DEEP LEARNING-BASED MONOCULAR RELATIVE POSE ESTIMATION OF UNCOOPERATIVE SPACECRAFT

M. Piazza[(1)], **M. Maestrini**[(2)], **and P. Di Lizia**[(2)]

[(1)]*Infinite Orbits Pte. Ltd., 30-00 High Street Centre, 1 North Bridge Road, Singapore 179094, Singapore, Email: massimo@infiniteorbits.io*

[(2)]*Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy, Email: {michele.maestrini, pierluigi.dilizia}@polimi.it*

ABSTRACT

This paper aims to present the design and validation of a deep learning-based pipeline for estimating the pose of an uncooperative target spacecraft, from a single grayscale monocular image.
The possibility of enabling autonomous vision-based relative navigation in close proximity to a non-cooperative space object has recently gained interest in the space industry. In particular, such a technology would be especially appealing for Active Debris Removal (ADR) missions as well as in an on-orbit servicing scenario. The use of a simple camera, compared to more complex sensors such as a LiDAR, has numerous advantages in terms of lower mass, volume, and power requirements. This would translate into a substantially cheaper chaser spacecraft, at the expense of increased complexity of the image processing algorithms.
The Relative Pose Estimation Pipeline (RPEP) proposed in this work leverages state-of-the-art Convolutional Neural Network (CNN) architectures to detect the features of the target spacecraft from a single monocular image. Specifically, the overall pipeline is composed of three main subsystems. The input image is first of all processed using an object detection CNN that localizes the portion of the image enclosing our target, i.e. the bounding box. This is followed by a second CNN that regresses the location of semantic keypoints of the spacecraft. Eventually, a geometric optimization algorithm exploits the detected keypoint locations to solve for the final relative pose, based on the knowledge of camera intrinsics and of a wireframe model of the target satellite.
The Spacecraft PosE Estimation Dataset (SPEED), a collection of 15300 images of the Tango spacecraft released by the Space rendezvous LABoratory (SLAB), has been used for training the deep neural networks employed in our pipeline, as well as for evaluating performance and estimation uncertainty. The proposed RPEP pipeline guarantees on SPEED a centimeter-level position accuracy and degree-level attitude accuracy, along with considerable robustness to changes in illumination and background conditions. In addition, our architecture showed to generalize well on real images, despite having exclusively exploited synthetic data to train the CNNs.

Keywords: pose estimation; uncooperative spacecraft; Active Debris Removal; deep learning; CNN; object detection; landmark regression, EPnP.

1. INTRODUCTION

1.1. Problem statement & motivation

The problem that will be tackled in this work is that of estimating the relative pose of an uncooperative spacecraft (S/C) from a single grayscale monocular image, in a close-proximity operations scenario. The term "uncooperative" is here referred to as a situation in which the target S/C is not equipped with supportive means (e.g. light-emitting markers) nor is capable of establishing a communication link. The satellite is modeled as a rigid body, which means that its six-dimensional pose space is defined in terms of 3 translation components and 3 attitude components, relative to the chaser S/C.

For estimating the pose of an uncooperative spacecraft relative to another satellite, two main approaches are possible. The motion of a space object might be in principle estimated by means of ground-based tracking. However, such an estimate would be affected by significant uncertainty[1] and its availability would depend on the target's visibility from ground. This renders this approach unsuitable for close-proximity operations between two spacecrafts. The second approach consists in estimating the pose of the target directly onboard the chaser S/C, by exclusively relying on the sensors available on the latter. This currently represents the only strategy that is suitable for close-proximity operations.
A possible choice to achieve onboard pose estimation may be the use of LiDAR and/or stereo camera sensors, which, nevertheless, can be extremely expensive and represent a substantial contribution to the mass and power budgets of the S/C.
In contrast, monocular cameras are characterized by

[1] even the use of advanced bi-static laser tracking techniques would still result in an unsuitable position uncertainty of about 20 m for an uncooperative object [1]

lower complexity and their use for autonomous relative navigation would translate into significant savings in terms of cost, mass, and power requirements. All these benefits come at the expense of very high complexity image processing algorithms. Besides, monocular sensors are characterized by weaker robustness to lighting conditions and variable backgrounds, compared to a LiDAR. This aspect is particularly challenging, given the low signal-to-noise ratio that characterizes spaceborne optical images.
Nonetheless, the advantages of using a monocular camera as a navigation sensor make it an appealing possibility, especially within the framework of on-orbit servicing and Active Debris Removal (ADR) missions.
Among the missions of this kind, that are slated for launch during the next few years, an example might be NASA's *Restore-L* mission [14], whose launch date is currently set for December 2023, along with the commercial servicing programs proposed by companies like Infinite Orbits and Astroscale. Also, the first-ever ADR mission, *ClearSpace-1* [6], is expected to launch in 2025. It is then clear that the ability to accurately estimate the pose of an uncooperative S/C, by relying on hardware with minimal complexity, represents a key enabling technology in all the aforementioned scenarios.

In the remainder of this section, we will present the state-of-the-art techniques for S/C pose estimation and we will also describe the Spacecraft PosE Estimation Dataset (SPEED) dataset that has been used to validate our algorithms.
Subsequently, in Section 2 the architecture proposed in this work is explained in detail.
Then, in Section 3 various performance metrics of our pipeline of algorithms are evaluated.
Finally, in Section 4, we summarized the results of the present work and we suggested some possible directions for further research in this field.

1.2. State-of-the-art

We will now present a brief survey of the state-of-the-art techniques used for estimating the pose of a spacecraft from a monocular image.
Typically, all these techniques make use of an image-processing subsystem that identifies the position in the image frame of certain semantic features of the S/C. This is followed by a pose solver consisting in a geometric optimization subsystem, that fits a known 3D model of the target S/C to the features matched in the image.
The aforementioned routine shall then be embedded in a navigation filter, to be used in an actual rendezvous scenario, during which the inter-spacecraft distance ranges from tens of meters to a few centimeters.
Depending on the approach adopted for image processing, two main classes of monocular pose estimation methods may be identified.

1.2.1. Feature-based pose estimation

Feature-based methods seek for correspondences between edges detected in the image and line segments of the known wireframe model of the spacecraft. In 2014, D'Amico proposed a monocular vision-based navigation system that, for the first time, enabled proximity navigation with respect to a completely uncooperative space object [5]. Indeed, unlike previous work, neither supportive means (e.g. light-emitting markers) nor a priori knowledge of the target's pose are required. The method has been successfully tested on actual flight imagery captured during the PRISMA mission [2]. However, two fundamental limitations are highlighted in [5]: the excessive computational cost, which prevents real-time usage on spaceborne hardware, and the lack of robustness to changes in lighting and background conditions.
In 2018, Sharma et al. proposed their Sharma-Ventura-D'Amico (SVD) feature-based method [19]. The method has been tested on actual flight imagery from the PRISMA mission and, compared to previous work, it proved enhanced computational efficiency and superior robustness to changes in background. The latter is achieved thanks to the fusion of state-of-the-art edge detectors with the Weak Gradient Elimination (WGE) technique, which eliminates gradients where they are weak and highlights those regions where gradients are strong.

