POLITECNICO
MILANO 1863

RE.PUBLIC@POLIMI
Research Publications at Politecnico di Milano

Post-Print

This is the accepted version of:

M. Pugliatti, V. Franzese, F. Topputo
Data-Driven Image Processing for Onboard Optical Navigation Around a Binary Asteroid
Journal of Spacecraft and Rockets, In press-Published online 04/01/2022
doi:10.2514/1.A35213

The final publication is available at https://doi.org/10.2514/1.A35213

Access to the published version may require subscription.

When citing this work, cite the original published paper.

Permanent link to this version
http://hdl.handle.net/11311/1196071

Data-driven Image Processing for Onboard Optical Navigation Around a Binary Asteroid

M. Pugliatti *, V. Franzese †, and F. Topputo‡
Politecnico di Milano, 20156, Milan, Italy

In this work data-driven image processing options for a CubeSat mission around a binary asteroid system are investigated. The methods considered belongs to two main branches of image processing methods: centroid and artificial intelligence. The former is represented by three variations of centroiding methods, the latter by three neural networks and one convolutional neural network. The first contribution of this work is an enhanced center of brightness method with a data-driven scattering law. This method is demonstrated to share similarities with neural networks in terms both of design and performance, with the advantage of relying on a traditional, robust, and fully explainable algorithm. The second contribution is given by the performance assessment between the different families of image processing methods. For this purpose, the Milani mission is considered as a case study: a 6U CubeSat that will visit the Didymos system as part of the Hera mission. From this analysis, it emerges that convolutional networks perform better than other methods across all metrics considered. This hints to the importance of filtering techniques to extract spatial information from images, which is a unique features of the convolutional approach over the other image processing methods considered.

Nomenclature

a	=	Major axis length of the error ellipse, pxl
a_i, b_i, ci	=	Coefficients of the orientation function
b	=	Minor axis length of the error ellipse, pxl
α	=	Shape parameter of the Gamma probability density function
β	=	Rate parameter of the Gamma probability density function
Γ	=	Bounding box vector
Γ_e	=	Expanded bounding box vector
$\Gamma_{x,y}$	=	Bounding box corner coordinates

*Ph.D. Candidate, Department of Aerospace Science and Technology, Via La Masa 34, mattia.pugliatti@polimi.it
†PostDoc Fellow, Department of Aerospace Science and Technology, Via La Masa 34, vittorio.franzese@polimi.it
‡Full Professor, Department of Aerospace Science and Technology, Via La Masa 34, francesco.topputo@polimi.it, AIAA senior member

$\Gamma_{l,h}$	=	Bounding box length and height
CoB	=	Center of Brigthness, pxl
CoF	=	Center of Figure, pxl
CoM	=	Center of Mass, pxl
$eCoB$	=	Center of Brigthness of the edge region, pxl
ε_{CoF}	=	Error metric for the CoF estimation, pxl
ε_{m}	=	Error metric for the CoF estimation in range on the image plane, m
ε_{PA}	=	Error metric for the phase angle estimation, deg
F	=	Expanding factor of the Bounding Box
M	=	Magnitude correction surface, pxl
M_{coeff}	=	Set of coefficients of the magnitude correction surface
MSE	=	Mean Squared Error
μ	=	Mean of the Gaussian probability density function
PA	=	Phase Angle, deg
p_i	=	Polynomial coefficients of the phase angle function
p_{ij}	=	Polynomial coefficients of the magnitude function
Φ	=	Orientation correction curve, deg
Φ_{coeff}	=	Set of coefficients of the orientation correction curve
X_0	=	Horizontal component of the error ellipse center, pxl
Y_0	=	Vertical component of the error ellipse center, pxl
w	=	Weight parameter of the WCOB method
θ	=	Orientation of the majorx axis of the error ellipse, deg
σ	=	Standard deviation of the Gaussian probability density function
σ_b	=	Binary threshold
σ_d	=	Detection threshold
σ_f	=	Filter threshold for the terminator detection

I. Introduction

In the proximity of a small body, deep-space CubeSats offer the advantage of diversifying and complementing large spacecraft missions [1]. Indeed, they can be exploited as opportunistic payloads to be deployed in situ, once the main spacecraft has reached its target. An example is given by the AIDA (Asteroid Impact and Deflection Assessment) collaboration between NASA and ESA to study and characterize an impact with the binary Didymos asteroid system [2],

composed of a primary called Didymos and a secondary called Dimorphos (for simplicity, also referred to as D1 and D2 in this work, respectively). As part of this collaboration, NASA will launch the DART (Double Asteroid Redirection Test) kinetic impactor spacecraft [3], whose impact with Dimorphos will be observed and characterized by LICIACube in 2022 [4]. As part of this cooperation, ESA will launch the Hera mission [5] together with two deep-space CubeSats, namely Juventas [6] and Milani [7, 8], to study and characterize the system after the impact.

Autonomous Optical Navigation (OpNav) is an enabling technology for present and future exploration missions. It exploits an Image Processing (IP) method to extract measures from optical observables, and then use them to generate a state estimate. Because images can be inexpensively generated onboard with low-cost and low-mass sensors, OpNav is experiencing a growing interest, and several IP methods exist. In the case of proximity to small-bodies, both their features and the full shape are resolved with common imaging sensors. IP methods relying on surface features are usually expensive to process but perform at higher accuracies, while those exploiting asteroid shapes are simpler but have lower performance. In this category there are centroiding techniques, which are simple, robust, and easy to use [9], but they lack accuracy when applied to irregular bodies and when low phase angles are considered. To overcome these issues, scattering laws can be introduced, as illustrated in [10, 11]. Limb and ellipse fitting methods [12–17] are alternatives that work well with regular shapes as those of planets and moons, which is not typical for small bodies. Attempts have been made to extend them also for irregular shapes, most notably in [12]. Correlation methods in the resolved regime can also be used. These either assume a regular shape of the body [18] or need a rough shape model beforehand [19, 20]. The aforementioned methods work under the assumption that the body is fully resolved in the image, however, a different set of methods exist exploiting surface features. In [21, 22] for example, correlations are performed between real images and surface maplets derived from high-fidelity shape models. These have been used in Rosetta [23] and Osiris-Rex [24] missions respectively, while in Hayabusa 2 [25] a close-range navigation strategy based on artificial landmarks positioned on the surface is used [26]. Features can also be used without high-fidelity models available, as for the case of the feature-based navigation of Hera discussed in [18], which is based on the Kanade-Lucas-Tomasi method described in [27].

Recent trends in Artificial Intelligence (AI) and computer vision demonstrated the capability of deep-learning convolutional architectures in image processing tasks [28–30]. Their adoption is becoming even more relevant for space applications [31, 32], in particular related to IP, navigation, and control [33–41]. The main advantage of these techniques is the embedding of a complex task within the weights and bias of a network. This comes at the cost of a large amount of realistic data needed for training.

Within this context, the aim of this work is twofold. First, to design an enhanced version of a centroiding technique representing a compromise between simplicity, robustness and accuracy. This is performed by the use of a data-driven scattering law and an hyper-parameter tuning derived from a methodology commonly used in machine learning. This method, also referred to as WCOB, is a unique contribution of this work and advances previous analytical scattering laws

applied to irregular asteroids and comets [10, 11]. Second, to assess how this and other traditional IP methods perform when compared to AI ones. This type of assessment is not commonly reported in literature, yet it can be fundamental in driving design choices in small-body missions. This is achieved by tuning different IP methods with the same set of images and then test in a close-proximity scenario around a binary asteroid system. This exhibits unique challenges due to the disturbing effect of the secondary body of the system, which is not the main target of the IP methods considered. To evaluate the IP performance, the present work considers images of the Didymos system acquired during the nominal phases of the Milani mission. The comparison between data-driven methods is fundamental to highlight how specific algorithmic design choices are reflected in the performance, which is a unique contribution of this work.

The paper is organized as follows. The tool for generation of images and labels is described in Section II followed up by a detailed description of the five different IP methods used in this work. A short overview of the Milani mission and the application of the different IP methods is presented in Section III. A brief parenthesis upon the explainability of the AI methods is discussed in Section IV, which highlights interesting similarities amongst traditional and AI-based data-driven IP methods. To conclude, the main findings of this work are summarized in Section V, together with some points for future works.

II. Methodology

The image generation tool and the five IP methods used in this work are described in this section. The objective of the IP methods is to extract the Center of Figure (CoF) and Phase Angle (PA) from images, quantities that can be exploited onboard for navigation purposes. This work focuses on the extraction of these quantities from the images. The AI methods are implemented in Tensorflow 2.5.0 using Google CoLab* GPUs for training, while the other methods are implemented in MATLAB R2021a†.

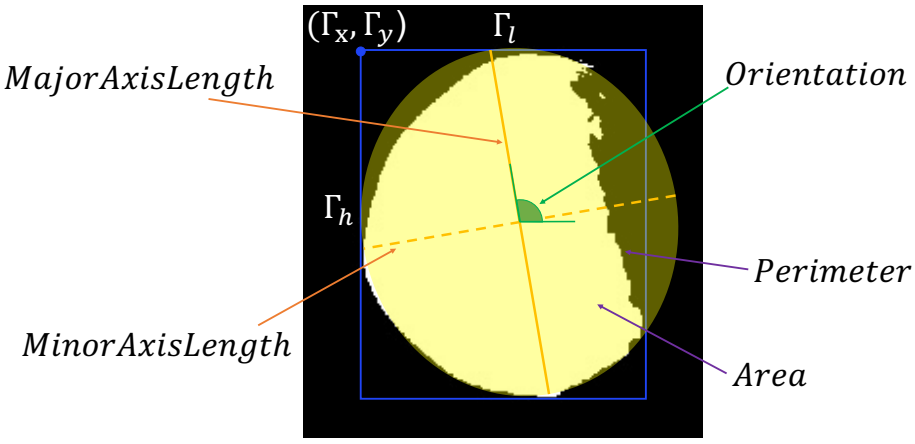


Figure 1 Sketch of the raw optical observables extracted from the binary blob of pixels of Didymos.

*https://colab.research.google.com/, last accessed: 22nd of July, 2021.
†https://it.mathworks.com/products/matlab.html, last accessed: 22nd of July, 2021.

Some conventions and nomenclatures are briefly explained to simplify the discussion. In this work, with CoM is intended the body's center of mass projected on the image while the CoF is the CoM estimated by the IP method. The error between the two represents the loss metric to minimize. In Figure 1 the raw optical observables extracted from the blob of pixels associated to Didymos are represented. The bounding box Γ detected around a blob of pixel is represented by a vector $\Gamma = [\ x\ y\ l\ h\ ]$, the components being respectively the x and y coordinates in image frame of the box corner, and the length and width of the box. The *MajorAxisLength*, *MinorAxisLength* and subsequent *Eccentricity*, are the ones of the ellipse that has the same second central moments as the blob of pixels. The *Orientation* is the angle between the *MajorAxisLength* and the x axis of the image reference frame. The *Perimeter* and *Area* are respectively the length of the boundary of the region and the number of pixels. The *Circularity* is computed as $\frac{4\pi Area}{Perimeter^2}$ while the *EquivalentDiameter* is computed as the diameter of the circle which has the same *Area* of the blob of pixels.

