Deep Learning Optical Navigation for Active Debris Removal of Deimos-1

Miguel Leiva[(1)], Manuel Sanjurjo-Rivo[(1)], Hodei Urrutxua[(2)], Ester Velázquez[(2)], Alfredo Escalante[(1)], Xin Chen[(2)]

[(1)] Department of Aerospace Engineering, Universidad Carlos III de Madrid
Leganes, Spain
Email: mleiva@pa.uc3m.es
[(2)] Fuenlabrada Engineering School, Universidad Rey Juan Carlos
Fuenlabrada, Spain
Email: hodei.urrutxua@urjc.es

***Abstract*–Active Debris Removal (ADR) missions are conditioned by close proximity orbital operations between a chaser spacecraft and a non-cooperative target. This single-vehicle responsive capability and inherent lack of inter-spacecraft communication translates into a set of extremely challenging requirements for the Guidance, Navigation and Control (GNC) system. In addition to computing an optimal trajectory and robustly controlling the platform, the GNC module must also autonomously deliver a real-time 6D pose estimation, performing reliably under demanding lighting conditions and complex target geometries. While no pre-existing classical relative navigation algorithm has been able to meet this set of stringent requirements, new Artificial Intelligence (AI) approximations are being tested due to their powerful performance within non-linear input scenarios, also complemented by the emergence of new synthetic dataset generation pipelines facilitated by powerful computing tools and high-fidelity software platforms. Accordingly, this document delves into the implementation of a *YOLOv8-pose* neural network architecture coupled with a *SQPnP+RANSAC* algorithm. Drawing from previous research lines, it introduces innovative elements such as the *Deimos-1* dataset generation pipeline and the common single-backbone neural network.**

I. INTRODUCTION

The ever-increasing presence of space debris in Earth's orbit poses a significant challenge to the sustainability of space activities. In this context, the growing threat of defunct satellites, spent rocket stages and assorted fragments require a joined endeavour to develop possible strategies to mitigate such risk. ADR missions are one of the main pillars in making the orbital environment more sustainable. Nonetheless, key technologies must still be developed and matured to successfully accomplish these kinds of missions.

Meanwhile, traditional algorithms designed for celestial body navigation [1] struggle to cope with the intricacies posed by factors such as target texture, geometry, and varying lighting conditions. While landmark-based techniques have been widely utilized in navigating celestial bodies, they encounter several limiting difficulties correlating 2D-3D correspondences in dynamic environments with limited target information.

Conversely, the integration of deep learning (DL) strategies has the potential to significantly improve the performance of classical algorithms in various ways. Firstly, they automatically learn relevant features from raw data, reducing the need for manual feature engineering [2]. Moreover, their inherent non-linearity enables DL models to capture complex relationships more effectively than traditional methods. Additionally, they scale well to large datasets and complex problems, thanks to parallel computing and distributed training techniques [3]. Lastly, transfer learning allows the adaptation of pre-trained models to new tasks or domains, thereby boosting performance even with limited data [4].

While previous papers from various research institutions have already addressed the topic, the evolving landscape of AI continually introduces new models and algorithms with enhanced performance, making it an area of ongoing interest and exploration. In spite of all the different paths to tackle this challenge, every one of them is trying to converge at the following model characteristics: to be capable of efficiently extracting the 6D coordinates of an unseen textureless object with minimal computing resources, even in challenging lighting conditions and subjected to partial occlusion. Moreover, the optimal model should rely on low-power sensors such as a single monocular RGB camera.

II. LITERATURE REVIEW

Close-range operations between two cooperative spacecrafts have been the corner stone of many space missions for decades, dating back to landmark projects like Apollo [5], the International Space Station (ISS), or Hubble repair missions [6]. However, non-cooperative scenarios for ADR or In-Orbit Servicing (IOS) are still under research [7][8]. Due to the previously described navigation challenges, these missions will have to carefully select a sensor package suited for the task, where monocular and infrared cameras, if compared to

LIDAR solutions, provide faster pose computation and less hardware complexity, but at the cost of more algorithmic complexity [9].

In particular, traditional image-based navigation systems for close-range operations typically employ iterative image processing algorithms and heavily rely on accurate initialization [10] or hand-engineered features [11][12][13]. Efforts have been made to address initialization challenges [14], but these methods still necessitate markers on the target spacecraft and struggle with changing lighting conditions.

By using specific neural network architectures, most of the shared limiting factors can be solved, however, these improvements are obtained at the cost of hardware requirements, data-intensive processes and less explainability. For instance, recent AI development around 6D pose estimation has managed to significantly improve performance metrics while reducing training data requirements. However, the new evolution of algorithms is generally focused on benchmarking, trading off accuracy over ease of implementation, reasonable inference times and hardware requirements [15][16]. Specifically, while FoundationPose [15] stands out for its impressive performance and efficient inference times, it still requires robust hardware capabilities and access to depth information, posing potential challenges in orbital operations. Consequently, choosing a deep learning algorithm tailored for space applications crucially requires striking a balance between performance metrics and available computing resources.

Following these requirements, a small number of research institutions have tackled the challenge from two high level approximations: direct and indirect methods. In particular, the first approach aims to directly map the input space into a 6D pose vector, process that can be accomplished through different classification and regression CNN architectures, where Sharma's SPN [17] method and Gao [18] serve as prime examples. In addition, D'Amico's research group contributed to the challenge by creating the open-source SPEED dataset [19] and allowing the public to improve the benchmark results during the Pose Estimation Challenge [20]. However, the competition was top-scored by the other kind of architecture, where the indirect method is primarily focused on deep learning object/keypoint detection and applying afterwards a classical 2D-3D correspondence method to extract the 6D coordinates. Importantly, these results showed that, even though direct methods are the final goal, indirect approximations are still the optimal solution for low power challenging applications. Park [21] and Chen [22] serve as the prime publications, been further improved by Li [23].