1.2.2. Deep learning-based pose estimation

Deep learning-based approaches, instead, make use of a Convolutional Neural Network (CNN) pipeline whose job, depending on the approach, may either consist in:

- regressing the position in the image frame of predefined keypoints, that later become the input of a pose solver [3, 11]
- directly estimating the pose, according to one of the following formulations of the pose estimation problem:
  - regression problem [12]
  - classification problem, which requires a sufficiently dense discretization of the pose space [16, 17]
  - hybrid classification-regression problem [18]

1.3. Spacecraft Pose Estimation Dataset

The Spacecraft PosE Estimation Dataset (SPEED) consists of 15300 grayscale images of the Tango spacecraft, along with the corresponding pose labels. 15000 of these images have been generated synthetically, while the remaining 300 are actual images of a 1:1 mock-up, captured under high-fidelity illumination conditions at the TRON facility. SPEED is the first and only publicly available

Machine Learning dataset for spacecraft pose estimation and has been released in February 2019, with the start of the Pose Estimation Challenge[2] organized by SLAB in collaboration with ESA (Feb-Jul 2019). The camera model used for rendering the synthetic images is that of the actual camera employed for capturing the 300 images of the mock-up. The related parameters are reported in Table 1.

Table 1: SPEED camera model

Parameter	Value
Resolution ($N_u \times N_v$)	1920×1200 px
Focal length f	17.6 mm
Pixel pitch ($\rho_u \equiv \rho_v$)	5.86 μm/px
Horizontal FoV	35.452°
Vertical FoV	22.595°

1.3.1. Synthetic images

All the photo-realistic renderings of Tango are generated using an OpenGL-based pipeline. In half of these 15k images, random Earth images are inserted in the background of the satellite. The Earth backgrounds are obtained by cropping 72 real images captured evenly spaced over 12 hours by the geostationary weather satellite Himawari-8. In all images with Earth background, the illumination conditions used for rendering Tango are consistent with those in the image of the Earth disk.
Besides, Gaussian blurring ($\sigma = 1$) and Gaussian white noise ($\sigma^2 = 0.0022$) are eventually superimposed to all images.
The relative position vector for each generated image is obtained by separately sampling the total distance and the bearing angles:

- total distance $\sim \mathcal{N}(\mu = 3, \sigma = 10)$ m (any value either < 3 m or > 50 m is rejected)
- bearing angles $\sim \mathcal{N}(\mu = [u_0, v_0], \sigma = [5u_0, 5v_0])$ px where $u_0 = \frac{N_u}{2}$, $v_0 = \frac{N_v}{2}$ denote the camera principal point

1.3.2. Actual mock-up images

Given the physical constraints of the TRON facility, the distance distribution of real images is very limited compared to synthetic ones and ranges between 2.8 m and 4.7 m. In addition, unlike the synthetic image source (for which pose labels are automatically annotated), the accurate determination of "ground truth" relative poses of the mock-up requires a complex calibrated motion capture system. The facility includes 10 Vicon Vero v1.3x cameras that track several infrared reflective markers placed onto Tango's body and in the robotic arm that holds the camera (the one that collects the 300 images in the dataset). High accuracy light sources are present to mimic sunlight and Earth's albedo.

[2]https://kelvins.esa.int/satellite-pose-estimation-challenge/ (accessed on March 13th 2021)

1.3.3. Dataset re-partitioning

As of November 2020, the Pose Estimation Challenge is still running in post-mortem mode, with a separate leaderboard for all the results submitted after July 2019. To maintain the integrity of the post-mortem competition, the ground truth labels of the test set have not been publicly disclosed.

Given the purposes of this work, which include a detailed evaluation of both performance and uncertainty of our pose estimation pipeline, it was clearly of paramount importance to be provided with test labels. It was therefore decided to perform a re-partitioning of the original training set (for which pose labels are publicly available) into three new training, validation, and test sets. In particular, the original 12k training examples were first of all randomly shuffled and then divided into:

- 7680 training images (64%)
- 1920 validation images (16%)
- 4800 test images (24%)

1.4. SLAB/ESA challenge

The SLAB/ESA Pose Estimation Challenge is based on the evaluation of a single scalar error metric. For convenience, we will refer to it as the "SLAB score". Although this metric is separately computed for both the real and synthetic datasets, participants are exclusively ranked based on the performance on synthetic images.

The SLAB score of each image is determined as the sum of a translation error and a rotation error, as defined in Equation (1). The translation error is computed as the norm of the difference between the Ground Truth (GT) relative distance vector $\mathbf{r}$ and the estimated one $\hat{\mathbf{r}}$, normalized with respect to the GT distance. The rotation error is defined as the quaternion error between the GT relative attitude and the corresponding estimate.

$$e_{\text{SLAB}}^{(\text{i})} = \underbrace{\frac{\left\|\mathbf{r}^{(\text{i})} - \hat{\mathbf{r}}^{(\text{i})}\right\|}{\left\|\mathbf{r}^{(\text{i})}\right\|}}_{e_t^{(\text{i})}} + \underbrace{2 \cdot \arccos\left|\mathbf{q}^{(\text{i})} \cdot \hat{\mathbf{q}}^{(\text{i})}\right|}_{E_q^{(\text{i})}} \quad (1)$$

The overall score is then just the average over all the N test images.

$$e_{\text{SLAB}} = \frac{1}{N}\sum_{i=1}^{N} e^{(i)}_{\text{SLAB}} \tag{2}$$

The outcome of the competition is described in [8] and summarized in Table 2.

Table 2: Leaderboard of the top 10 teams

Team name	Synthetic images score	Real images score	Translation error [m] ($\mu \pm \sigma$)	Quaternion error [deg] ($\mu \pm \sigma$)
1. UniAdelaide	**0.0094**	0.3752	0.032 ± 0.095	0.41 ± 1.50
2. EPFL_cvlab	0.0215	**0.1140**	0.073 ± 0.587	0.91 ± 1.29
3. pedro_fairspace	0.0571	0.1555	0.145 ± 0.239	2.49 ± 3.02
4. stanford_slab	0.0626	0.3951	0.209 ± 1.133	2.62 ± 2.90
5. Team_Platypus	0.0703	1.7201	0.221 ± 0.530	3.11 ± 4.31
6. motokimura1	0.0758	0.6011	0.259 ± 0.598	3.28 ± 3.56
7. Magpies	0.1393	1.2659	0.314 ± 0.568	6.25 ± 13.21
8. GabrielA	0.2423	2.6209	0.318 ± 0.323	12.03 ± 12.87
9. stainsby	0.3711	5.0004	0.714 ± 1.012	17.75 ± 22.01
10. VSI_Feeney	0.4658	1.5993	0.734 ± 1.273	23.42 ± 33.57

2. RELATIVE POSE ESTIMATION PIPELINE

In this section we will present the architecture of the Relative Pose Estimation Pipeline (RPEP) proposed in this paper. Our algorithms require the knowledge of camera intrinsics and of the 3D model of the target spacecraft to rendezvous with. Based on this, the architecture that has been developed is capable of estimating the pose of the target spacecraft, from a single monocular grayscale image given as input.