A. SIPTO

SIPTO (Static Image Processing TOol) is a tool developed internally at Politecnico di Milano by the Deep-Space Astrodynamics Research & Technology Group[‡] used for design, validation and testing of OpNav techniques. SIPTO is made by a combination of MATLAB, Spice[§] and Blender[¶] to generate and analyse image databases in a static assessment. The first step is to define a set of datapoints with the desired geometric conditions around a target body. By using Spice, the celestial bodies, spacecraft, and Sun can be queried to match in-flight conditions. The relative geometry among the bodies is given as input to Blender to generate a database. The renderings are collected by specifying the camera characteristics depending on the use case considered. The database (composed by images, labels, and auxiliary data) is then processed by a suite of algorithms for image processing. The output of each algorithm is collected and then a statistical analysis is performed on their performances.

In order to design, train and validate the IP methods considered in this work, a global database of interest is generated. This database, named DB_0, collects a large statistical sample of $10,000$ images of the Didymos binary system seen from different geometric configurations. It comprehends randomly generated points between 4 km and 14 km with azimuth angle between -95 deg and + 95 deg and elevation between -45 deg and +45 deg with respect to a reference frame co-planar with the orbital plane of the secondary. These conditions are representative for a mission designed to actively observe an asteroid from the illuminated side. During the generation of the database, the angular position of the secondary with respect to the primary is changed randomly, constraining the secondary to be tidally locked with the primary. For simplicity, the spacecraft assumes ideal pointing towards the CoM of the primary and images are obtained with the NavCam characteristics but without noise. Moreover, the X axis of the NavCam is aligned with the equatorial plane of the binary system, assuming that the Z axis represents the boresight direction and the X and

[‡]https://dart.polimi.it/, last accessed: 22nd of July, 2021.
[§]https://naif.jpl.nasa.gov/naif/toolkit.html, last accessed: 22nd of July, 2021.
[¶]https://www.blender.org/, last accessed: 22nd of July, 2021.

Y axes are respectively the ones associated with the longest and shortest size of the sensor. The shape models used are enhanced versions of the current ones for the Didymos system. They are processed with procedural changes and re-mesh to simulate roughness and albedo variations with cloud and Voronoi patterns. A 10 m crater is also added on the secondary (which in this work is considered only as a disturbing element) to simulate the DART impact.

B. COB and CHCOB

The COB and Convex Hull COB (CHCOB) are simple, traditional, robust, and well-known methods used to estimate the centroid of a figure by its center of brightness. In the formulation used in this work, the CoB is computed over the binary image using the following equation:

$$CoB_x = \frac{\sum_{i,j=1}^{N} I_{ij} x_{ij}}{\sum_{i,j=1}^{N} I_{ij}} \quad CoB_y = \frac{\sum_{i,j=1}^{N} I_{ij} y_{ij}}{\sum_{i,j=1}^{N} I_{ij}} \tag{1}$$

where I_{ij} is the logic value that determines if the pixel (x_{ij}, y_{ij}) is illuminated or not, whereas CoB_x and CoB_y are the components in pixel of the CoB. The image is pre-processed with thresholding methods to delete background noise. The binary threshold used is determined automatically for each image using the Otsu method [42]. Then, a Region Of Interest (ROI) that surrounds the connected pixels of the asteroid is automatically determined and the centroiding method is applied inside the ROI. The CHCOB is a variant of the COB method in which the convex hull of the object is considered in place of the binary mask. This improves the estimate for irregularly shaped bodies and at high-phase angles since convexified irregular regions on the terminator naturally shifts the CoB towards the CoM. Both methods suffer a bias given by the irregular shape of the body and a large offset when high-phase angles are considered. To overcome the latter, scattering laws can be used [10, 11].

C. WCOB

The Weighted COB (WCOB) is a variant of the COB in which the estimate is corrected by a scattering law derived empirically through data for the irregular body considered. The main goal of the WCOB is to generate a correction vector on the image that pushes the CoB towards the CoM. The steps necessary to generate such correction are now detailed using the nomenclature illustrated in the pipeline sketched in Figure 2.

Once the image is acquired from the NavCam (*step 1*), it is binarized using an arbitrary threshold (σ_b) or Otsu method (*step 2*). The latter is used in this work. An object recognition algorithm is used on the blob of pixels detected in the image to recognize the primary and secondary asteroids (*step 3*). This algorithm works by sorting the blob of pixels larger than a pre-defined threshold (σ_d, set to 100 pixels given the expected dimensions of the Didymos asteroids) based on their areas and by rejecting the ones which are too close to an expanded bounding box (Γ_e) around the biggest blob, which is automatically labelled as the primary. The expansion is managed by the F parameter, which has been set to 1.2 in this work. If a blob pass the rejection test, it is then classified as the secondary, it being the second biggest group of

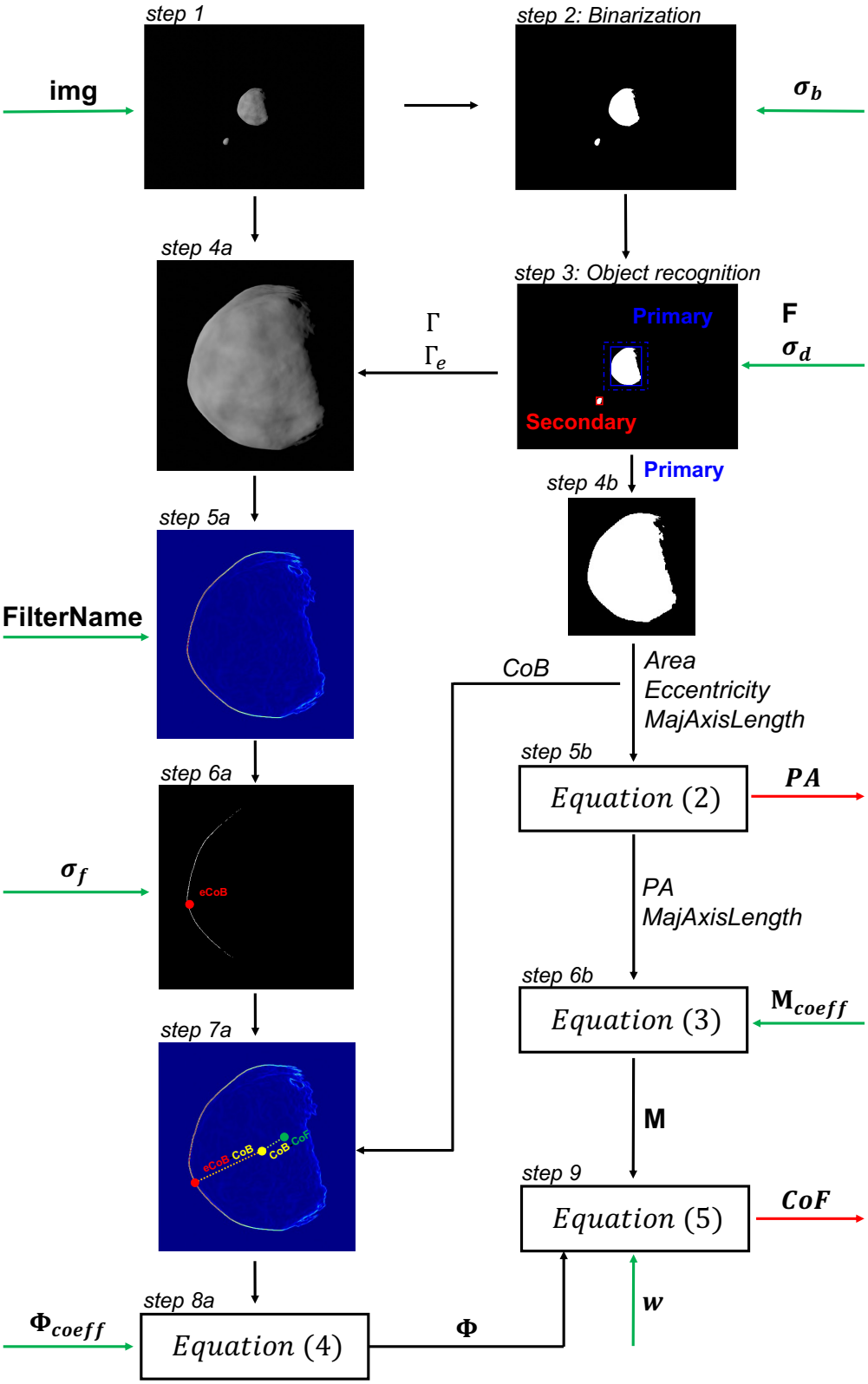


Figure 2 IP pipeline of the WCOB. Input and output are represented respectively by green and red arrows.

pixels sufficiently far away from the primary. With this method, the detection of the secondary in the image achieves a very low number of false-positive (0.11%) but a much higher number of false-negative (8.69%). Once the primary is detected, the focus is put on the portion of the image around it. From this step forward, the IP pipeline of the WCOB is flowing in two parallel branches: one in which the magnitude of the correction is determined (branch *b*) and another one in which the orientation of the correction is determined (branch *a*). The two branches merge together in (*step 9*) to determine the final CoF estimate. Starting from (branch *b*), in (*step 4b*) optical observable of the primary are computed such as the area, centroid, eccentricity, and length of the major axis from the cropped portion of the image. As illustrated in Figure 3 using the datapoints of the DB_0 database, a correlation exists between the eccentricity of the asteroid and the PA. To exploit it, a second-order polynomial is fit to the data in Figure 3 in the least square sense so that the PA can be computed as (*step 5b*):

$$PA(x) = p_2 x^2 + p_1 x + p_0 \tag{2}$$

where p_0, p_1, and p_2 are the polynomial coefficients evaluated from the fitting, while x is the eccentricity of the blob of pixels associated with the asteroid. The polynomial coefficients used are reported in Appendix A.

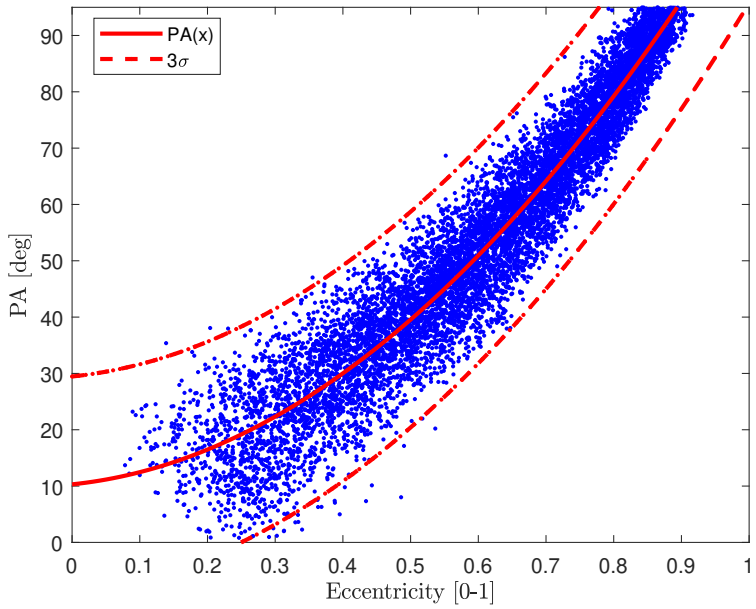


Figure 3 PA vs Eccentricity for all datapoints of DB_0 (blue), of the PA function (solid red), and 3σ value (dashed red).