III. Problem Formulation

Crafting an accurate deep learning 6D pose estimation algorithm for space operations is a data-intensive endeavour with several detailed and individual steps. During this process, the Deimos-1 dataset generation pipeline will have to be tightly coupled with the previously selected model and derived training and validation performances, all while being representative of the spacecraft, its environment and the lighting conditions the model will encounter once deployed.

Deimos 1	
Mass (kg)	89
Shape (m)	Cubical: 0.63 x 0.66 x 0.64
Orbit	SSO: p = 658.1 km a = 660.3 km i = 97.7 º T = 97.7 min
Attitude State	N/A

Table 1. Deimos-1 mission parameters.

Lastly, it will be crucial to integrate the developed algorithm into a forthcoming navigation function capable of combining multi-sensory inputs to compute an enhanced accuracy. This upgraded functionality must undergo rigorous testing, including software-in-the-loop and hardware-in-the-loop processes. Therefore, compatibility of the deep learning architecture with simulation software and laboratory equipment is paramount to ensure seamless integration and reliable performance.

IV. Deep Learning Model Selection

Following the literature review, selecting an appropriate deep learning model involves identifying an architecture suited for the task rather than prioritizing pure performance. This is done by considering factors like complexity, computational efficiency, accuracy, and integration feasibility with specific hardware. Consequently, our selected model leans towards space-oriented methodology, particularly favouring indirect methods.

Despite the use of an already well-established architecture, improvements over previous research projects can be anticipated due to the enhanced performance of recent object and keypoint detection neural networks. Specifically, the chosen deep learning models will have to do two things:

1. Object Detection: ability to identify and locate a specific object in an image, precisely drawing a 2D bounding box that limits the object's area. Current benchmark in **Table 2**.

2. Keypoint Detection: ability to identify and 2D locate a specific feature of the pre-detected object. Current benchmark in **Table 3.**

Model	Input (px)	Inference Time (ms)		Parameters (M)	AP5095
		CPU (ONNX)	GPU (A100)		
Co-DETR	800	-	35	348	66.0
InternImage-H	800	-	1.8	2180	65.5

Table 2. Object detection top performing models.

Model	Input (px)	Inference Time (ms)		Parameters (M)	AP50
		CPU (ONNX)	GPU (A100)		
HigherHRNet (W32)	640	-	300	28.6	87.1
DEKR	640	-	300	65.7	88.3

Table 3. Keypoint detection top performing models.

Before selecting a specific model based on individual performances, one must be aware that indirect space-oriented architectures traditionally handled object and keypoint detection separately, first identifying the object and then extracting keypoints within the preestablished 2D bounded area. However, recent advancements in network architectures have enabled the consolidation of both strategies into a single model, leveraging shared convolutional layers and filters to save computational resources. Particularly, a prominent example is the *YOLO-pose* architecture [24], offering versatile performance configurations depending on the number of parameters (**Table 4**). In essence, by integrating object and keypoint detection layers, this model delivers fast and precise bounding boxes and keypoint information.

Model	Input (px)	Inference Time (ms)		Parameters (M)	AP50	AP5095
		CPU (ONNX)	GPU (A100)			
YOLOv8n-pose	640	131.8	1.18	3.3	80.1	37.3
YOLOv8s-pose	640	233.2	1.42	11.6	86.2	44.9
YOLOv8m-pose	640	456.3	2.00	26.4	88.2	50.2
YOLOv8l-pose	640	784.5	2.59	44.4	90.0	52.9
YOLOv8x-pose	640	1607.1	3.73	69.4	90.2	53.9

Table 4. YOLOv8-pose model summary.

Moreover, the reduced parameter count significantly lowers training requirements compared to existing keypoint detection benchmarks. Specifically, in the realm of object detection, high-performance neural networks often entail substantial parameter counts and prolonged inference times. However, given the relatively consistent background scenarios in orbital environments, top-tier models are not essential for providing a reliable solution. Consequently, *YOLO-pose* emerges as the preferred model for both object and keypoint detection, particularly its *YOLOv8n-pose* and *YOLOv8s-pose* variants. As, despite their smaller size, they exhibit minimal performance drop compared to medium and large versions while requiring significantly fewer parameters and training resources. In the following sections, both model's performance will be compared to assess if such dimension difference has enough effect to consider a bigger neural network.

V. 2D-3D Correspondence Method

Selecting an indirect approach requires developing an end-to-end architecture that is capable of transforming the neural network's 2D output into a 6D vector that displays the relative position between the camera and the object. This nonlinear correspondence process has been a focal area in computer vision for decades, with only a handful of algorithms capable of efficiently executing it:

1. *P3P*: limited to three keypoints correspondence, allowing a fourth under noise [25].
2. *EPnP*: tackles n>= 4 keypoint problems, refining solutions via Gauss-Newton method [26].
3. *SQPnP*: solved through sequential quadratic programming [27].

Given *Deimos-1*'s selected number of keypoints (10), *P3P* is unsuitable. On the other hand, *SQPnP* surpasses *EPnP* in efficiency and accuracy. However, most solvers still struggle with keypoint outliers, often requiring additional methods. Consequently, prior projects have engaged in incorporating a voting scheme, with Random Sample Consensus (*RANSAC*) emerging as the predominant choice. This algorithm iteratively selects subsets from data to identify the best-fitting model, making it a robust choice for estimating model parameters in various fields.