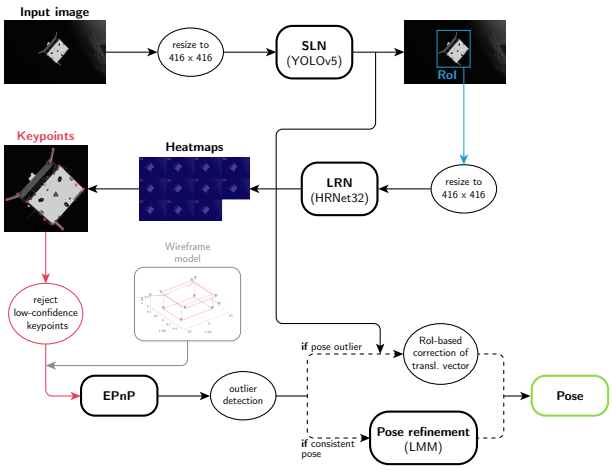

Figure 1: Architecture of the pose estimation pipeline at inference time

The outline of our architecture is represented in Figure 1 and it consists of three main subsystems.

The first subsystem, called the Spacecraft Localization Network (SLN) and described in Section 2.1, is responsible for identifying the Region of Interest (RoI) in the image.

This is followed in the pipeline by the Landmark Regression Network (LRN), that we detailed in Section 2.2, whose role is to detect semantic keypoints of the target S/C in the RoI.

The third and last subsystem is the pose solver, which, given the landmarks identified by LRN, seeks for the corresponding best pose fit based on the known wireframe model of the target. It will first run the Efficient Perspective-n-Point (EPnP) algorithm [9] to obtain an initial estimate of the pose and, in a nominal situation (i.e. if no pose outlier is detected), it will successively refine the initial solution using the Levenberg-Marquardt Method.

2.1. Spacecraft Localization Network

The Spacecraft Localization Network (SLN) is the first image processing subsystem of the Relative Pose Estimation Pipeline (RPEP) proposed in this paper.
The You Only Look Once (YOLO) architecture [13], which is a state-of-the-art one-stage method for object detection, has been chosen for this purpose. In particular, the most recent iteration of the CNN, `YOLOv5` [21], was trained to detect the Tango satellite.
The SLN receives as input a grayscale image, that is properly resized to match the input size of 416×416 of our `YOLOv5` architecture. This subsystem outputs the so called Region of Interest (RoI), namely the Bounding Box (BB) coordinates associated with the portion of the image containing the S/C. Based on this, further processing of the image will exclusively focus on the identified RoI.

First of all, a one-stage detection approach was chosen over region proposal networks (e.g. [15]) given the clear superiority in terms of computational efficiency of the former class of methods. This is of paramount importance in a spaceborne navigation scenario, where the computing power constraints always make it necessary to opt for efficient yet robust algorithms. In this sense, the smallest model-size version of `YOLOv5`, named `YOLOv5s`, proved particularly interesting for our purposes and has been eventually selected. With 7.5M parameters to train and 191 layers, it appears as an excellent trade-off between speed and accuracy.

2.1.1. Training

The SPEED training labels released by SLAB only include the pose of the Tango spacecraft, provided in terms of translation vector and attitude quaternion for each image. This means that any further label that might be used for intermediate processing steps will have to be annotated.
The minimum rectangle enclosing the S/C in the image frame can be obtained by projecting onto the image plane a simple wireframe model of Tango, based on the known pose. We may at this point annotate the BB of each image by taking the minimum and maximum values of the (P_x, P_y) coordinates of the small amount of points in this simplified 3D model.

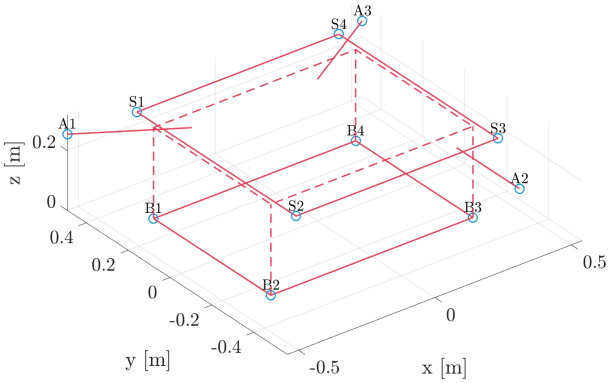

Figure 2: Wireframe model of the Tango spacecraft

The wireframe model used in our work is depicted in Figure 2. In particular, the model is composed of 11 semantic keypoints:

- points B1 to B4 are the edges of the bottom surface
- points S1 to S4 are the edges of the solar panel
- points A1 to A3 indicate the tips of the Formation Flying Radio Frequency (FFRF) antennas

These very same keypoints are also used by the successive subsystem, the Landmark Regression Network (LRN), which we will describe in subsection 2.2. The reason behind this choice is that these landmarks represent strong visual features of the spacecraft, and, independently of the pose, most of them will not be occluded by other surfaces.

In order to avoid unintentionally cropping out of portions of the S/C from the detected RoI during inference, we slightly relax the minimum rectangle enclosing the projected wireframe model. Specifically, the BB labels are enlarged by the 10% of the average side between width and height of the minimum rectangle. In so doing, the CNN is indeed trained to predict a relaxed bounding box. In Figure 3, the dashed yellow line indicates the minimum rectangle, while the continuous line is the actual BB label.

The network was trained for 125 epochs using Stochastic Gradient Descent (SGD), with a mini-batch size of 48 images, learning rate $\alpha = 10^{-3}$, momentum equal to 0.9 and a weight decay of 5×10^{-5}. The binary cross-entropy loss was used during training.
In addition, given the assumption of single-class/single-object in the image, a few simplifications in the algorithm were introduced compared to the generic multiple-class/multiple-object framework, for which `YOLO` has been developed. In so doing, we are able to get rid of some unnecessary computation, also making sure that the algorithm outputs one single RoI, provided that the prediction confidence is at least 60%. In other words, we can directly output the prediction with the highest "objectness" score, with no need to process the raw results using the non-max suppression algorithm.

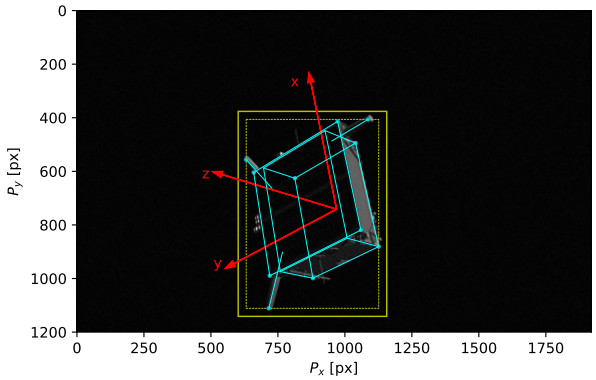

Figure 3: Bounding box label of the `img001971.jpg` training image

2.1.2. Performance evaluation

The performance of SLN has been evaluated using the Average Precision (AP) and Intersection over Union (IoU) metrics [4]. The AP is defined as the area under the precision-recall curve, that is $\int_0^1 P(R)\,\mathrm{d}R$. The 10 precision-recall curves in correspondence of the IoU thresholds 0.5, 0.55, 0.6, 0.65, ..., 0.95 are reported in Figure 4.[3] These thresholds are used for defining a correct detection (i.e. a True Positive).