As illustrated in Figure 3, the PA estimated with Equation 2 would be capable to provide a rough estimate which is more precise at higher PA values. The fit error follows a Gaussian distribution with mean $\mu = 0$ deg and standard deviation $\sigma = 6.383$ deg. A fine estimate of the PA could be performed with the use of data from additional sensors, however, in

this work only imaging data is considered available. Similar to what has been done for the eccentricity-PA relationship, it is observed in Figure 4 that a correlation exists between the PA, MajorAxisLength (which is the length in pixel of the major axis associated to the blob of pixels of the primary), and the distance in pixel between the CoB and CoM for the datapoints of the DB_0 database. This is exploited by the means of a fifth-order polynomial surface which is once again fit using a least squares method:

$$M(x,y) = \sum_{\substack{i=0,\cdots,5 \\ j=0,\cdots,5 \\ i\cdot j \leq 6}} p_{ij} x^i y^j \tag{3}$$

The p_{ij} coefficients used are reported in Appendix A. In (*step 6b*), by using Equation 3, the magnitude correction term of the WCOB method is estimated.

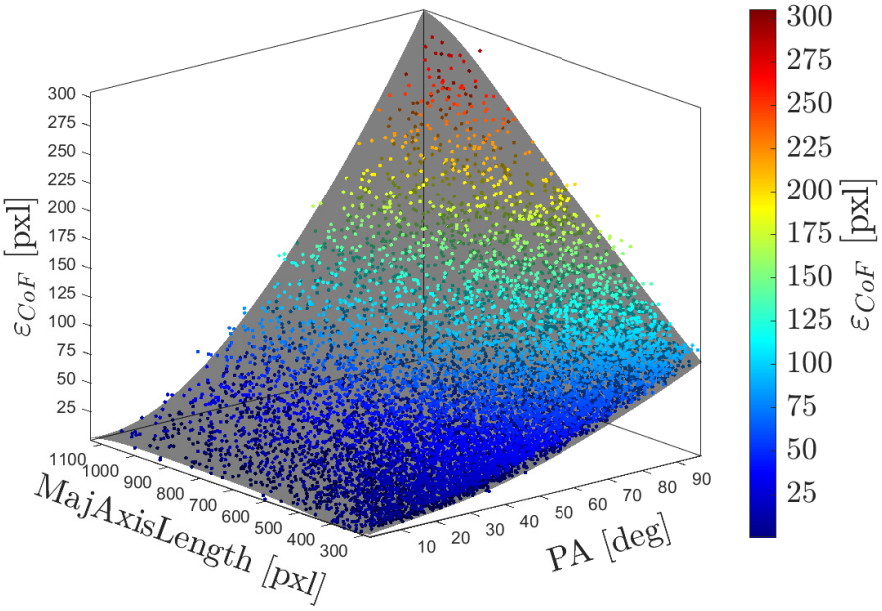
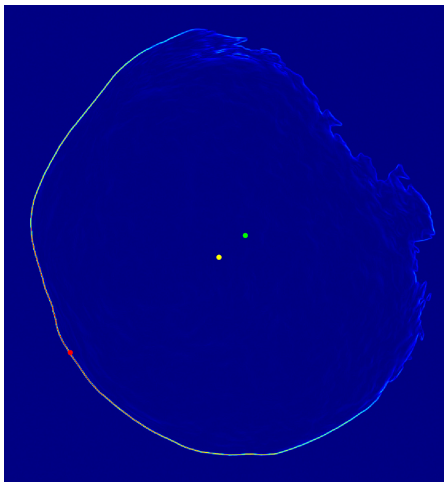


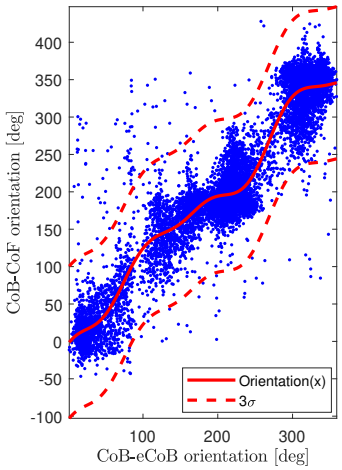
Figure 4 CoM-CoB offset as function of the length of the major axis and PA. The black surface represents the magnitude correction fit.

Moving to (branch *a*), in (*step 4a*) the cropped grayscale image around the region in which the primary is detected is considered instead of the binary mask. A filter is applied in (*step 5a*) to exacerbate differences between the soft and sharp gradient respectively over the terminator and edge of the asteroid. To do so, in this work a Sobel filter is used. In (*step 6a*) the activation map of the filter is binarized using an arbitrary threshold (σ_f, 80% the maximum value detected in each activation map). The CoB of the binary mask of the activation map is then computed as the edge CoB (eCoB) and used to provide a direction about lighting conditions. In (*step 7a*) the eCoB-CoB orientation is computed. Once again, as illustrated by comparing the eCoB-CoB orientation with the CoB-CoM one (which represents the ideal

orientation to use for the correction) a rough correlation between the two exists, as it is possible to see in Figure 5.



(a)



(b)

Figure 5 (a) Output of the Sobel filter on Didymos with the CoM (green), CoB (yellow) and eCoB (red) highlighted. (b) Correct orientation vs estimated one for all points in DB_0 (blue), of the Φ function (red) and 3σ value (dashed red).

Once again, by using a fitting function is possible to relate the estimated orientation with the ideal one. A sum of sines function is found to perform best for the fit in (*step 8a*):

$$\Phi(x) = a_1 sin(b_1 x + c_1) + a_2 sin(b_2 x + c_2) + a_3 sin(b_3 x + c_3) + a_4 sin(b_4 x + c_4) \tag{4}$$

The coefficients for this curve can be found in Appendix A. The fit error follows a Gaussian distribution with mean $\mu = 0.758$ deg and standard deviation $\sigma = 33.895$ deg, as illustrated in Figure 5. The last fundamental step is performed in (*step 9*) when the estimated magnitude and correction are combined to provide a CoF estimate using the following equation:

$$\begin{bmatrix} CoF_x \\ CoF_y \end{bmatrix} = \begin{bmatrix} CoB_x \\ CoB_y \end{bmatrix} + w \times M(PA(e), MajAxisLength) \times \begin{bmatrix} cos(\Phi(eCoB, CoB)) \\ sin(\Phi(eCoB, CoB)) \end{bmatrix} \tag{5}$$

where w is a weighting factor that can be used to tune the correction term. It is immediate to understand that when $w = 0$ the WCOB degenerates into the COB method. In this work, a value of $w = 1$ is used for all geometric conditions. It is important to remark that as in the COB and CHCOB methods, the WCOB method uses optical observable and variables which are translation and rotation invariant.

D. Neural Network

The way the WCOB works is by defining relationships existing between raw and high-level features and using the latter in nonlinear equations to generate CoF and PA estimates. As illustrated in Section 2, a lot of effort is put into designing an image processing pipeline to do so and a lot depends on the ability of the designer to spot trends and relationships as well as experience in defining the algorithm steps. A different approach could be to use the capability of a Neural Network (NN) as a universal function approximator to find relationships without explicitly defining or constraining them. Both approaches are data-driven, while the former relies on the ability of the designer, the latter lets the relationship be discovered through a machine learning approach. With the NN hyper-parameters being set, the weights and bias of the network are optimized through back-propagation given that labeled input-output relations are provided.

A sketch of the NN method is illustrated in Figure 6. The cropped grayscale image of D1 identified by an object recognition algorithm is binarized using the Otsu method. By applying a set of traditional IP methods a feature vector representing 12 properties associated to the blob of pixels of D1 is generated. The properties considered are illustrated in Figure 6. The feature vector is then used as input of a NN architecture which outputs the CoF as well as the PA. Note that to keep the method rotation and translation invariant, the coordinates of the bounding box (Γ_x and Γ_y) are not considered in the feature vector and that the CoB components are scaled with respect to them. The advantage of this approach is that the input feature vector is computed with traditional and well known IP methods, which are guaranteed to be robust and explainable. Moreover, since the NN uses a similar set of parameters as the WCOB method, it is possible to understand if there exist variables and trends that have not been identified by the designer which can be spotted through training of the NN.

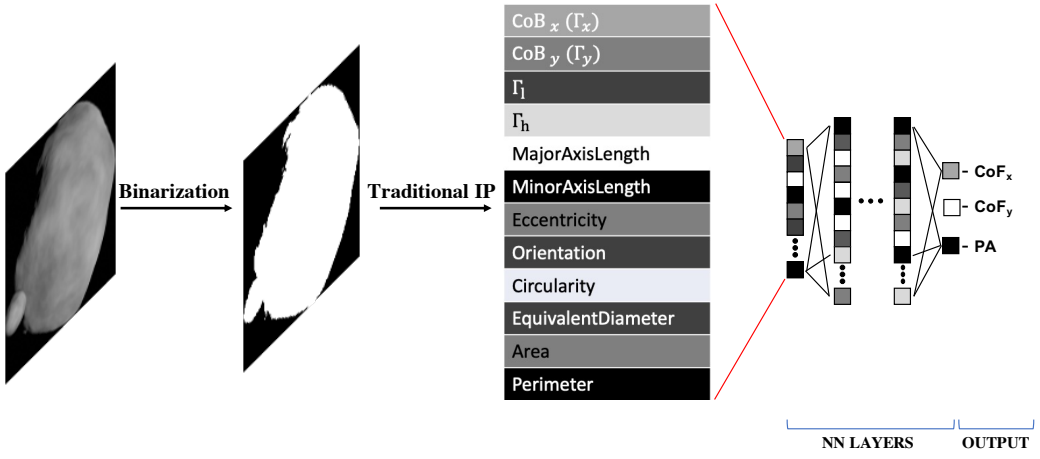


Figure 6 Sketch of the architecture of the NN method.

To train and validate the NN, the DB_0 database is divided into a $80/20\%$ partition. Three different NN architectures are instantiated: NN_1, NN_2, and NN_3. The first one is copied from the one between the fully connected and output layers of the Convolutional Neural Network (CNN) described in the next section. The second one has a deeper architecture and

departs in terms of complexity to its convolutional counterpart. The third and last one has a much deeper architecture that contains approximately the same amount of parameters of the CNN. The three architectures are summarized in Appendix B, together with the performances achieved during training and validation.

E. Convolutional Neural Network

The CNN is a type of deep architecture that exploits sequences of convolution layers and neural network layers to extract and synthesize data from images. CNNs embed the capability to correlate spatial information in the image's pixels intensity with the use of kernels whose weights and bias are determined through training. A representation of the CNN method is illustrated in Figure 7. Differently from the NN, the CNN acts directly on the grayscale images of the databases. A feature vector representing the image is not defined a priori and it is the task of the convolutional layers of the CNN to synthesize it for the NN head in the fully connected layer. Before the CNN is applied to the raw images, a pre-processing step is necessary to prepare the input data. Indeed, images are generated with ideal pointing and with a resolution set as the one of the NavCam. The first is an issue in terms of label variability. Since the CoF label would have the same value equal to the sensor center for any given image, the CNN could simply learn to output a constant value , without actually learning from the data. The second is an issue for the computational cost of the training. To overcome both, images are cropped and resized to 128×128 pixels using the bounding box Γ associated with the asteroid. During pre-processing, the CoM and CoB labels are treated accordingly and defined with respect to Γ instead. This provides variability of the labels during training and at the same time allows for the output of the CNN to be scaled back to NavCam resolution for comparison with the other methods.