The following **Fig.1**, **Fig.2** and **Fig.3** display how the integration of a *RANSAC* voting scheme can significantly improve, on average, the 2D-3D correspondence problem if compared to a single PnP (*SQPnP*) algorithm. By performing a Monte Carlo analysis with varying pixel-noise levels to a range of keypoints, it is possible to assess the impact on angular and translational errors.

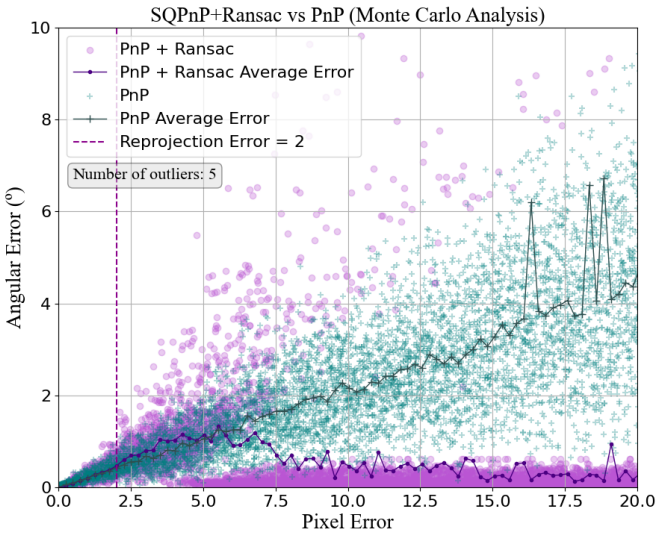

Fig. 1. PnP rotational improvement with RANSAC. Monte Carlo analysis for 5 outliers.

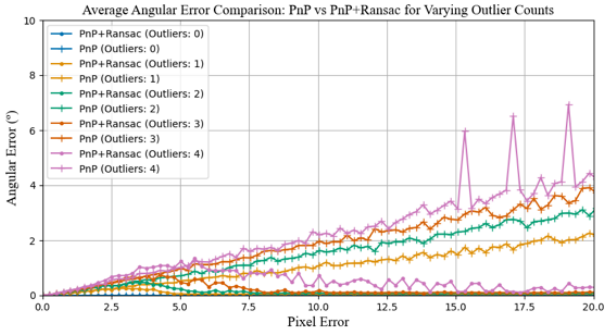

Fig. 2. PnP rotational improvement with RANSAC. Monte Carlo analysis for 0 to 4 outliers.

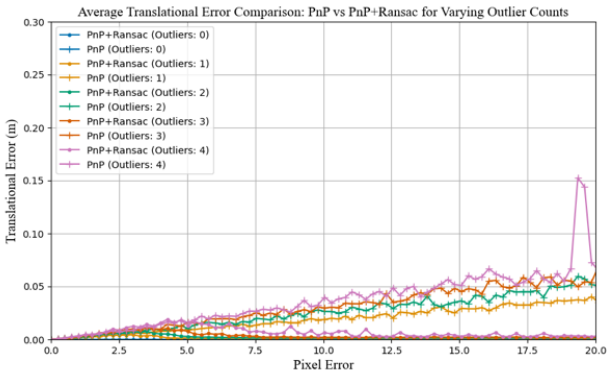

Fig. 3. PnP translational improvement with RANSAC. Monte Carlo analysis for 0 to 4 outliers.

The results correspond to a number of ten keypoints specific of the *Deimos-1* spacecraft, with a reprojection error (maximum pixel error to be considered an outlier) of 2 pixels. However, future work will require fine-tuning to select the precise reprojection value.

VI. Dataset Generation Process

The effectiveness of data-driven algorithms is directly correlated with the quality and similarity of the training data to the final operational scenario. Therefore, acquiring a dataset that closely resembles the target environment is essential for achieving optimal algorithmic performance.

However, in the realm of orbital operations, datasets featuring thousands of pose-annotated images of spacecrafts are notably scarce or almost non-existent. Consequently, researchers are forced to resort to synthetic datasets that replicate the lighting conditions and surface characteristics of the target spacecraft. To create such datasets, the chosen 3D design framework must additionally offer a software interface to automate the creation of images and annotation of information. This particular combination of high-fidelity scene creation and automated dataset generation considerably narrows down the selection of viable tools. As a result, thanks to its ease of 3D model creation, extensive fine-tuning capabilities, open-source nature, and Python interface, *Blender* [28] emerges as the preferred software solution for this project's early stage. However, future works will likely rely on higher fidelity but more complex graphic engines like *UnrealEngine5* [29].

A. Spacecraft 3D modelling

During the 3D modelling process, design emphasized external visual representation over internal CAD-like features in order to streamline the design process. To be specific, the main objective was to recreate the material and surface characteristics of the object that will be encountered in orbit. This process was initially estimated by comparing the results with *SPEED* [19] dataset and other satellite in-orbit imagery, but future improvements based on higher fidelity software and real laboratory images will significantly improve the output.

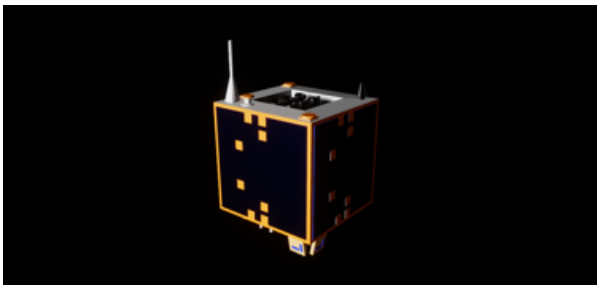

Fig. 4. Deimos-1 3D model

B. Scene

Once an accurate representation of the target is created, the next step involves developing a precise model of the lighting conditions and background that will be encountered by the camera throughout its operational lifetime. This process requires a thorough assessment of the specific mission profile and orbital parameters. Following **Table 1**, the selected spacecraft will follow a LEO SSO orbit ranging altitudes around 660 km, being subjected to rapidly changing lighting conditions and planet Earth as posible background, including all its different textures and surface or atmospheric conditions.

