEE Control Syst. Mag. 35 (2015) 56–58.
[36] J.-F. Shi, S. Ulrich, Uncooperative spacecraft pose estimation using monocular monochromatic images, J. Spacecr. Rockets 58 (2021) 284–301.
[37] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2002) 1330–1334.
[38] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.
[39] B.K. Horn, Closed-form solution of absolute orientation using unit quaternions, J. Opt. Soc. Amer. A 4 (1987) 629–642.
[40] K. Shoemake, Animating rotation with quaternion curves, in: Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, 1985, pp. 245–254.
[41] M. Lourakis, A. Argyros, Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? in: Tenth IEEE International Conference on Computer Vision, Vol. 1, Vol. 2, ICCV'05, 2005, pp. 1526–1531, http://dx.doi.org/10.1109/ICCV.2005.128.
[42] D. Scaramuzza, A. Martinelli, R. Siegwart, A flexible technique for accurate omnidirectional camera calibration and structure from motion, in: Fourth IEEE International Conference on Computer Vision Systems, ICVS'06, 2006, http://dx.doi.org/10.1109/ICVS.2006.3, 45–45.
[43] J.B. Kuipers, Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality, Princeton University Press, 1999.
[44] D.A. Vallado, Fundamentals of Astrodynamics and Applications, fourth ed., Microcosm Press, 2013.
[45] A. Van Oosterom, J. Strackee, The solid angle of a plane triangle, IEEE Trans. Biomed. Eng. (2007) 125–126.
[46] B.E. Saleh, M.C. Teich, Fundamentals of photonics, vol. 332, Wiley New York, 2008.
[47] OptiTrack, https://www.optitrack.com/. (Accessed October 2023).
[48] J.J. Craig, Introduction to Robotics: Mechanics and Control, third ed., Pearson Prentice Hall, Upper Saddle River, NJ, 2005.