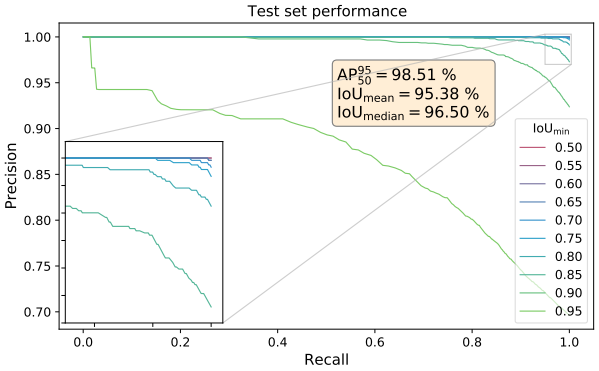

Figure 4: Precision-recall curves, in correspondence of different IoU thresholds

The AP_{50}^{95} metric is then simply the average of the AP values computed for each of the 10 curves in Figure 4. The `YOLOv5` architecture achieves an excellent accuracy, with $AP_{50}^{95} = 98.51\%$, after only 125 training epochs.
Indeed, by comparing the mean and median IoU metrics in Table 3, our spacecraft localization subsystem outperformed at this task both the SLAB baseline and the architecture proposed by the `UniAdelaide` team, which respectively ranked 4th and 1st in the Pose Estimation Challenge.

[3] the curves here provided are specifically computed considering the interpolated precision

Table 3: Performance comparison of SLN with other state of the art RoI detection subsystems

	stanford_slab [11]	UniAdelaide [3]	Our SLN
Mean IoU	91.9%	95.34%	**95.38%**
Median IoU	93.6%	96.34%	**96.50%**

2.2. Landmark Regression Network

The Landmark Regression Network (LRN), which is the second image processing subsystem in our pipeline, receives as input the grayscale RoI detected by SLN. The input size of LRN is again 416×416, which means that RoIs whose largest side is greater than 416 pixels will undergo downscaling. If on the contrary the RoI gets smaller than LRN's input size, then the portion of the image that borders the BB is used to fill the rest of the input window. This is indeed useful in the event of an inaccurate detection where a portion of the S/C would be cropped out.
The unprecedented accuracy demonstrated by the High-Resolution Network (HRNet) architecture [20] in the field of human pose estimation led to the decision of implementing this model in our architecture.
This CNN has been trained to regress 11 heatmaps with a size of 416×416, corresponding to the 11 semantic keypoints specified in Figure 2. The final predicted landmark locations are then obtained as the individual peaks in each heatmap, which will appear as 2D pseudo-Gaussians.

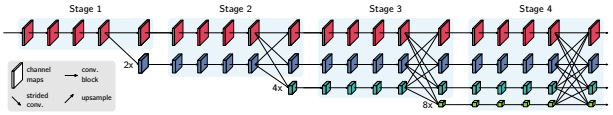

Figure 5: Main body of the HRNet architecture

The strength of HRNet lies in two main distinctive aspects.

- Most previous architectures recover high resolution representations by performing an upsampling process downstream of a high-to-low resolution network. In contrast, HRNet maintains the initial high-resolution representation throughout the entire network. This clearly eliminates the loss of information associated with traditional approaches, resulting in more accurate heatmaps, which is of paramount importance in a spaceborne relative navigation scenario.
  In particular, the network starts with a high-resolution subnetwork whose resolution is kept unaltered up to the last layer. As it is depicted in Figure 5, lower-resolution subnetworks are gradually stacked in parallel as we go deeper in the network.
- Instead of aggregating high- and low-resolution representations, HRNet performs repeated multi-scale fusions to boost the low-level representations with the aid of high-level representations, and vice-versa.

In [20] two different versions of the HRNet model are presented, which were named HRNet32 and HRNet48. The numbers 32 and 48 indicate versions of the network having respectively 32 and 48 channels in the highest-resolution subnetworks in the last three stages.
It was decided to implement the HRNet32 version, given a performance level quite close to the larger version of the network. The latter appears slightly superior, but this comes at the expense of more than twice the number of FLoating-Point Operations compared to the smaller model [20].

2.2.1. Training

The Ground Truth labels have been annotated by projecting onto the image frame the 11 keypoints defined in the 3D wireframe model of Tango, based on the known training poses. The corresponding GT training heatmaps are then set to 2D Gaussians with 1-pixel standard deviation and mean value in correspondence of the projected landmark coordinates.

Despite the use of high-end GPUs on the Google Colab platform, the training of this architecture turned out to be very expensive and has only been carried out for 80 epochs. ADAptive Momentum (ADAM) optimization [7] has been used, with a batch-size of 16 images, $\beta_1 = 0.9$, $\beta_2 = 0.99$, learning rate equal to 10^{-3} and a weight decay of 10^{-4}.
The loss function for the ith image is defined as the mean squared error between the regressed heatmap $\hat{\mathbf{H}}$ and the corresponding Ground Truth $\mathbf{H}$, averaged over all the n landmarks lying inside the image frame:

$$\mathbf{L}_{\text{MSE}}^{(\text{i})} = \frac{1}{n} \sum_{j=1}^{n} v_j^{(\text{i})} \cdot \left[\hat{\mathbf{H}}_j^{(\text{i})} - \mathbf{H}(\mathbf{p}_j^{(\text{i})}) \right]^2 \qquad (3)$$

The loss computed for an entire mini-batch is simply the average over all images in the batch, namely $\mathbf{L}_{\text{MSE}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{L}_{\text{MSE}}^{(i)}$.

2.2.2. Performance evaluation

The performance of our LRN has been evaluated in terms of AP and Object Keypoint Similarity (OKS).[4] The 10 precision-recall curves in Figure 6 are computed in correspondence of the 10 equally spaced OKS thresholds, from 0.5 to 0.95. The Average Precision is then calculated for each of these curves, from which we eventually obtain the global metric $\text{AP}_{50}^{95} = 98.97\%$. This indeed indicates an excellent regression accuracy, obtained after only 80 training epochs.