B.1. Lighting Conditions

Balancing realism with the neural network's feature inference capabilities poses a significant challenge in determining appropriate lighting conditions. While neural networks excel at extracting features and discerning complex relationships, relying solely on RGB data imposes limitations on available information. Thus, lighting scenarios must be carefully constrained within the network's feature extraction capability. Achieving this balance entails iteratively testing the network's limits and establishing boundaries to ensure effective processing and analysis of images.

During the dataset generation process, primary light sources will encompass the Sun, Earth's albedo, and supplementary 360º low-power lighting sources. Strategically positioned, these additional sources prevent the spacecraft from being entirely obscured when the camera, Sun, and Earth relative orientation allows. They are also low power enough to avoid over-exposure under different conditions. In addition, the resulting images will be black and white graded to reduce image size and colour effects, forcing the model to obtain a geometrical representation of the spacecraft.

B.2. Earth Background

Despite neural networks' impressive ability to identify objects in complex environments, they still require specific training to excel in such tasks. Without incorporating an Earth background during training, neural networks may struggle to distinguish the target spacecraft, even if they perform well in other conditions.

In the *SPEED* [19] dataset, pose-annotated images were initially created, followed by the addition of an Earth background to selected images. However, leveraging the abundance of high-definition satellite imagery available, one can directly create a high-fidelity representation of the planet within the dataset generation process, eliminating the need for additional post-processing. Nonetheless, accurately representing the Earth is not limited to spherically shaping satellite imagery, it also involves incorporating surface reflectivity, atmospheric diffraction effects, and detailed textures to ensure a realistic representation.

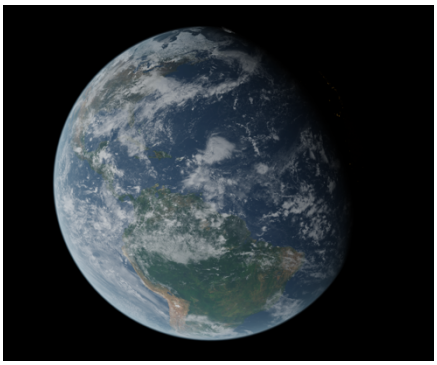

Fig. 5. High-fidelity Earth model.

C. Camera parameters

While object and keypoint detection DL models are not very sensible to camera parameters, they only provide the initial solution for the end-to-end architecture. This 2D measurement still requires a 3D correspondence method to compute the 6D solution based on the real coordinate transformation between the sensor reading and the physical space. To do this computation, the intrinsic camera parameters are fundamental, as every lens and sensor records the scene in a different manner.

Focal length	30 mm
Centre	cx = cy = 320
Distortion Parameters	0
Sensor Dimensions	32 mm x 32 mm
Resolution	640 x 640 pixel

Table 5. Dataset camera parameters.

In addition, the size and shape of the image are dictated by the deep learning model as well as the available hardware, with high-resolution imagery often not providing additional information due to the model's incorporation of filters previous to the convolutional ones, where they scale down high-resolution dimensions to fit the constrained input space, set at 640x640 pixels for the selected model.

D. Dataset Generation pipeline

The dataset generation process aims to autonomously create an image portfolio that manages to cover all the target's surfaces, pictured under representative lighting conditions and background scenarios, all while guaranteeing that the acquired information is within the model's ability to compute its committed task. To do this, the camera will have to move around the target and cover all areas. However, one must find an equilibrium between surface coverage and dataset size, as a fine angular discretization can rapidly scale towards hundreds of thousands of images.

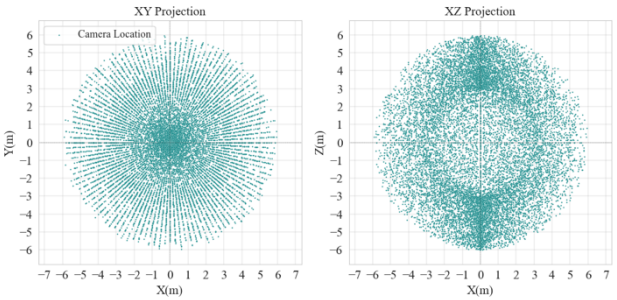

Fig. 6. Camera coordinate distribution.

Meanwhile, in order to avoid the generation of poorly illuminated images that could hinder the deep learning model's ability to derive meaningful solutions, it is necessary to control the relative angular distribution between the Sun, Earth, camera, and target.

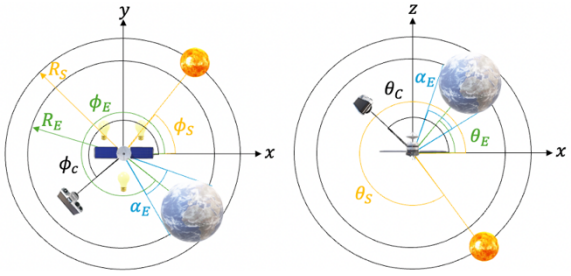

Fig. 7. Scene angular distribution.

Where all the objects move around the target following an azimuth/elevation distribution that guarantees the stipulated constraints for each individual image while still being uniformly distributed along the dataset. In addition, the introduction of supplementary low-power light sources can address unavoidable over-shadow conditions, forcing the pipeline to produce high-fidelity images while ensuring that the neural network has sufficient information for inference.