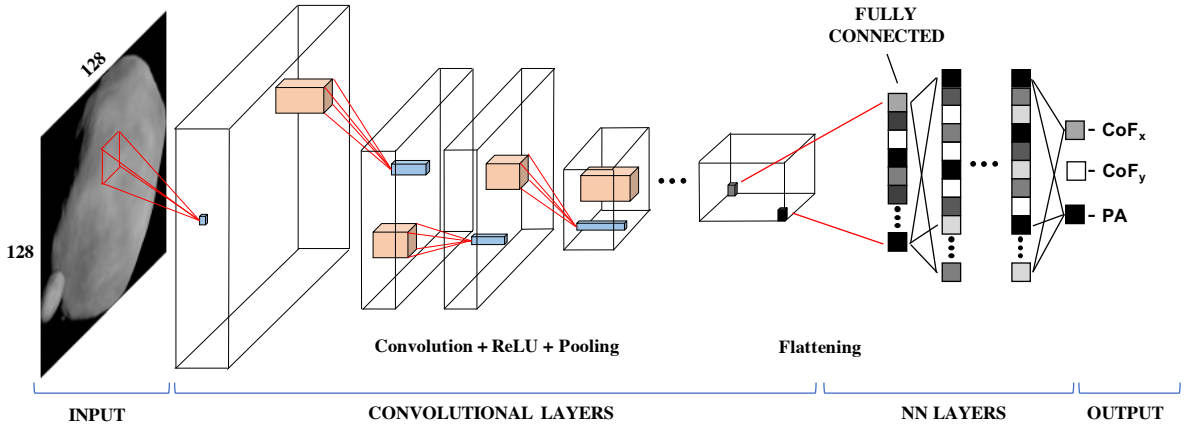


Figure 7 Sketch of the architecture of the CNN method.

The same split used to train and validate the NN is used for the CNN. The architecture is summarized in Appendix B, together with the performances achieved during training and validation. Although the CNN has not specifically been trained to be rotation invariant, this property could be introduced with data augmentation simply by generating rotated

images in the training and validation sets or with an offset of the orientation in the pre-processing step before providing the data to the CNN.

III. Case study

A. Milani mission

ESA's Hera mission will deploy two CubeSats in the proximity of the Didymos binary system in 2027: Juventas [6] and Milani [7, 8]. The primary, D1, is expected to be an irregular spheroid with an estimated diameter of 780 m while the secondary, D2, is modelled as a triaxial ellipsoid with longest axis spanning 170 m [7]. The Milani objectives are both scientific and technological: to map and study the surface composition of D1 and D2, and to demonstrate CubeSat technologies in deep space. These includes Inter-Satellite Link (ISL) with Hera and autonomous OpNav capabilities. Milani is designed as 6U CubeSat with orbital maneuvering capabilities and attitude control. The current design foresees a duration of 73 days, and is divided into four different phases: release (REL), far-range phase (FRP), close-range phase (CRP), and experimental phase (EXP). Within these phases, Milani shall be satisfying a series of payload-driven constraints for scientific acquisition [7]. The relevant end-to-end mission profile of Milani is shown in Figure 8.

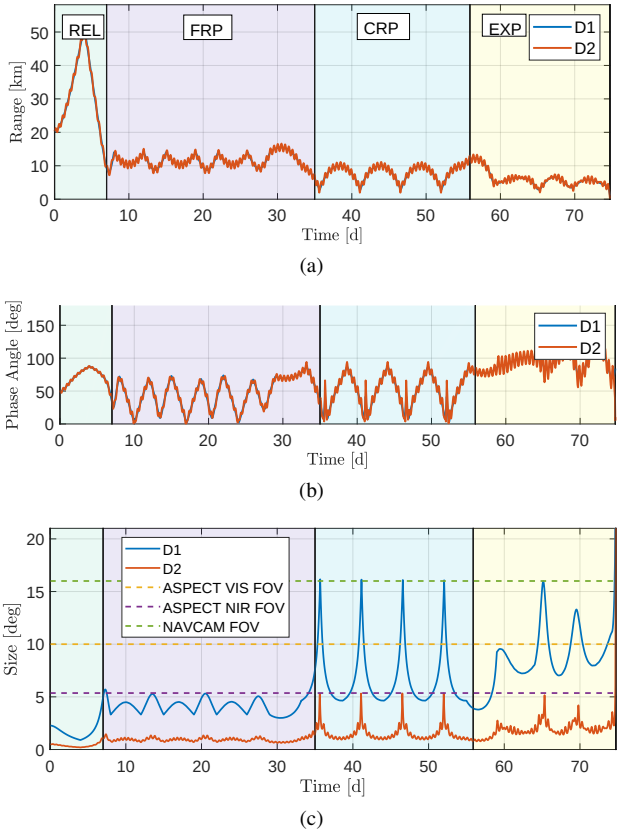


Figure 8 Milani's range (a) and phase angles (b) w.r.t. D1 and D2, and apparent size of D1 and D2 (c).

In this work, Milani's trajectory during the FRP and CRP is considered as a test case for autonomous OpNav. In Figure 9 the trajectories of these two phases are illustrated in the DidymosEquatorialSunSouth (DESS) centered in the system barycenter.

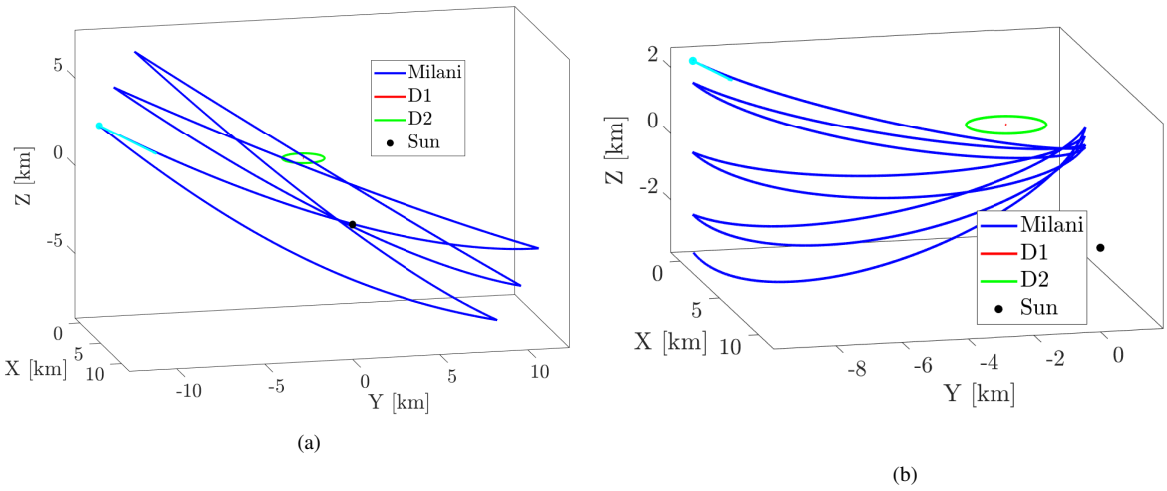
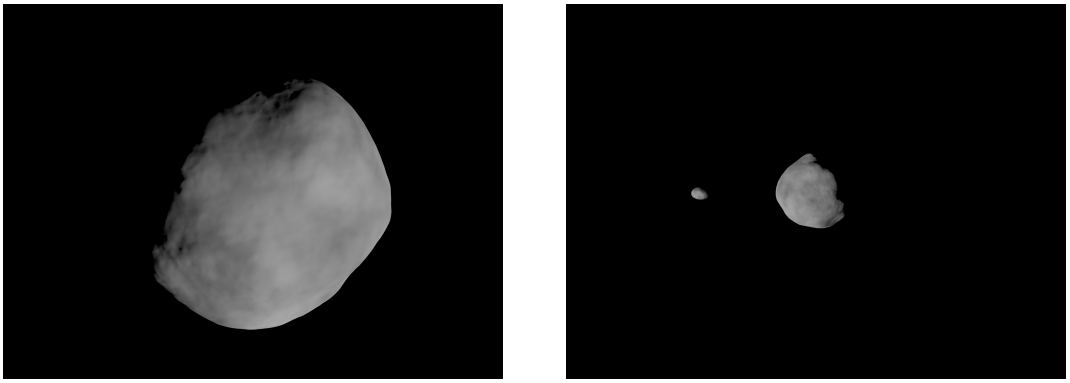


Figure 9 Milani operative orbits in the FRP (a) and CRP (b). The cyan point and arrow indicate the beginning of each phase.

The orientation of the axis of the DESS is defined such that the Z-axis points to the South pole of D1, which is assumed perpendicular to the equatorial plane described by the orbital motion of D2 about D1. The X-axis points towards the Sun projection in the equatorial plane, and the Y-axis complete the right-hand set. As it can be seen from Figure 8, the Milani range to the asteroids is in the order of 8-14 km during the FRP, and in the order of 2-11 km during the CRP (Figure 8a) while the Phase Angle (PA) never exceeds 95 deg (Figure 8b). Moreover, the apparent size of the bodies exceeds the NavCam's FOV for a limited amount of time during the CRP (Figure 8c). Two representative images of the system are illustrated in Figure 10 from 4 km and 14 km.



(a) (b)

Figure 10 Synthetic images of Didymos bodies observed from (a) 4 km and (b) 14 km with a 21×16 deg FOV NavCam.

The Milani nominal navigation strategy is based on a ground-based approach that uses radiometric tracking with Hera and optical information from D1 and D2 [7]. However, the mission will also perform an OpNav experiment involving the validation of onboard image processing and filtering methods.

In this work, the Milani test case is considered for the data-driven IP methods for different reasons. First, the system will be visited twice before the start of the mission (first by DART and a then by Hera), and therefore a rough shape model will be available to incorporate synthetic and/or real images for training. Second, the Didymos binary system allows the characterization of a robust IP method in terms of disturbance of the secondary over the primary. Third and last, there is a need for a lightweight but accurate IP method for the onboard navigation of the CubeSat, which may not be satisfied by a simple centroiding technique given the large excursion in terms of range and phase angle of the nominal trajectories. In addition to this, computational times shall be considered as well for practical implementation. Amongst the IP methods considered, it is expected that the evaluation cost would increase from the simplest (COB, $CHCOB$ and $WCOB$) to the more complex ones (NNs and CNN), the latter requiring substantially more evaluations than the former. This analysis however is out of the scope of this work and is left for future iterations.

B. Results

In this section, the results of the IP methods presented are evaluated over two databases representative of the foreseen nominal trajectory for the Milani mission: these are the *FRP* and *CRP*. The two databases are constituted respectively by $12,102$ and $12,584$ images and are obtained by sampling the Milani trajectories every 150 s. The three databases used in this work are represented in Figure 11. The NavCam used to generate the databases has a 21×16 deg FOV and a sensor 2048×1536 pixels wide.