[4] similarly to the IoU in an object detection framework, the OKS indicates the average degree of overlap between detected keypoints and their actual location

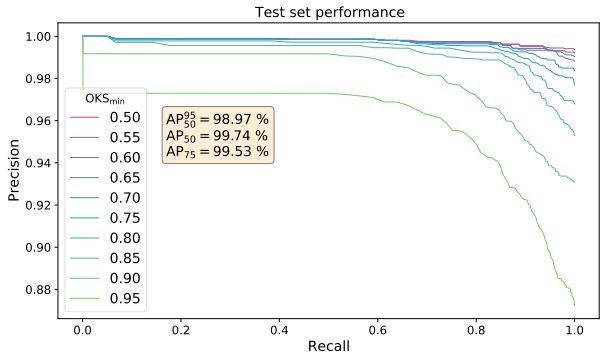

Figure 6: Precision-recall curves, in correspondence of different OKS thresholds

2.3. Pose solver

The pose solver is the third and last subsystem of our Relative Pose Estimation Pipeline, that identifies the best pose fit, based on the keypoints detected by LRN. The pose solver also leverages Bounding Box information (i.e. the output of SLN) to identify the presence of outliers among the considered keypoints, and partially correct the resulting wrong pose estimate.

2.3.1. Keypoint selection

The availability of a heatmap that provides a confidence score for a given detected landmark can be leveraged to filter out potential outliers, which may cause a pose solver to diverge or to output a completely wrong pose. In particular, two hyper-parameters have been tuned, in order to find a good compromise between rejecting potential outliers and retaining a sufficient number of points. Regarding this last goal, it is clearly beneficial in terms of accuracy to over-constrain the 3D model, as long as we keep adding precise keypoint detections. The hyper-parameters that have been consequently selected are:

- # landmarks$_{\text{min}}$: size of the minimal set of landmarks, i.e. the minimum number of the highest-confidence detected landmarks to be always retained, independently of their associated scores
- confidence$_{\text{min}}$: minimum confidence required to retain any landmark in addition to the minimal set

This means that, in general, only a subset of the 11 keypoints will be effectively fed to the pose solver.
The two above mentioned hyper-parameters have been optimally tuned by minimizing the MNPE defined in Equation (19). All the results presented in the remainder of this discussion have been obtained in correspondence of the optimal values: confidence$_{\text{min}} = 0.8$ and #landmarks$_{\text{min}} = 7$.

2.3.2. Initial pose estimation and refinement

After discarding low-confidence landmarks, the remaining ones are fed to the EPnP algorithm [9], which computes a first pose estimate and does not require any initial guess. This method consists in a closed-form solution to the Perspective-n-Point (PnP) problem, having complexity of order $\mathcal{O}(n)$. EPnP is characterized by a weak robustness to the presence of outliers among the input keypoints. However, if no outliers are present, the resulting pose estimate turns out to be quite accurate.
At this point, our algorithm checks whether or not the estimated pose is consistent with the BB detected by SLN. Indeed, it was concluded that, after proper training, we can "trust" SLN more than LRN, just because the former actually performs a simpler task. Thus, whenever an inconsistency is found between the two subsystems, it is reasonable to believe that LRN is to blame. In other words, whenever the projection of Tango's 3D model (based on the initial pose estimate) is inconsistent with the detected BB, this is very likely due to the presence of one or more outliers among the retained landmarks, which translates into a completely wrong pose computed by EPnP.

If no inconsistency is found in the output of EPnP and the reprojection error is acceptable, this initial pose is refined using the Levenberg-Marquardt Method, that iteratively minimizes the reprojection error.

2.3.3. Outlier identification & translation correction

If, on the contrary to what previously described, a pose outlier is flagged by our algorithm, we will then partially correct the pose by replacing the translation vector output by EPnP with an approximate yet robust estimation.

In order to identify a possible pose outlier, an approximate translation vector $\tilde{\mathbf{t}}_{\text{C/B}}$ is first of all computed, by exploiting a RoI-based estimation. This method leverages the knowledge of the characteristic length L_C of the spacecraft, along with the BB's center $(P_x^{\text{BB}}, P_y^{\text{BB}})$ and diagonal length d_{BB} that are detected by SLN. The aforementioned dimensions and coordinates are indicated in Figure 7.

Given our camera intrinsics (Table 1), we are able to relate the size of our real-world S/C to the corresponding size in the image frame, hence obtaining the following expression for the distance between the camera-fixed and the body-fixed frames

$$\tilde{t}_{\text{C/B}} = \frac{f/\rho_u + f/\rho_v}{2} \cdot \frac{L_C}{d_{\text{BB}}} \tag{4}$$

We may similarly compute also the azimuth and elevation

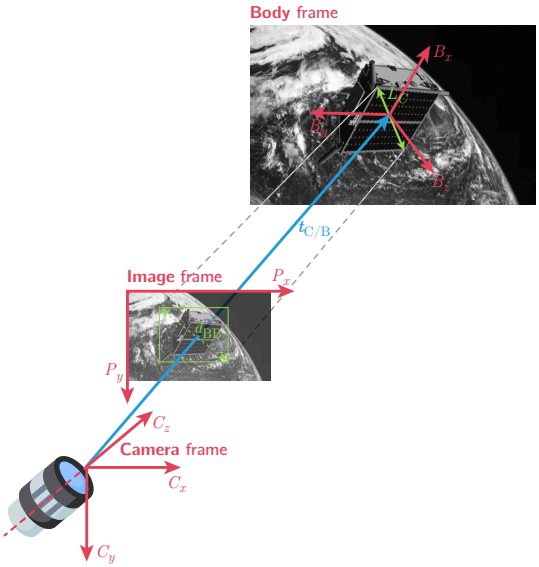

Figure 7: Reference frames and RoI

angles, α and β, as

$$\alpha = \arctan\left(\frac{P_x^{\text{BB}} - u_0}{f/\rho_u}\right) \tag{5}$$

$$\beta = \arctan\left(\frac{P_y^{\text{BB}} - v_0}{f/\rho_v}\right) \tag{6}$$

At this point, a coarse estimate of the camera-to-body translation vector may be derived as

$$\tilde{\mathbf{t}}_{\text{C/B}} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ \tilde{t}_{\text{C/B}} \end{Bmatrix} \tag{7}$$

An outlier will be flagged whenever any of the following conditions is encountered.

- The projected geometric center of the S/C, according to the pose estimated by EPnP, has a $> 50\%$ offset[5] from the BB center

$$\frac{|\hat{p}_x^c - P_x^{\text{BB}}|}{w_{\text{BB}}} > 0.5 \quad \text{or} \quad \frac{|\hat{p}_y^c - P_y^{\text{BB}}|}{h_{\text{BB}}} > 0.5 \tag{8}$$

- Large mismatch between the distance estimated by EPnP, $\hat{t}$, and the one obtained from the RoI-based approximation, $\tilde{t}$

$$\left|\frac{\hat{t} - \tilde{t}}{\tilde{t}}\right| > 75\% \tag{9}$$

[5]the pixel offset is normalized with respect to the BB width and height

- Medium distance mismatch and low average confidence of the retained landmarks

$$\left|\frac{\hat{t} - \tilde{t}}{\tilde{t}}\right| > 15\% \quad \text{and} \quad \text{confidence}_{\text{avg}} < 50\% \tag{10}$$

- Medium distance mismatch and high relative reprojection error

$$\left|\frac{\hat{t} - \tilde{t}}{\tilde{t}}\right| > 15\% \quad \text{and} \quad \frac{E_{\text{repr}}}{d_{\text{BB}}} > 10\% \tag{11}$$

Whenever a pose outlier is flagged by our algorithm, the initial estimate of the translation vector will be replaced by the corresponding RoI-based approximation $\tilde{\mathbf{t}}_{\text{C/B}}$.