The pseudocode displayed in **Fig.8** gathers the followed image generation pipeline, where the basic principle is that the camera moves around the stationary spacecraft, covering all surface areas in a grid of $\Delta\phi_c \cdot \Delta\theta_c$ accuracy. Once the camera is placed at an exact location, a rotation matrix to point the camera's $\bar{z}_c$ axis at the target is applied, followed by a bounded random 3D translation and rotation around such axis to increase the dataset variance. Then, the background scenario is set by randomly moving the Earth. This operation is performed by applying a certain azimuth and elevation to the given fixed distance, followed by a random body rotation matrix to change the background image of the Earth. Upon determining this configuration, the Sun is positioned in a randomly-constraint way to avoid eclipse scenarios, ensuring it remains unobstructed by the Earth. This process entails limiting the azimuth range to ensure this condition is met and the ϕ_c/θ_c pair never enters the Earth's α_E generated cone.

Dataset Generation Pipeline

$\boldsymbol{\phi_c} = range[0, 2\pi, \Delta\phi_c]$
$\boldsymbol{\theta_c} = range[-\pi/2, \pi/2, \Delta\theta_c]$
for *i in* ϕ_c*:*
  for *j in* θ_c*:*
    Camera Pose
    $\boldsymbol{D_c} = random.uniform[d_{min}, d_{max}]$ *# Compute distance from target*
    $\bar{\boldsymbol{X}}_c$ *# Compute camera coordinates in target's reference frame*
    $\bar{\boldsymbol{X}}_c = \bar{X}_c + \bar{t}_{random}$ # Apply bounded random rotation to off-centre target
    $\bar{\boldsymbol{R}}_c$ *# Compute rotation matrix to aim the camera at the target*
    $\bar{\boldsymbol{R}}_c = \bar{R}_c \cdot \bar{R}_{random,\bar{z}_c}$ # Apply random rotation around camera pointing axis

    Earth Pose
    $\boldsymbol{\phi_E} = random.uniform[0, 2\pi]$
    $\boldsymbol{\theta_E} = random.uniform[-\pi/2, \pi/2]$
    $\bar{\boldsymbol{R}}_E = \bar{R}_E \cdot \bar{R}_{random}$ # Apply random background change

    Sun Pose
    $\boldsymbol{\phi_S} = random.uniform[\phi_E + \alpha_e/2, 2\pi + \phi_E - \alpha_e/2]$ # Avoid Earth's eclipse
    $\boldsymbol{\theta_S} = random.uniform[-\pi/2, \pi/2]$

Fig. 8. Dataset generation pipeline pseudocode.

It is worth noting that the image generation process utilizes scaled dimensions rather than real-world measurements, allowing high fidelity while optimizing computational resources. **Table 6** has the established measures for generating the *Deimos-1* dataset, where the only real-size transformation are applied to the camera parameters and target dimensions. However, the Earth is proportionally dimensioned both in radius and relative distance with the spacecraft d_E.

$\Delta\phi_c$ (°)	3
$\Delta\theta_c$ (°)	1.5
d_{min} (m)	4
d_{max} (m)	6
α_E (°)	45
R_E (m)	1200
d_E (m)	1300
Nº Images	14400

Table 6. Dataset generation dimensions.

After applying the described procedure to the established dimensions, one is left in **Fig. 9** with the following dataset coordinate distribution:

Fig. 9. Dataset instances coordinate distribution.

Displaying the *Deimos-1*, Earth, and Sun in a uniformly distributed composition.

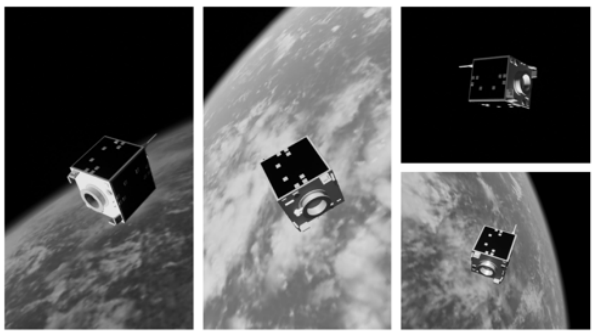

Fig. 10. Various Deimos-1 dataset samples.

Another crucial aspect during dataset generation is the

precise annotation of all data, which constitutes a time-consuming task, as each spatial transformation must be precisely computed and applied to the predetermined keypoints, ensuring that the neural network receives accurate information regarding the expected position of each keypoint for the given pose.

Additionally, the bounding box process is also labour-intensive. The a priori simple geometry of the object but filled with superficial protuberances of various shapes makes determining its boundaries heuristically a challenging task. To address this topic, we've adopted an approach involving the creation of a parallel black and white dataset devoid of background interference. Subsequently, a post-processing pipeline is applied to transform the grayscale images into pure black and white, followed by a filtering step that automatically extracts the object boundaries. This methodology helps the bounding box annotation process and ensures accuracy in identifying the object's boundaries.

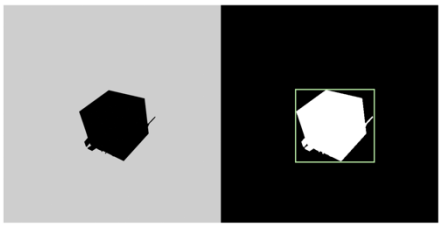

Fig. 11. Bounding box generator process.

The final stage concerning the generated images involves data augmentation, wherein a set of controlled transformations is applied to the images. This process aims to enhance the neural network's ability to generalize by exposing it to a broader range of image characteristics beyond those present in the training dataset. The transformations applied include the following:

1. HSV Augmentation: Hue, Saturation, and Value augmentation are performed on the image.
2. Image Translation: Random translation is applied.
3. Image Scale: Random scaling of the image.
4. Horizontal Image Flip: The image is flipped horizontally with a certain probability.
5. Mosaic Augmentation: It involves combining multiple images into a single training sample.