To simplify the plot readability, a jet color-code mapping is used from blue to red in all images passing from COB, CHCOB, WCOB, NN_1, NN_2, NN_3, and CNN. The metrics used to evaluate the performances are defined as follow:

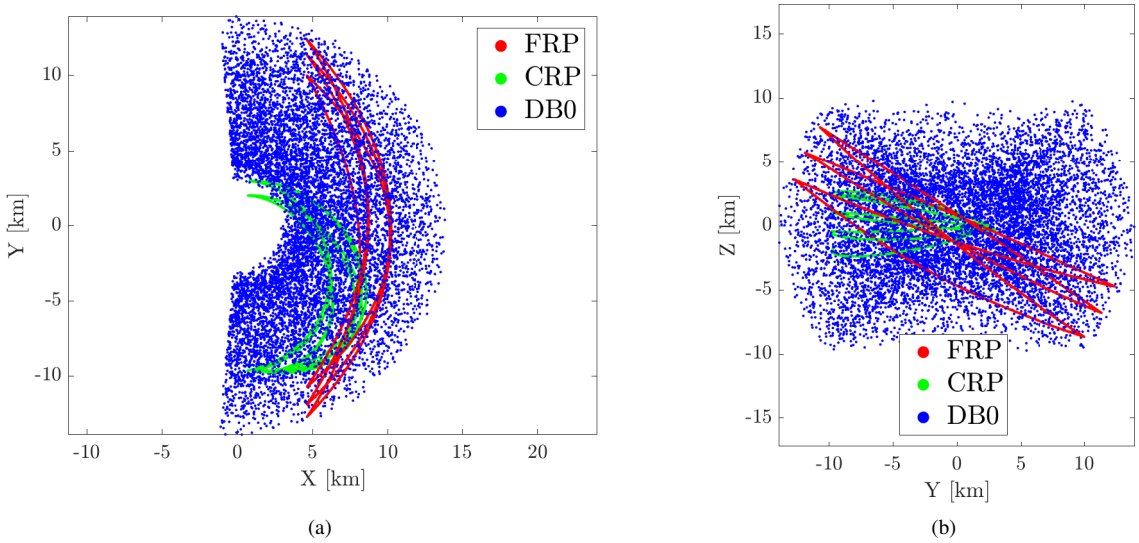


Figure 11 DB_0 **cloud of points generated with SIPTO for training/validation (blue), and for testing (red and green, representing *FRP* and *CRP* respectively).**

$$\varepsilon_{CoF} = |CoM - CoF| \tag{6}$$

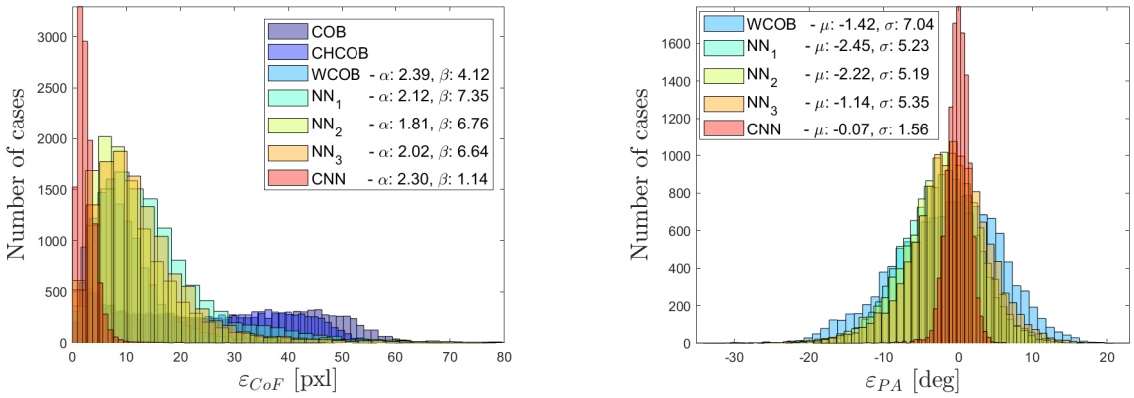
$$\varepsilon_{PA} = PA_{true} - PA_{est} \tag{7}$$

$$\varepsilon_m = 2r_{true} \cdot \tan\left(\frac{IFOV}{2} \cdot \varepsilon_{CoF}\right) \tag{8}$$

where ε_{CoF} represents the norm of the error in pixel between the estimated CoM and the real one, ε_{PA} is the error between the real phase angle and the one estimated from the image, and ε_m represents the error between the real and estimated CoM in meters computed in the image plane given the true range at which the image is taken and the Instantaneous FOV ($IFOV$) of the NavCam. The same set of plots are now presented for the performance analysis in the *FRP* and *CRP* databases.

1. Far Range Phase

The ε_{CoF} and ε_{PA} histograms in Figure 12 show in a simple way the global performances of all methods and illustrates the tendency passing from the COB to the CNN to decrease the mean and standard deviation of the probability distributions.



(a) (b)

Figure 12 Histograms of the ε_{CoF} (a) and of the ε_{PA} (b) for all IP methods in the *FRP* database.

It is possible to see that the performances can be subdivided into three main groups. First, the COB and CHCOB are roughly represented by uniform distributions across the interval from 0 to 50 pixels for the ε_{CoF}, while the ε_{PA} is not generated with these methods. Second, the WCOB, NN_1, NN_2 and NN_3 are fit by γ-distributions for the ε_{CoF} while Gaussian distributions are used for the ε_{PA}. In the legend of Figure 12 it is possible to see the associated shape and rate parameters (α, β) or mean and standard deviation (μ, σ) of the various distributions. The methods of the second group show the tendency to re-organize the ε_{CoF} into distributions with lower mean and standard deviation when compared to the COB and CHCOB. This is a direct consequence of the scattering laws being applied, implicitly or explicitly. This is valid for the third category as well, represented by the CNN, which is visually much more accurate than all other methods considered. The performance metric are also represented in numerical form in Table 1.

To appreciate the CoF accuracy decomposed by components, the error ellipses of the various methods are illustrated in Figure 13 using a 99% confidence level. From this figure it is possible to appreciate even more the higher accuracy passing from the COB to the CNN method. It is also interesting to appreciate from the COB and CHCOB how the ellipses would naturally have a larger size from the x component of the sensor (which is the one most affected by the PA) and how the WCOB and AI methods act by decreasing the ellipses size and orientation, which for the case of the WCOB is accompanied by $\sim$ 90 deg rotation. The parameters defining the error ellipses in Figure 13 are reported in Table 1, namely: a, b, θ, X_0 and Y_0 represent the the major and minor axis, the major axis orientation, and the ellipse geometrical center.

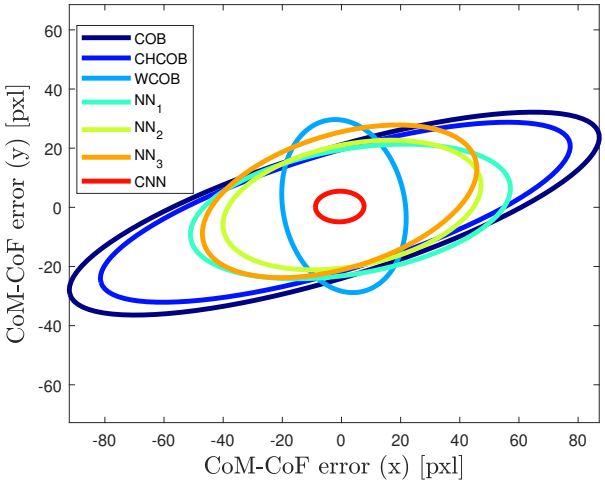


Figure 13 Error ellipses of the CoF at 99% confidence level for all IP methods in *FRP* database.

The ε_{CoF} and ε_{PA} are also reported as function of the phase duration in Figure 14. Only 7 days are displayed for clarity, owing to the repetitiveness of the pattern. By comparing Figure 14 with the range and PA curves in Figure 8 it is possible to see that the COB and CHCOB achieve the worst performances at high PA (which also coincide with a higher distance from D1).

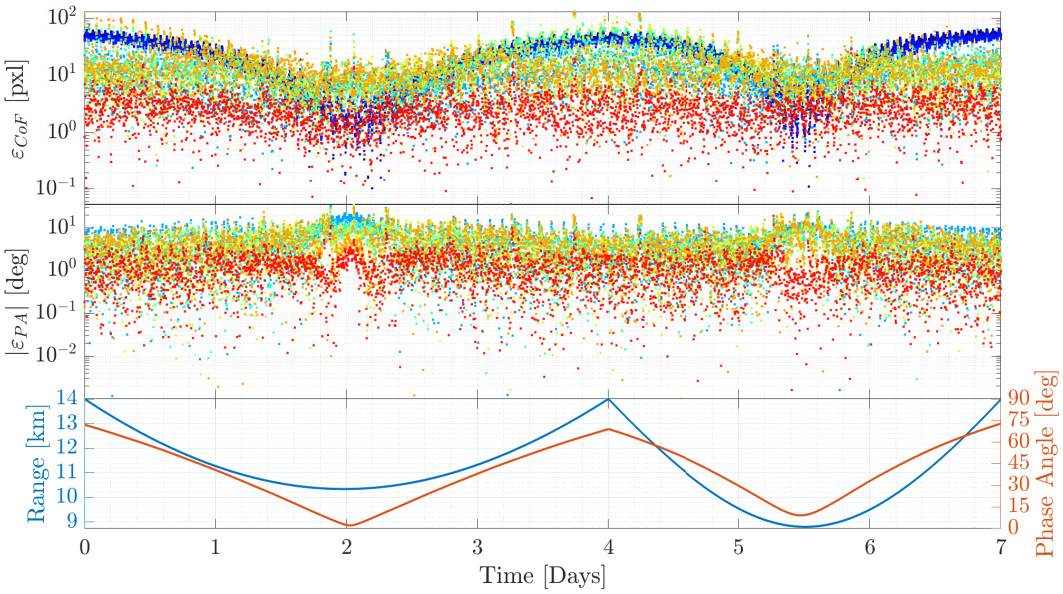


Figure 14 Performances of the CoF and PA estimates of the different IP methods in the *FRP* database with range and PA as function of time.

Table 1 Performance metrics of all IP methods in the *FRP* database.

Metric	COB	CHCOB	WCOB	NN_1	NN_2	NN_3	CNN
ε_{CoF}							
μ [pxl]	26.829	23.632	9.838	15.607	12.230	13.430	2.611
σ [pxl]	17.030	15.305	6.369	10.708	9.091	9.444	1.724
a [pxl]	93.461	82.720	29.560	54.770	44.292	48.063	8.219
b [pxl]	21.690	19.842	20.616	21.130	20.568	22.564	5.177
θ [deg]	196.997	196.699	101.473	189.647	191.158	197.172	181.296
X_0 [pxl]	-2.429	-2.082	0.841	3.022	3.568	-0.630	-0.581
Y_0 [pxl]	-2.129	-1.662	0.420	-1.408	0.780	2.032	0.270
ε_m							
μ [m]	57.320	50.544	20.163	32.202	24.838	27.114	5.283
σ [m]	40.788	36.614	13.792	23.452	18.987	19.488	3.590
ε_{PA}							
μ [deg]	n.a	n.a	-1.417	-2.447	-2.218	-1.141	-0.070
σ [deg]	n.a	n.a	7.040	5.228	5.193	5.348	1.560

This is expected and motivates the need for scattering laws. It is possible to see how the methods developed from the WCOB to the CNN thrive in terms of performances in these conditions. However, it is possible to appreciate an inverse trend for low values of PA, where the best performing methods are the COB, CHCOB, and CNN. These methods locally outperform the others around day 2 and 5.5, which are portions of the Milani trajectory associated with low values of PA. In these conditions, it seems that the correction performed in the WCOB and NN is counterproductive, making the CoF estimate deviate as high as 30 pixels. This has been observed directly on the WCOB to be driven mainly by two sources of error. The first is due to an error in the PA estimate, which is affected by larger noise when low a PA is considered, as illustrated in Figure 3. The second is given by an orientation error which is caused by a much softer difference between terminator and edge region at low PA, which causes the *branch b* of the WCOB to not properly assess the correct orientation. Since in these cases both the magnitude and orientation terms of the CoF correction are highly degraded, the CoF estimate is highly inaccurate in these conditions. Interestingly, this drawback seems to impacts both the WCOB as well as the NN method but not the CNN one, hinting at the fact that it may be introduced by the image descriptors used and that it can be addressed with the use of further spatial information from the image. Lastly, it is highlighted that the WCOB can be easily adapted to this scenario by setting a weighting factor w equal to 0 below a predetermined value of PA.