3. RESULTS

3.1. Error metrics

Prior to presenting the results achieved by our Relative Pose Estimation Pipeline, we will dwell on the definition of the error metrics that allow us to evaluate the performance of our architecture on the Spacecraft PosE Estimation Dataset (SPEED), in terms of mean and median pose error.
In our case, median error is actually more representative of the accuracy as compared to mean: the reason is that we experienced the presence of very few outliers, which are nevertheless characterized by an error that is orders of magnitude larger than nominal detections.

3.1.1. Translation error

The absolute translation error for a given image is obtained as

$$E_t = \left\|\hat{\mathbf{t}}_{\text{C/B}} - \mathbf{t}_{\text{C/B}}\right\| \tag{12}$$

which can be easily normalized if we divide it by the GT distance:

$$e_t = \frac{E_t}{\|\mathbf{t}_{\text{C/B}}\|} \tag{13}$$

3.1.2. Rotation error

Absolute error The absolute rotation error might be measured in two different fashions.

In terms of quaternion error, which represents the overall attitude error with a single scalar metric, it will be computed as

$$E_{\text{q}} = 2 \cdot \arccos|\mathbf{q} \cdot \hat{\mathbf{q}}| \tag{14}$$

In terms of Euler angles, the error will be obtained as the difference between a given estimated Euler angle and the corresponding GT

$$E_{\theta_j} = |\hat{\theta}_j - \theta_j|, \tag{15}$$

Normalized error The main weakness of the SLAB score defined in Equation (1) is that, although it accounts for distance-normalization in its translation component, it does not account for normalization of the rotation error component. This means that the same absolute angular error has the same exact effect upon measured performance, independently of whether that occurs in correspondence of a close-range image or at a distance in which the RoI is just a very small fraction of the entire image area.

This led us to introduce a normalized version of the quaternion error defined in Equation (14), which accounts for the angular size of the object relative to the FoV of the camera. In particular, an object's angular size is defined as the angle measured between the two lines of sight corresponding to opposite sides of the object. In our case, we will consider the angle associated with the diagonal size of each GT Bounding Box.

If we resort to the pinhole camera model and assume that the lens is set for infinity focus, the diagonal angular size associated with the spacecraft can be computed as

$$\alpha = 2 \cdot \arctan \frac{\rho \cdot d_{\text{BB}}}{2f} \tag{16}$$

In Equation (16), $\rho \equiv \rho_u \equiv \rho_v$ is the pixel pitch [μm/px], d_{BB} is the diagonal length of the BB [px], while f is the focal length [mm].

Note that, in order to normalize the rotation error, we need to divide it by a quantity that increases as the attitude gets harder to estimate. We will therefore divide the quaternion error defined in Equation (14) by the portion of the diagonal FoV of the camera that is not occupied by the spacecraft, which reads

$$e_q = \frac{E_q}{\text{FoV}_{\text{diag}} - \alpha} \tag{17}$$

where, considering an $N_u \times N_v$ image, the diagonal FoV can be obtained as

$$\text{FoV}_{\text{diag}} = 2 \cdot \arctan \frac{\rho \cdot \sqrt{N_u^2 + N_v^2}}{2f} \tag{18}$$

3.1.3. Pose error

The overall pose error is simply measured as the sum of the translation and rotation errors.

The SLAB score, which has already been defined in Equation (1), measures the total error as the mean of $\left(e_t^{(\text{i})} + E_q^{(\text{i})}\right)$ computed over all the N test images.

After having highlighted the weaknesses the aforementioned metric, we are hereby proposing an alternative performance index that we deem to be more relevant. It has been called the Median Normalized Pose Error (MNPE):

$$e_{\text{MNP}} = \underset{i=1}{\overset{N}{\texttt{median}}}\left(e_t^{(\text{i})} + e_q^{(\text{i})}\right) \tag{19}$$

where e_t and e_q are defined in Equations (13) and (17), respectively.

3.2. Performance evaluation

Our Relative Pose Estimation Pipeline, achieved a SLAB score of 0.04627 on our test set. This means that, based on the official leaderboard of the SLAB/ESA Pose Estimation Challenge reported in Table 2, our architecture would virtually score 3rd place, hence outperforming the SLAB baseline.
Indeed, this performance level has been confirmed by participating in the post-mortem competition, which is still running on the ESA website. Figure 8 has been printed from the website of the post-mortem competition[6] and reports the score achieved by the 5 top teams, as of March 13th 2021.

Timeline

The competition is in progress.

Leaderboard

Name	Submissions	Last Submission	Best Submission	Real Image Score	Best Score
	competition winner UniAdelaide			0.36340645622528017	0.00864899489025079
arunkumar04	5	June 11, 2020, 2:09 p.m.	June 11, 2020, 3:22 a.m.	0.2897316198709755	0.009653543468537 69
wangzi_nudt	27	Feb. 4, 2021, 7:03 a.m.	Feb. 3, 2021, 2:35 a.m.	0.16838921336519672	0.012316958900754 66
UT-TSL	1	July 29, 2020, 8:46 p.m.	July 29, 2020, 8:46 p.m.	0.29182320619186036	0.040888808313561543
massimo.piazza	5	Dec. 2, 2020, 3:44 p.m.	Dec. 2, 2020, 3:44 p.m.	0.12025061876822 63	0.045002069996446 09
haoranhuang	85	March 13, 2021, 12:13 p.m.	Jan. 20, 2021, 3 p.m.	0.2593928561824155	0.051007326955399 24

Figure 8: Top 5 participants of the post-mortem competition

It can be noted that our architecture attained a SLAB score, on the synthetic original test set of SPEED, equal to 0.04500. This corresponds to a performance level that is practically identical to the one estimated on our test set. The score on the synthetic distribution is here labeled as "best score", while the "real image score" indicates the accuracy achieved on the 300 real images of a mockup of the Tango spacecraft.

[6] https://kelvins.esa.int/pose-estimation-challenge-post-mortem/leaderboard/ (accessed on March 13th 2021)

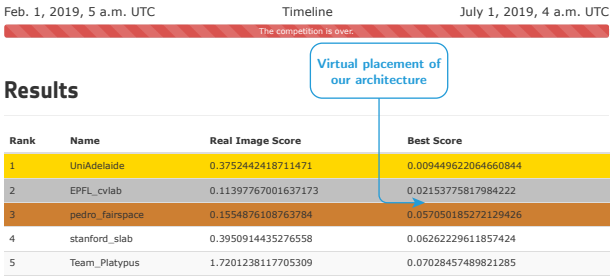

Figure 9: Top 5 participants of the original competition (Feb - Jul 2019)

Figure 9 reports instead the top 5 participants (out of 48 individuals/teams) of the original competition.[7] By comparing the results it can be seen that, in terms of synthetic score, only two of these 48 participants outperformed our architecture. In addition, if we were to merge the leaderboards of the two competitions (66 overall participants), only the `EPFL_cvlab` team would achieve a better score on the real dataset (0.11398 vs. 0.12025).