VII. TRAINING

Once the dataset is created, one must start the training process to validate afterwards both the model's performance and the created images. However, the approach to this process can vary significantly based on the hardware resources at hand. CPUs are not viable for datasets of this size, while low-power GPUs may face challenges with extended training times, particularly when they have limited memory, resulting in reduced batch sizes and more gradient updates, potentially impacting convergence.

Following the training results in **Fig. 12**, the *YOLOv8n-pose* model demonstrated progressive improvement during training and further learning capacity over 300 epochs. However, validation results clearly indicate an overfitting behaviour beyond this threshold, suggesting that the model reached its full capacity for the given dataset, rendering further training unnecessary. As for the selected hyperparameters, they facilitated rapid convergence and stable training, particularly owing to the utilization of a large batch size of 128.

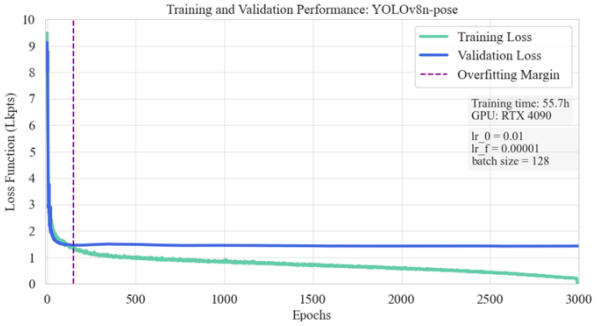

Fig. 12. YOLOv8n-pose training process.

In **Fig. 13**, the larger *YOLOv8s-pose* architecture exhibited a performance comparable to that of the smaller model.

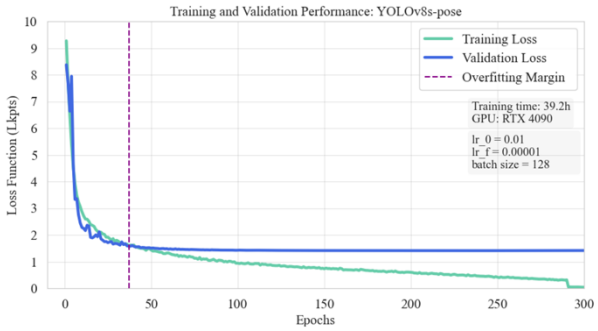

Fig. 13. YOLOv8s-pose training process.

While it achieved superior results with the training batch, it failed to enhance the validation samples outcome, thus perpetuating the overfitting issue.

Hence, despite the smaller size, faster training times, and lower computational burden of the *YOLOv8n-pose* architecture, one should expect minimal discrepancies between the two models for the given *Deimos-1* dataset. As a result, the following analysis will be assessed through this reduced neural network.

VIII. RESULTS

Finally, now that both the deep learning model and 2D-3D correspondence method have been selected and trained, this section will integrate both algorithms in order to translate an input image into the corresponding 2D space and expected final 6D vector.

The integrated model validation will cover two phases:

1. Evaluation of the pose estimation performance within the validation set of the *Deimos-1* dataset, providing a physical magnitude to the loss function final results.

2. Evaluation of out-of-distribution trajectory, examining the model's performance concerning two specific trajectories: one characterized by optimal lighting conditions slightly divergent from those in the dataset, and a suboptimal one.

1. Validation set

Fig. 14 and **Fig. 15** demonstrate that the average error, coupled with the obtained standard deviation, meets the criteria for integration into a navigation function. This is because the operational scenario typically involves low rotational rates, which, when combined with additional sensor measurements and a Kalman filter, render a 3° error sufficiently accurate for effective navigation.

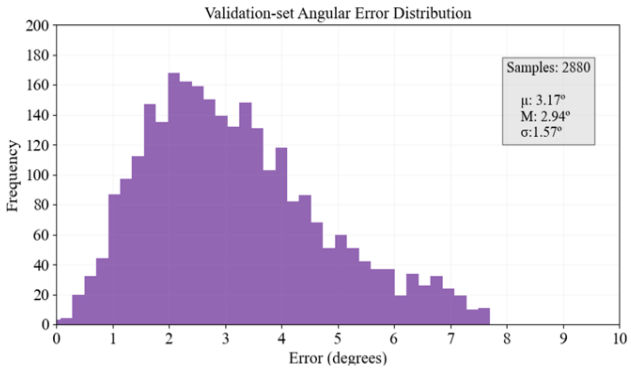

Fig. 14. Deimos-1 validation-set angular error.

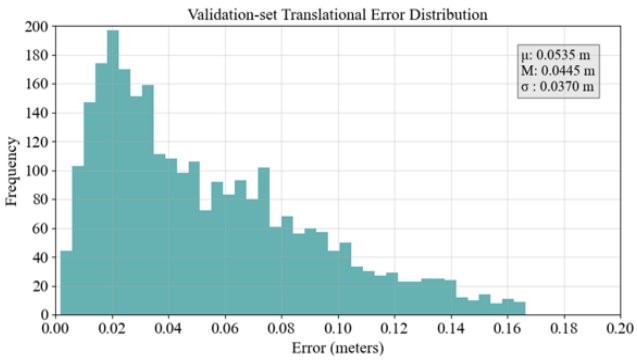

Fig. 15. Deimos-1 validation-set translational error.