Another interesting phenomenon is observed in Figure 14. Looking at it carefully, for example around days 1.89, 3.29, 4.25 and others, it is possible to identify isolated points with suddenly unstable performances. These are caused by the perturbing effect in the images of D2. These perturbations are observed to be small whenever in the image D2 spans entirely over D1, casts shadows, or in both cases. On the other hand, they are observed to have a large impact on all methods when D2 is touching the edge of D1. Both these cases are illustrated in Figure 15. The latter is because the object recognition algorithm is not capable to distinguish between D1 and D2 and in these cases the image properties associated with D1 also encompass the blob of pixels of D2, which is offsetting them. This is true for all methods but seems to be less severe for the CNN, which may have learned through training to better handle these cases.

Eventually, to have a global view of the best-performing IP method, the COB, WCOB, NN (represented by the NN_2 instance), and CNN are compared in terms of the smallest ε_{CoF}. In Figure 16 it is possible to appreciate the best-performing methods as function of the range from D1 and PA. As discussed before, the COB is considered the best at lower PA, although its performance is contested by the CNN in this interval. Globally, the COB is the best for only 3.40% of the points. The WCOB and NN are found to be the best especially in the low to medium and medium to high intervals of PA, covering respectively the 7.30% and 5.14% of the points. Finally, the CNN is considered the best performing method across a wide range of conditions since 84.17% of the points is performed with the smallest error with this method.

(a) (b)

Figure 15 Examples of D2 disturbances when (a) D2 eclipses D1 and (b) D2 shape touches the edge of D1.

2. Close Range Phase

In this section, the analysis performed before for the *FRP* is now presented for the *CRP* database. Only the relevant differences in terms of performances are given. Differently from the *FRP*, the *CRP* of the Milani mission reaches ranges below 4 km, which is considered the lower limit in the DB_0 database. For this reason, the performances are illustrated in detail for the points above 4 km, while they are briefly examined for those below 4 km, 1006 in total, as an interesting

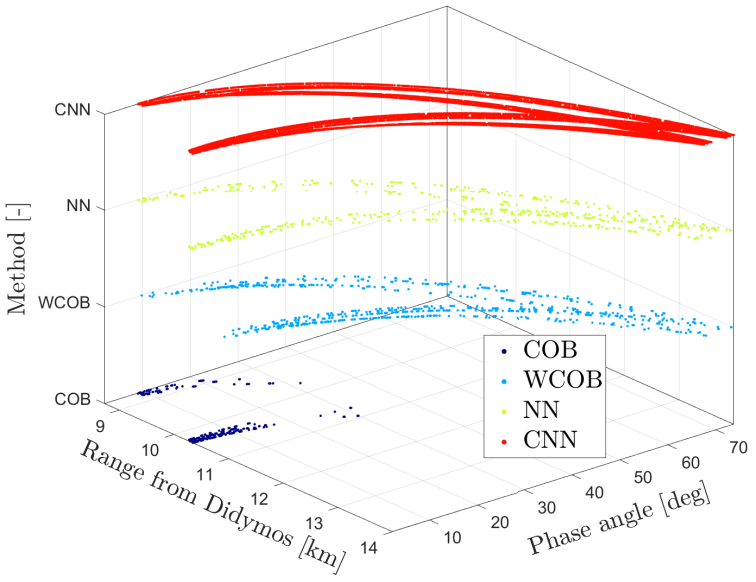


Figure 16 Scatter-plot of the method with the smallest ε_{CoF} as function of the range and PA in FRP.

example of applications of data-driven methods applied outside their design parameter space.

The ε_{CoF} and ε_{PA} histograms in Figure 17 show in a simple way the global performance of all methods. The trends identified in the *FRP* database are observed also in this case. The only difference can be seen in the range of ε_{CoF} which spans a larger interval as a consequence of this phase taking place closer to D1 than the FRP. It is also observed that the CNN distribution is closer to the ones of the NN and WCOB than in Figure 12. Once again, all data used to plot the histograms in Figure 17 are represented in numerical form in Table 2.

The error ellipses of the various methods are illustrated in Figure 18 using a 99% confidence level. As for the *CRP* case, it is interesting to see how the ellipses would naturally have a larger size with the COB and CHCOB in the x component.

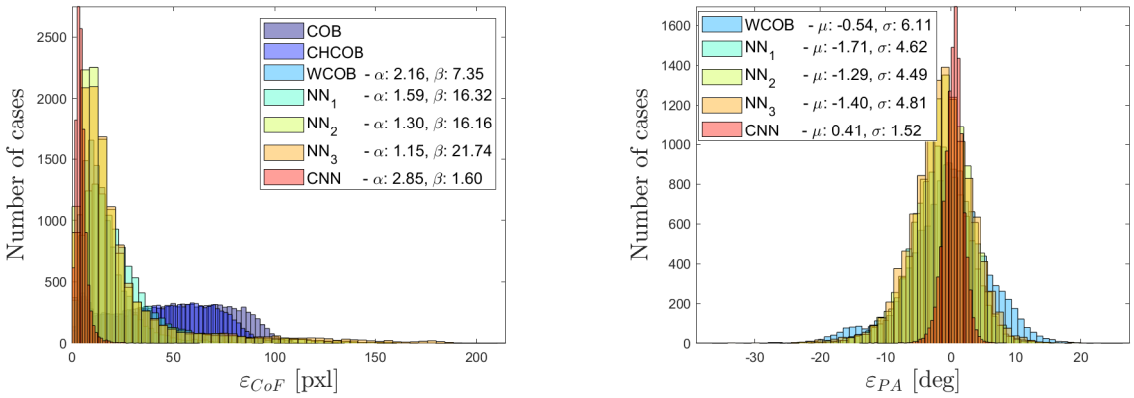


Figure 17 Histograms of the ε_{CoF} (a) and of the ε_{PA} (b) for all IP methods in the *CRP* database.

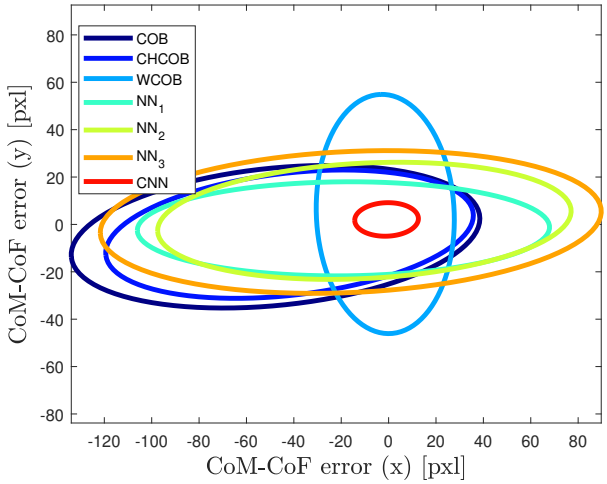


Figure 18 Error ellipses of the CoF at 99% confidence level for all IP methods in *CRP* database.

In this case however, the COB and CHCOB ellipses develops predominantly over the region with $x < 0$. This is a direct consequence of the CRP trajectory, which is not symmetrical around the X-axis of the DESS reference frame, as seen in Figure 10. The orientation of the error ellipses of all other IP methods has retained the same orientation of the *FRP* database. In *CRP* however, the NN methods have larger ellipses than the COB and CHCOB.

The ε_{CoF} and ε_{PA} are also plotted as function of the phase duration in Figure 19, the same considerations drawn from Figure 14 hold true in this case. The points below 4 km are omitted in Figure 19 for clarity as they belong to a region outside the one considered for training. By comparing the performances above and below 4 km in Table 2 and Table 3 it is possible to see that in the latter case the WCOB and the NN are highly degraded while the COB, CHCOB, and CNN are not. The COB and CHCOB perform reasonably well since the points closer to D1 are also associated with the lowest PA and the CoF estimate in these methods is still retaining accuracy due to the fairly regular shape of D1.

On the other hand WCOB, NN and CNN are all data-driven approaches, thus it is expected that they would suffer degraded performances when tested outside their design range. This is true for the WCOB and NN, but surprisingly it is not the case for the CNN, which actually is confirmed the best also below 4 km. This behavior is explained by the pre-processing step used in the CNN method for the cases in which D1 is fully resolved in the FOV and below 4 km. In these conditions, when D1 edges are still not cut by the FOV, the scaling and resizing of the images around the bounding box is a determining factor in having the D1 appear in the image as seen from further away. A similar concept called "telescopic range" has been used actively as means to extend the functioning interval of data-driven algorithms in [35]. This phenomenon is responsible for the good performances of the CNN in these ranges.

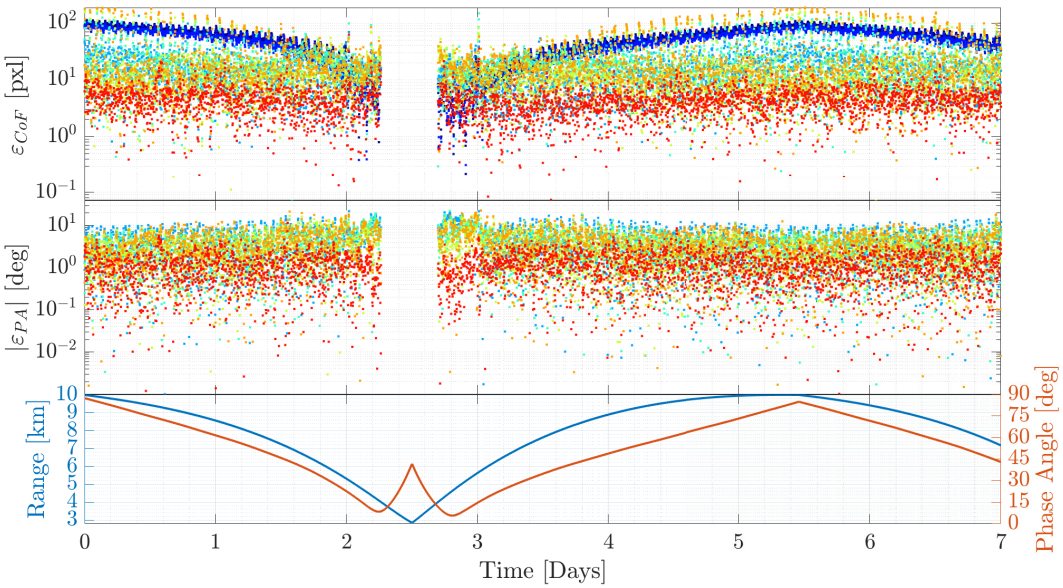


Figure 19 Performances of the CoF and PA estimates of the different IP methods in the *CRP* database with range and PA as function of time. Points below 4 km are omitted.

Finally, to have a global view of the best-performing IP method, the COB, WCOB, NN (represented by the NN_2 instance), and CNN are compared in terms of the smallest ε_{CoF}. In Figure 20 it is possible to appreciate the best performing method as function of the range from D1 and PA, the first limited to values above 4 km. Similar results to the ones in *FRP* are obtained. Globally, the COB is the best for only 3.34% of the points. The WCOB and NN are found to be the best for 7.82% and 9.56% of the points respectively. Finally, the CNN is considered the best performing method across a wide range of conditions since 79.28% of the points achieve the smallest error with this method.