In Table 4 we reported the most important performance metrics attained by our architecture on our test set.

Table 4: Global end-to-end performance of the RPEP

	Absolute error	
	Mean	**Median**
E_t	10.36 cm	3.58 cm
$\mathbf{E}_t$	[0.52 0.56 10.25] cm	[0.24 0.27 3.50] cm
E_q	2.24°	0.81°
$\mathbf{E}_\theta$	[1.57° 0.84° 1.72°]	[0.52° 0.33° 0.34°]
	SLAB score = 0.04627	MNPE = 0.00648
	Standard deviation of the error	
$\sigma_{\mathbf{E}_t}$	[1.62 1.71 30.44] cm	
$\sigma_{\mathbf{E}_\theta}$	[8.92° 5.11° 10.82°]	
$\sigma_{\mathbf{e}_t}$	[0.001157 0.001093 0.014890]	
$\sigma_{\mathbf{e}_\theta}$	[0.022179 0.012689 0.026854]	

It can be immediately noticed that there is a substantial difference between mean and median error. In particular, the latter is typically $\sim$ 3 times smaller, both in terms of translation and rotation errors. This immediately highlights the presence of pose outliers, which are small in number yet with an error that is orders of magnitude larger compared to the extremely accurate detections that nominally take place.

A total of 13 pose outliers out of 2400 test images has been detected and partially corrected (in terms of translation only). All these images are characterized by a mid-to-large relative distance, a cluttered background with presence of Earth and, as a consequence, very low prediction confidence for all the keypoints detected by LRN. The most common scenario in these cases is the one in which a given landmark is mistaken for another one that is visually similar. This leads to an inconsistency which, nonetheless, the EPnP algorithm tries to fit, thus resulting into a completely wrong pose estimation.
The RoI-based approximation employed for correcting the relative translation vector yields substantial improvements, although the accuracy of the boresight component is highly dependent on the range between chaser and target.

To conclude, in Figure 10 we provided some pose visualization results obtained by running inference on randomly chosen test images with various lighting and background conditions, that are here sorted based on ground truth relative distance.

4. CONCLUSIONS & FUTURE WORK

The main contribution of this work is the development of a deep learning-based pipeline capable of estimating the relative pose of an uncooperative spacecraft from a single monocular image, provided the knowledge of the target's 3D model and with no need of any other a priori information.
We therefore introduced our Relative Pose Estimation Pipeline (RPEP), which is composed of three main subsystems.

1. Spacecraft Localization Network (SLN). Its aim is to identify in the input image the RoI, in which the S/C is located. This allows cropping out irrelevant portions of the image, so as to avoid unnecessary computation. SLN is a Convolutional Neural Network (CNN) based on the `YOLOv5` architecture. For this subsystem alone, the measured Average Precision is $\text{AP}_{50}^{95} = 98.51\%$, with a mean IoU of 95.38%.

2. Landmark Regression Network (LRN). It processes the output of the previous subsystem in order to detect the position in the RoI of pre-defined semantic keypoints of the S/C. This CNN is based on the `HRNet32` architecture. The Average Precision of the landmark regression task, measured in terms of OKS thresholds, is $\text{AP}_{50}^{95} = 98.97\%$.

3. Pose solver. This third and last subsystem receives as input the landmarks detected by LRN and seeks for the best pose fit of the known 3D wireframe model of the satellite, that minimizes the reprojection error. Our algorithm is based on the EPnP method for computing an initial pose estimate, which is iteratively refined using the Levenberg-Marquardt Method (LMM). The pose solver is also in charge of flagging and partially correcting possible pose outliers.

The performance of our pipeline has been tested on the synthetic images from the Spacecraft PosE Estimation

[7] https://kelvins.esa.int/satellite-pose-estimation-challenge/results/ (accessed on March 13th 2021)

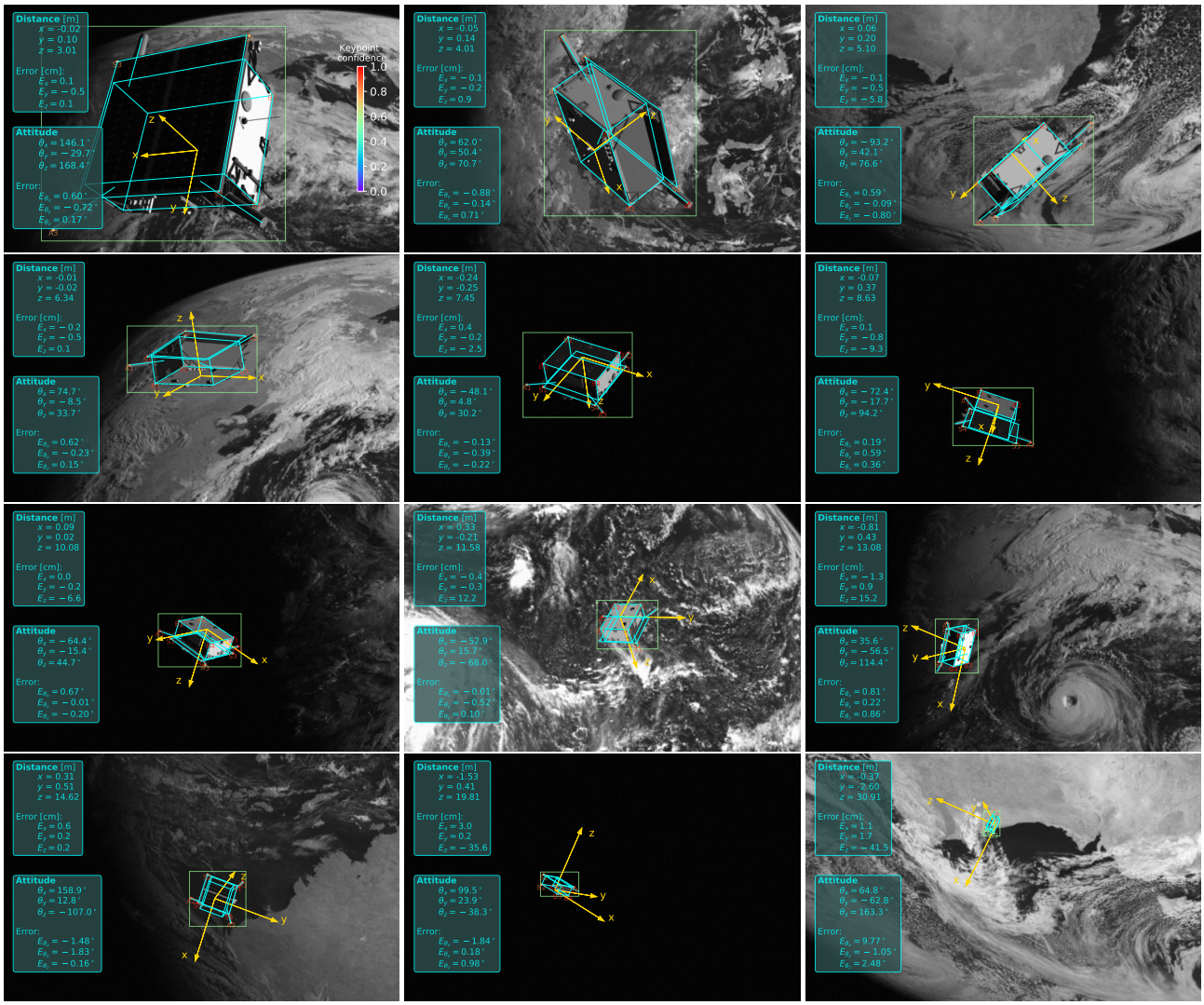

Figure 10: Prediction visualization mosaic of test images with increasing inter-spacecraft distance