However, it is observed that there is a higher variance in angular computation compared to translational one. This phenomenon is often encountered due to the relatively low noise levels in the inferred keypoints in relation to the overall dimensions of the spacecraft. Furthermore, the non-linear relationships involved in inferring the angular component contribute to its higher variance.

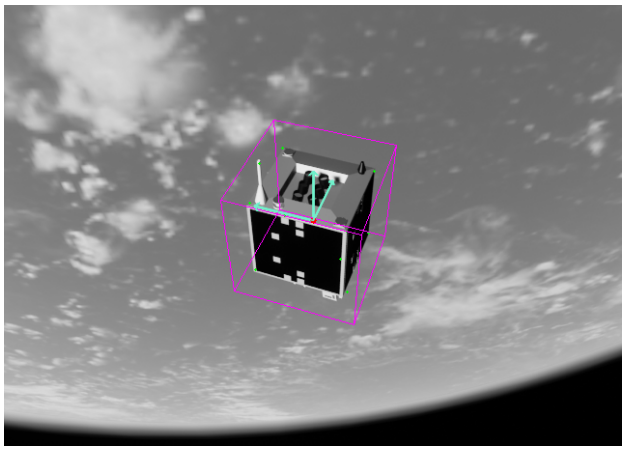

Fig. 16. Sample pose estimation visualization.

2. Trajectory 1: nominal lighting conditions

This trajectory was designed in a way that the relative position between the sun, spacecraft and camera orientation allowed the neural network to infer the relative pose with adequate margin. The specific light orientation was able to illuminate key features of the spacecraft while the power was bounded in a way that didn't overexpose the image.

If **Fig. 17** is compared to the validation set performance, one can observe a 1.1° average degradation in inference accuracy coupled with a wider error distribution of +1.08 standard deviation. However, the results are still within a sensible margin.

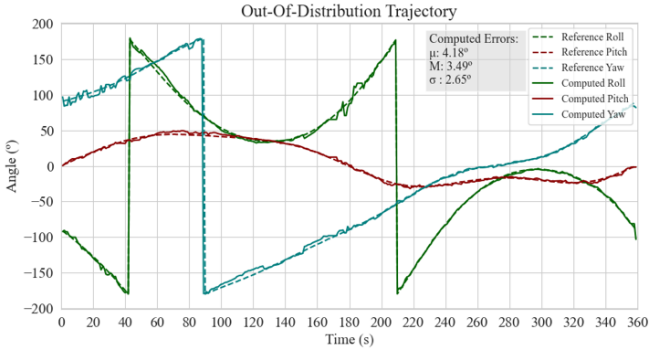

Fig. 17. Optimal trajectory pose estimation.

3. Trajectory 2: Suboptimal lighting conditions

This trajectory was specifically crafted to test the model's capabilities and evaluate its performance under challenging lighting conditions that deliberately induced occlusion.

Assessing the model's performance under these conditions provides valuable insights. In **Fig. 18**, while overall performance remains comparable to that of the previous scenario, there are instances where the model struggles to compute a solution, particularly in specific angular relationships. Further analysis reveals that this error stems from a geometric inconsistency exacerbated by the lighting conditions and the symmetrical nature of *Deimos-1*. These factors combine to obscure the few discernible geometric features crucial for the neural network to infer a valid solution within the almost-symmetrical spacecraft.

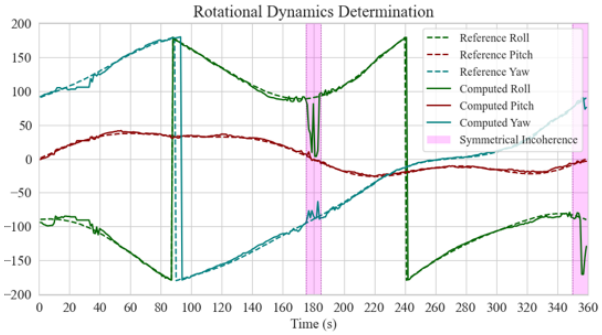

Fig. 18. Suboptimal trajectory pose estimation.

IX. Conclusion

In summary, the established models lay a solid foundation for future enhancements in space-oriented 6D pose estimation. Despite challenging orbital conditions and a difficult target, the current performance shows significant promise, indicating the potential for accurate pose estimation with low computational requirements.

Moving forward, we plan to assess the accuracy of our models under improved and more realistic material surface lighting conditions and enhanced camera effects. This next phase will involve integrating real laboratory images, which are crucial for the generation of high-fidelity datasets and validating with hardware in the loop.

Additionally, while our current architecture demonstrates notable efficiency, one should expect that ongoing advancements in hardware capabilities will enable the integration of larger and more powerful models. Furthermore, given the dynamic landscape of artificial intelligence and computer vision, it is essential to continually monitor existing benchmarks and literature. This proactive approach ensures that we remain up-to-date with the latest advancements, preventing the adoption of outdated architectures and allowing for periodic adjustments to the project's scope as needed.

Finally, our model lays the groundwork for crafting a sophisticated multisensory navigation system designed specifically for Autonomous Rendezvous and Docking (ADR) missions within the LEO environment. Through the integration of various sensor inputs, this navigation system promises superior performance in manoeuvring spacecraft in close proximity operations.

X. References

[1] Norman, C. D., Christopher J. Miller, R. D. Olds, Courtney Mario, Eric Edward Palmer, O. S. Barnouin, Michael G. Daly, John R. Weirich, Jeffrey A. Seabrook, Carina A. Bennett, Bashar Rizk, Brent J. Bos and Dante S. Lauretta. "Autonomous Navigation Performance Using Natural Feature Tracking during the OSIRIS-REx Touch-and-Go Sample Collection Event." *The Planetary Science Journal 3* (2022): n. pag.