Table 2 Performances metrics of all IP methods in the *CRP* database for the points above 4 km.

Metric	COB	CHCOB	WCOB	NN_1	NN_2	NN_3	CNN
ε_{CoF}							
μ [pxl]	49.496	43.480	15.860	25.940	20.944	25.064	4.562
σ [pxl]	27.594	24.680	10.797	20.575	18.396	23.341	2.701
a [pxl]	86.633	78.051	50.519	87.017	87.387	105.846	13.447
b [pxl]	28.978	25.754	29.097	19.845	24.350	29.766	7.067
θ [deg]	185.724	186.542	92.405	180.461	182.927	182.629	182.024
X_0 [pxl]	-47.732	-41.824	-1.413	-19.072	-10.227	-15.978	-0.864
Y_0 [pxl]	-5.193	-4.115	4.401	-1.854	1.515	1.024	2.100
ε_m							
μ [m]	79.802	70.123	24.046	39.913	31.389	38.246	6.737
σ [m]	51.835	46.211	18.255	34.834	29.083	37.877	4.371
ε_{PA}							
μ [deg]	n.a	n.a	-0.542	-1.709	-1.289	-1.403	0.408
σ [deg]	n.a	n.a	6.109	4.624	4.491	4.806	1.522

Table 3 Performances metrics of all IP methods in the *CRP* database for the points below 4 km.

Metric	COB	CHCOB	WCOB	NN_1	NN_2	NN_3	CNN
ε_{CoF}							
μ [pxl]	28.041	24.073	85.318	54.939	156.873	158.551	10.196
σ [pxl]	23.388	20.190	98.285	49.678	114.699	105.920	7.785
a [pxl]	73.994	67.350	297.627	146.243	299.376	277.838	45.570
b [pxl]	32.008	28.192	78.603	43.389	43.549	37.531	14.762
θ [deg]	171.304	176.778	168.547	199.764	45.998	220.227	173.671
X_0 [pxl]	24.528	19.927	76.704	49.916	118.960	118.682	-7.394
Y_0 [pxl]	-5.114	-3.385	-14.994	13.009	99.920	104.334	1.978
ε_m							
μ [m]	15.812	13.615	46.793	31.043	89.651	91.001	6.013
σ [m]	11.939	10.378	50.839	26.086	59.136	54.005	4.415
ε_{PA}							
μ [deg]	n.a	n.a	-0.701	3.405	-0.752	1.490	-1.043
σ [deg]	n.a	n.a	6.759	7.266	5.663	6.652	2.292

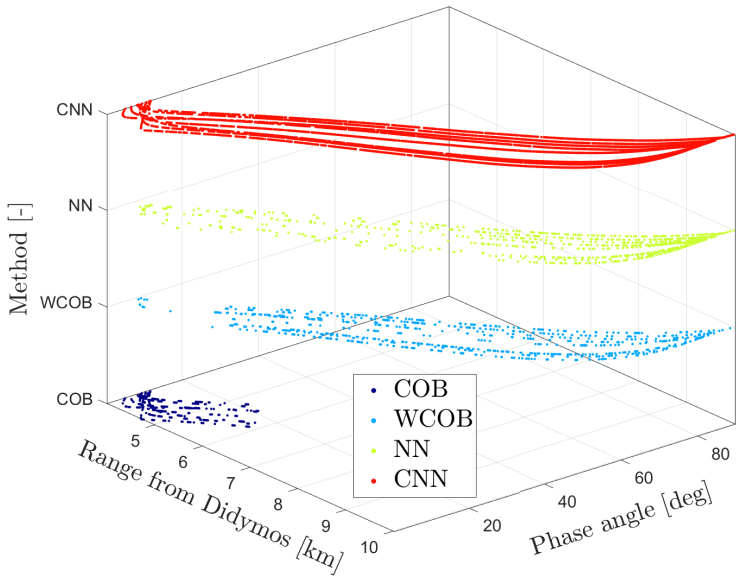


Figure 20 Scatter-plot of the method with the smallest ε_{CoF} as function of the range and PA in *CRP*.

IV. Neural networks explainability

In this section an attempt is made to explain the rationale behind the functioning of the NN methods. This analysis is motivated by the need to better understand which inputs are meaningful to generate the output. This is useful to understand the underlying IP pipeline of the NN method but also as a way to understand whether certain relationships and parameters neglected in the design of the WCOB method could play a role in the generation of the output. This analysis is performed on the NN_2 network, representative for all NN methods considered.

To do so, SHAP [43] values are used. These implement a game theory approach which breaks down the contribution given by each player to the results of a game. Analogously for the case considered in this work, SHAP values quantify the contribution of each feature on the model prediction, independently from its complexity. They do so following a formulation and statistical properties which are discussed in detail in [43]. In a nutshell, SHAP values give an interpretation of the impact of a given feature exploring all the possible model predictions generated by a coalition between such feature and the remaining ones. Doing so for all the possible combinations however would not be tractable, so approximations and samplings are necessary as explained in [43]. The SHAP values can then be used as a proxy to visualize the impact of a given feature on the model prediction, the higher is the SHAP value in magnitude for a particular feature, the higher is its effect in the output.

The SHAP values of a sample of 2500 random cases from *DB0* of the NN_2 predictions are computed with a kernel explainer. From Figure 21 it is possible to see from a global perspective the impact of each feature looking at the mean of of the absolute SHAP values.

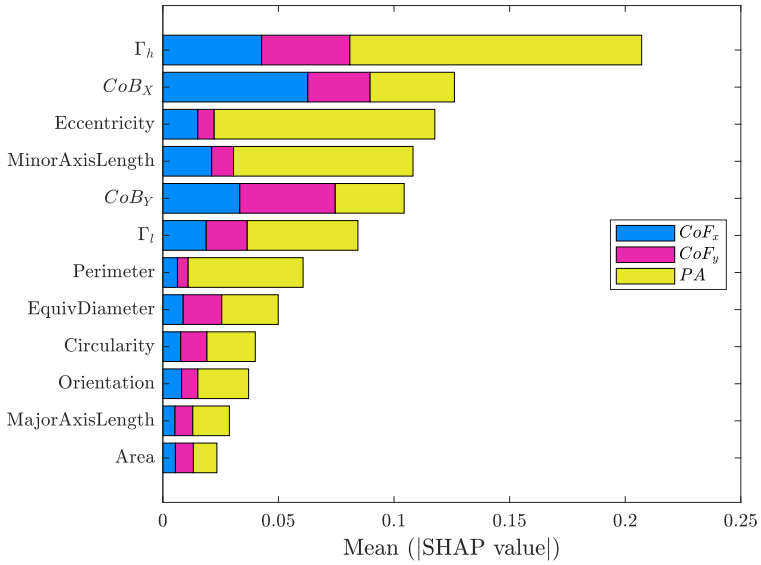


Figure 21 Stacked histogram of the mean absolute SHAP values of the NN output for each element of the input feature vector.

In particular, it is interesting to note that the PA is mostly driven by Γ_h, eccentricity and length of the minor axis. Of these features, only the eccentricity is range invariant. In contrast to the WCOB, the NN seems capable to synthesize a more accurate PA estimate when combining the eccentricity with the height of the bounding box and the minor axis length. This hints to the fact that it would be possible to have a slight improvement in the PA estimate than the one of the WCOB when considering multiple optical observables. The complexity of such a new formulation has to be balanced against the expected improvement. On the other hand, the CoF estimate is driven in the x component by the CoB_x and Γ_h while in the y component by CoB_y and Γ_h. This hints to a possible relationship found by the NN between CoF coordinates with CoB coordinates and Γ_h. The NN may have learnt to correlate a correction between CoF and CoB and to scale it properly as a function of the range from the body by exploiting Γ_h as a proxy for the range. Also this case offers a reflection for a possible change in the features selected in the WCOB method. Alternatively, the SHAP values are illustrated case by case for the three output of the NN_2. In Figure 22 it is interesting to observe the correlation effects given by combinations of low-high features and low-high SHAP values. For example, it is possible to observe how the CoF_x and CoF_y are largely influenced by the CoB_x and CoB_y respectively. The former however show polarization between SHAP values and feature values: low values of the CoB_x tends to output low values of CoF and vice-versa. However the same is not true for the CoB_y, which make sense since the geometry of the problem (and consequently the PA) is affecting the estimate mostly around the CoF_x component. Another interesting effect to take note of is the behaviour of Γ_l and Γ_h for the CoF: in both components, high values of Γ_l are related to low values of the CoF components, while high values of Γ_h are related with high values of the CoF components.

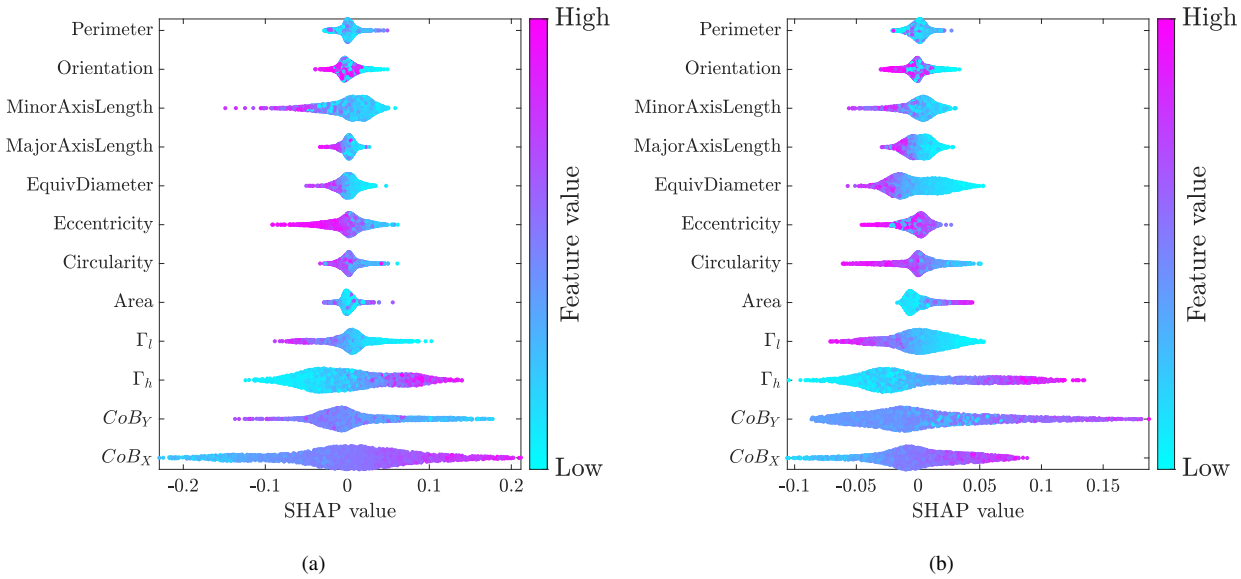


Figure 22 SHAP values of the input features of the NN method for the CoF_x (a) and CoF_y (b) outputs, colored by the feature's value.