Dataset (SPEED). The latter consists of 15300 images of the Tango satellite and is the first and only publicly available Machine Learning dataset for spacecraft pose estimation.

Our architecture demonstrated to outperform the baseline developed by SLAB within the framework of the Pose Estimation Challenge. In particular, a SLAB synthetic score of 0.04500 has been achieved in the post-mortem competition, which means that our RPEP virtually ranks 3rd in original Pose Estimation Challenge. In addition, the same error metric evaluated on the real test set of SPEED corresponds to 0.12025, which, as of March 13th 2021, is the 2nd best score ever obtained since the beginning of the original competition in February 2019.

From the analysis of the results obtained on the test images in SPEED, it was concluded that the accuracy of our estimation strongly correlates with two main factors.

- Inter-spacecraft distance: there will clearly be a progressive drop in performance as the range between chaser and target increases.
- Presence of Earth in the image background: it is intuitive that images with a black background, due to the sharp contrast between the RoI and the rest of the image, will result into features that are easier to detect and hence higher accuracy of the estimated pose.

This means that pose estimation may be particularly challenging in the event of long-range images with cluttered backgrounds, which is indeed the case of our pose outliers. The end-to-end performance of our pipeline, evaluated across the entire test set, corresponds to an absolute translation error of 10.36 cm (mean) and 3.58 cm (median), while the quaternion error is $2.24°$ (mean) and $0.81°$ (median).

We will now provide a few directions for future work, that are necessary steps in the roadmap to spaceborne im-

plementation of a fully vision-based relative navigation system. They are listed here below.

- Performance evaluation in a dynamic rendezvous scenario. The output of the pipeline, which still processes individual frames, is fed to a navigation filter, which accumulates information from sequential images to provide a more accurate dynamic estimate of the pose. A detailed evaluation of the uncertainty in our raw estimates coming from the RPEP is clearly of paramount importance.
- Implementation of an algorithm for identifying individual keypoint outliers, hence removing them from the subset of landmarks processed by the pose solver. One of the main drawbacks of our current architecture, is that whenever a pose outlier is detected, no action is taken in order to correct the attitude.[8]
- Implementation of data augmentation techniques such as the Neural Style Transfer, in order to randomize the S/C's texture in the images used for training our CNNs. This is of particular importance to address the issue of mismatch in terms of textures and reflective properties, between the synthetic imagery used during offline training and the actual flight imagery processed during online inference. Randomizing the textures of our training images would largely improve the robustness to such mismatches [10].
- Inference testing on space-grade hardware or on an off-the-shelf microcomputer such as the Raspberry Pi.

REFERENCES

1. Bamann, C. & Hugentobler, U. *Accurate orbit determination of space debris with laser tracking* in *Proceedings of the 7th European Conf. on Space Debris, edited by Flohrer T. and Schmitz F., ESA Space Debris Office, Darmstadt, Germany* (2017).
2. Bodin, P. *et al.* The prisma formation flying demonstrator: Overview and conclusions from the nominal mission. *Advances in the Astronautical Sciences* **144,** 441–460 (2012).
3. Chen, B., Cao, J., Parra, A. & Chin, T.-J. *Satellite pose estimation with deep landmark regression and nonlinear pose refinement* in *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2019), 0–0.
4. COCO. `https://kelvins.esa.int/satellite-pose-estimation-challenge/`
5. D'Amico, S., Benn, M. & Jørgensen, J. L. Pose estimation of an uncooperative spacecraft from actual space imagery. *International Journal of Space Science and Engineering 5* **2,** 171–189 (2014).
6. ESA. `https://www.esa.int/Safety_Security/Clean_Space/ESA_commissions_world_s_first_space_debris_removal`
7. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
8. Kisantal, M. *et al.* Satellite Pose Estimation Challenge: Dataset, Competition Design and Results. *IEEE Transactions on Aerospace and Electronic Systems* (2020).
9. Lepetit, V., Moreno-Noguer, F. & Fua, P. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision* **81,** 155 (2009).
10. Park, T. H. & D'Amico, S. Generative model for spacecraft image synthesis using limited dataset (2020).
11. Park, T. H., Sharma, S. & D'Amico, S. Towards Robust Learning-Based Pose Estimation of Noncooperative Spacecraft. *arXiv preprint arXiv:1909.00392* (2019).
12. Proença, P. F. & Gao, Y. *Deep learning for spacecraft pose estimation from photorealistic rendering* in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), 6007–6013.
13. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. *You only look once: Unified, real-time object detection* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 779–788.
14. Reed, B. B., Smith, R. C., Naasz, B. J., Pellegrino, J. F. & Bacon, C. E. in *AIAA space 2016* 5478 (2016).
15. Ren, S., He, K., Girshick, R. & Sun, J. *Faster r-cnn: Towards real-time object detection with region proposal networks* in *Advances in neural information processing systems* (2015), 91–99.
16. Sharma, S., Beierle, C. & D'Amico, S. *Towards Pose Determination for Non-Cooperative Spacecraft Using Convolutional Neural Networks* in *Proceedings of the 1st IAA Conference on Space Situational Awareness (ICSSA)* (2017), 1–5.
17. Sharma, S., Beierle, C. & D'Amico, S. *Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks* in *2018 IEEE Aerospace Conference* (2018), 1–12.
18. Sharma, S. & D'Amico, S. Neural Network-Based Pose Estimation for Noncooperative Spacecraft Rendezvous. *IEEE Transactions on Aerospace and Electronic Systems* (2020).
19. Sharma, S., Ventura, J. & D'Amico, S. Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous. *Journal of Spacecraft and Rockets* **55,** 1414–1429 (2018).

[8] in a dynamic scenario, one may actually implement a navigation filter that, whenever a pose outlier is flagged, performs the propagation step but skips the update step, without necessarily having to identify the inconsistent keypoint detection(s)

20. Sun, K., Xiao, B., Liu, D. & Wang, J. *Deep high-resolution representation learning for human pose estimation* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2019), 5693–5703.

21. Ultralytics. *YOLOv5 (*`https://github.com/ultralytics/yolov5`*)*