[2] LeCun, Yann, Léon Bottou, Yoshua Bengio and Patrick Haffner. "Gradient-based learning applied to document recognition." Proc. *IEEE* 86 (1998): 2278-2324.

[3] Hegde, Vishakh and Sheema Usmani. "Parallel and Distributed Deep Learning." (2016).

[4] Zhuang, Fuzhen, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong and Qing He. "A Comprehensive Survey on Transfer Learning." Proceedings of the *IEEE* 109 (2019): 43-76.

[5] Langley, R. D.. "Apollo experience report: The docking system." (1972).

[6] Zimpfer, Douglas J., Peter M. Kachmar and Seamus T. Tuohy. "Autonomous Rendezvous, Capture and In-Space Assembly: Past, Present and Future." (2005).

[7] Forshaw, Jason L., Guglielmo S. Aglietti, Nimal Navarathinam, Haval Kadhem, Thierry Salmon, Aurélien Pisseloup, Éric Joffre, Thomas Chabot, Ingo Dr. Retat, Robert Axthelm, Simon Barraclough, Andrew Ratcliffe, Cesar Bernal, François Chaumette, Alexandre Pollini and Willem Herman Steyn. "RemoveDEBRIS: An in-orbit active debris removal demonstration mission." *Acta Astronautica*

127 (2016): 448-463.

[8] Hu, Qinglei, Wei Chen and Lei Guo. “Fixed-Time Maneuver Control of Spacecraft Autonomous Rendezvous With a Free-Tumbling Target.” *IEEE Transactions on Aerospace and Electronic Systems* 55 (2019): 562-577.

[9] Ventura, Jacopo. “Autonomous Proximity Operations for Noncooperative Space Targets.” (2016).

[10] Opromolla, Roberto, Giancarmine Fasano, Giancarlo Rufino and Michele Grassi. “Pose Estimation for Spacecraft Relative Navigation Using Model-Based Algorithms.” *IEEE Transactions on Aerospace and Electronic Systems* 53 (2017): 431-447.

[11] Cropp, Alexander. “Pose estimation and relative orbit determination of a nearby target microsatellite using passive imagery.” (2001).

[12] Liu, Chang and Weiduo Hu. “Relative pose estimation for cylinder-shaped spacecrafts using single image.” *IEEE Transactions on Aerospace and Electronic Systems* 50 (2014): 3036-3056.

[13] Sharma, Sumant and Simone D’Amico. “Comparative assessment of techniques for initial pose estimation using monocular vision.” *Acta Astronautica* 123 (2016): 435-445.

[14] Benn, Mathias. “Vision Based Navigation Sensors for Spacecraft Rendezvous and Docking.” (2011).

[15] Wen, Bowen, Wei Yang, Jan Kautz and Stanley T. Birchfield. “FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects.” *ArXiv* abs/2312.08344 (2023): n. pag.

[16] Nguyen, Van Nguyen, Thibault Groueix, Mathieu Salzmann and Vincent Lepetit. “GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence.” *ArXiv* abs/2311.14155 (2023): n. pag.

[17] Sharma, S.; Beierle, C.; D’Amico, S. “Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks.” In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2019; *IEEE*: New York, NY, USA, 2018; pp. 1–12.

[18] Proença, P.F.; Gao, Y. “Deep learning for spacecraft pose estimation from photorealistic rendering.” In Conference on Robotics and Automation, Online, 31 May 2020; *IEEE*: New York, NY, USA, 2020; pp. 6007–6013.

[19] Park, T.H., Märtens, M., Lécuyer, G., Izzo, D., & D’Amico, S. (2021). “SPEED+: Next-Generation Dataset for Spacecraft Pose Estimation across Domain Gap.” 2022 *IEEE Aerospace Conference* (AERO), 1-15.

[20] https://kelvins.esa.int/satellite-pose-estimation-challenge/leaderboard/leaderboard

[21] Park, Tae Ha, Sumant Sharma and Simone D’Amico. “Towards Robust Learning-Based Pose Estimation of Noncooperative Spacecraft.” *ArXiv* abs/1909.00392 (2019): n. pag.

[22] Chen, Bo, Jiewei Cao, Álvaro Parra and Tat-Jun Chin. “Satellite Pose Estimation with Deep Landmark Regression and Nonlinear Pose Refinement.” 2019 *IEEE/CVF International Conference on Computer Vision Workshop* (ICCVW) (2019): 2816-2824.

[23] Li, Kecen, Haopeng Zhang and Chenyu Hu. “Learning-Based Pose Estimation of Non-Cooperative Spacecrafts with Uncertainty Prediction.” *Aerospace* (2022): n. pag.

[24] Maji, Debapriya, Soyeb Nagori, Manu Mathew and Deepak Poddar. “YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss.” 2022 *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW) (2022): 2636-2645.

[25] Geng, Qinghua and Weiming Liu. “Very Robust Direct Solver of the P3P Problem.” 2019 6th *International Conference on Systems and Informatics* (*ICSAI*) (2019): 1378-1382.

[26] Lepetit, V.; Moreno-Noguer, M.; Fua, P. (2009). "EPnP: An Accurate O(n) Solution to the PnP Problem". *International Journal of Computer Vision*. 81 (2): 155–166. doi:10.1007/s11263-008-0152-6. hdl:2117/10327. S2CID 207252029.

[27] Terzakis, George and Manolis I. A. Lourakis. “A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem.” *European Conference on Computer Vision* (2020).

[28] Blender. (n.d.). Retrieved from https://www.blender.org/

[29] Unreal Engine 5. (n.d.). Retrieved from https://www.unrealengine.com/es-ES/unreal-engine-5