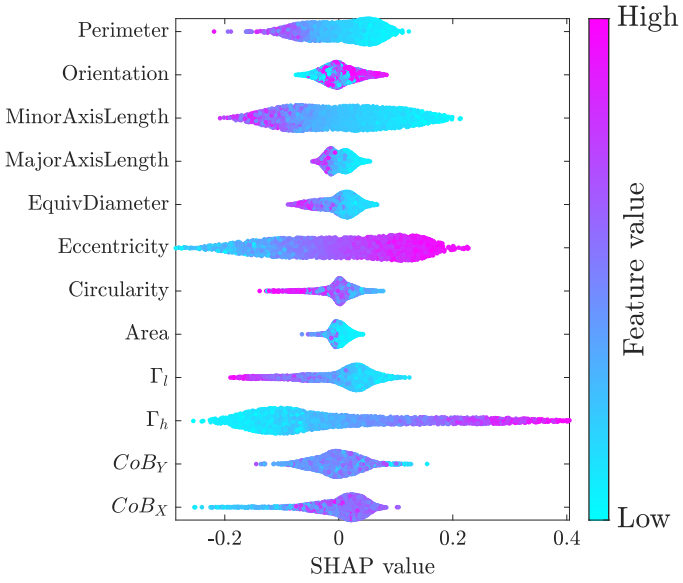


Figure 23 SHAP values of the input features of the NN method for the PA output, colored by the feature's value.

Finally, the SHAP values are analyzed for the PA estimate in Figure 23. It is very interesting to observe that the major contribution to the output are in order Γ_h, eccentricity, length of the minor axis, and perimeter (as seen in Figure 21). All of these parameters show a strong polarization between feature values and SHAP values. Of these quantities, the eccentricity is the only one which is range invariant. High values of eccentricity corresponds to high values of PA and

vice-versa, which is the same sort of relationship exploited in the WCOB method (see Figure 3). It is then interesting to observe that also Γ_h, length of the minor axis and perimeter could be exploited efficiently in the PA estimate, as high values of PA seems to be correlated with low values of perimeter and minor axis length and higher values of Γ_h and vice-versa. All these quantities can be considered as a proxy of the eccentricity, although they are not range dependant.

V. Conclusion

In this work, five different IP methods have been assessed for possible applications for a CubeSat mission around a binary system. The Didymos binary asteroid system observed by the Milani mission has been exploited as a use case. All tuning parameters have been set over a global database, while they have been tested on two databases representative of the two main phases of the Milani mission.

The COB and CHCOB are used as the baseline for a traditional IP approach. All other methods perform better than these, mainly because scattering laws are put in place explicitly or implicitly to tackle down the known issues related to these traditional approaches. The WCOB, NN, and CNN are all data-driven methods. However, while the WCOB bases its foundation on traditional IP techniques, the NN and CNN are based on the machine learning paradigm. In the WCOB a scattering law is empirically derived from data and applied to the COB estimate. This method has commonalities between AI and traditional approaches and its one of the main contributions of this work. It differs from the first one because the optical features extracted are fully known and because the interpolating functions used are expressed by simple functions. However, similarly to AI methods, being a data-driven approach it needs data to tune its hyperparameters. An interesting result is that this method performs at the same level as NN methods. This means that the WCOB presented in this work is a traditional method (and fully explainable) capable to unleash the performances of a NN which operate with a similar set of input. Moreover, a brief explainability analysis of the NN method suggests that the proper set of variables have been identified to design the method and that through the usage of the highly complex nonlinear evaluation functions of a deep-learning NN only a marginal improvement is expected. The WCOB method being fully explainable means that it can be deployed with much more confidence than a NN on a flying mission because its main blocks are well understood, robust, and known in the literature. On the other hand, if much higher accuracy is sought, the CNN is well capable to outperforms all other methods. This is because the CNN capability to extract spatial information is a key functionality in improving the accuracy of the optical observable.

The difference in performances between WCOB, NN, and CNN are explained by the type of input and methods adopted. The NNs work with explicit features determined from traditional IP, the WCOB uses a subset of these and one filter, while the CNN uses a large sample of implicit features and filters which are not defined a-priori but determined through training. The NN proves that filters are not needed to have good performances if highly complex evaluation functions are used. On the other hand, the CNN is designed with a large number of filters to extract data from images. With such capability in place, performances are boosted above the ones of the NN and WCOB. In the middle, there is the WCOB

method, which demonstrates that traditional IP functions, one filter, and few features are enough for good results. In the current WCOB implementation, a single filter is used on the image, future work would go into the direction of including multiple filters to attempt boosting the performance towards those of the CNN. All methods illustrated in this work can be adapted easily to different shapes, an analysis that compares the performance of the methods also across different shape profiles would also be valuable. Additional output useful for state estimation can also be included such as the range from D1. Finally, variants of the WCOB with different weighting parameters w shall be investigated.

Acknowledgments

M.P. and F.T. would like to acknowledge the funding received from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813644. The authors would also like to acknowledge the support of the entire Milani team, lead by Tyvak International.

References

[1] Walker, R., Binns, D., Bramanti, C., Casasco, M., Concari, P., Izzo, D., Feili, D., Fernandez, P., Fernandez, J. G., Hager, P., and Koschny, D., "Deep-space CubeSats: thinking inside the box," *Astronomy & Geophysics*, Vol. 59, No. 5, 2018, pp. 5–24. doi:10.1093/astrogeo/aty237.

[2] Michel, P., Cheng, A., and Küppers, M., "Asteroid Impact and Deflection Assessment (AIDA) mission: science investigation of a binary system and mitigation test," *European Planetary Science Congress*, Vol. 10, 2015, pp. 123–124.

[3] Atchison, J. A., Ozimek, M. T., Kantsiper, B. L., and Cheng, A. F., "Trajectory options for the DART mission," *Acta Astronautica*, Vol. 123, 2016, pp. 330–339. doi:10.1016/j.actaastro.2016.03.032.

[4] Dotto, E., Della Corte, V., Amoroso, M., Bertini, I., Brucato, J., Capannolo, A., Cotugno, B., Cremonese, G., Di Tana, V., Gai, I., and Ieva, S., "LICIACube-the Light Italian Cubesat for Imaging of Asteroids in support of the NASA DART mission towards asteroid (65803) Didymos," *Planetary and Space Science*, 2021, p. 105185. doi:10.1016/j.pss.2021.105185.

[5] Michel, P., Küppers, M., and Carnelli, I., "The Hera mission: European component of the ESA-NASA AIDA mission to a binary asteroid," *COSPAR Scientific Assembly*, Pasadena, California, 2018, pp. 1–42.

[6] Goldberg, H. R., Karatekin, Ö., Ritter, B., Herique, A., Tortora, P., Prioroc, C., Gutierrez, B. G., Martino, P., and Carnelli, I., "The Juventas CubeSat in Support of ESA's Hera Mission to the Asteroid Didymos," *Small Satellite Conference*, 2019, pp. 1–7.

[7] Ferrari, F., Franzese, V., Pugliatti, M., Giordano, C., and Topputo, F., "Preliminary mission profile of Hera's Milani CubeSat," *Advances in Space Research*, Vol. 67, No. 6, 2021, pp. 2010–2029. doi:10.1016/j.asr.2020.12.034.

[8] Ferrari, F., Franzese, V., Pugliatti, M., Giordano, C., and Topputo, F., "Trajectory Options for Hera's Milani CubeSat Around (65803) Didymos," *The Journal of Astronautical Sciences*, 2021. doi:10.1007/s40295-021-00282-z.

[9] Gil-Fernandez, J., and Ortega-Hernando, G., "Autonomous vision-based navigation for proximity operations around binary asteroids," *CEAS Space Journal*, Vol. 10, No. 2, 2018, pp. 287–294. doi:10.1007/s12567-018-0197-5.

[10] Bhaskaran, S., Riedel, J. E., and Synnott, S. P., "Autonomous nucleus tracking for comet/asteroid encounters: the Stardust example," *1998 IEEE Aerospace Conference Proceedings*, Vol. 2, 1998, pp. 353–365.

[11] Wright, C., Liounis, A. J., and Ashman, B., "Optical navigation algorithm performance," , 2018. 1st Annual RPI Workshop on Image-Based Modeling and Navigation for Space Applications, Troy, NY.

[12] Liounis, A. J., "Limb Based Optical Navigation for Irregular Bodies," , 2018. 1st Annual RPI Workshop on Image-Based Modeling and Navigation for Space Applications, Troy, NY.

[13] Christian, J. A., "Optical Navigation Using Planet's Centroid and Apparent Diameter in Image," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 2, 2015, pp. 192–204. doi:10.2514/1.g000872.

[14] Christian, J. A., and Robinson, S. B., "Noniterative Horizon-Based Optical Navigation by Cholesky Factorization," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 12, 2016, pp. 2757–2765. doi:10.2514/1.g000539.

[15] Christian, J., "Accurate planetary limb localization for image-based spacecraft navigation," *Journal of Spacecraft and Rockets*, Vol. 54, No. 3, 2017, pp. 708–730. doi:https://doi.org/10.2514/1.A3369210.2514/1.A33692.

[16] Teil, T., Schaub, H., and Kubitschek, D., "Centroid and Apparent Diameter Optical Navigation on Mars Orbit," *Journal of Spacecraft and Rockets*, Vol. 58, No. 4, 2021, pp. 1107–1119. doi:10.2514/1.a34815.

[17] Owen, W. M. J., "Methods of optical navigation," *AAS Spaceflight Mechanics Conference, New Orleans, Louisiana*, 2011, pp. 1–19.

[18] Pellacani, A., Graziano, M., Fittock, M., Gil, J., and Carnelli, I., "HERA vision based GNC and autonomy," *Proceedings of the 8th European Conference for Aeronautics and Space Sciences. Madrid, Spain, 1-4 july 2019*, 2019. doi:10.13009/EUCASS2019-39.

[19] Qian, W., Wei, Z., De, X., and Xiaoyan, M., "Model-based line-of-sight detection of an irregular celestial body for autonomous optical navigation," *2015 34th Chinese Control Conference (CCC)*, 2015, pp. 5527–5532. doi:10.1109/ChiCC.2015.7260503.

[20] Baker, D. A., and McMahon, J. W., "Limb-based shape modeling and localization for autonomous navigation around small bodies," *2020 Astrodynamic Specialist Conference, Lake Tahoe*, AAS/AIAA, 2020.

[21] De Santayana, R. P., and Lauer, M., "Optical measurements for rosetta navigation near the comet," *Proceedings of the 25th International Symposium on Space Flight Dynamics (ISSFD), Munich*, 2015.

[22] Mario, C., and Debrunner, C., "Robustness and performance impacts of optical-based feature tracking to OSIRIS-REx asteroid sample collection mission," *39th Annual AAS Guidance and Control Conference*, 2015.

[23] Glassmeier, K.-H., Boehnhardt, H., Koschny, D., Kührt, E., and Richter, I., "The Rosetta mission: flying towards the origin of the solar system," *Space Science Reviews*, Vol. 128, No. 1, 2007, pp. 1–21. doi:10.1007/s11214-006-9140-8.

[24] Lauretta, D. S., Balram-Knutson, S. S., Beshore, E., Boynton, W. V., Drouet d'Aubigny, C., DellaGiustina, D. N., Enos, H. L., Golish, D. R., Hergenrother, C. W., Howell, E. S., Bennett, C. A., Morton, E. T., Nolan, M. C., Rizk, B., Roper, H. L., Bartels, A. E., Bos, B. J., Dworkin, J. P., Highsmith, D. E., Lorenz, D. A., Lim, L. F., Mink, R., Moreau, M. C., Nuth, J. A., Reuter, D. C., Simon, A. A., Bierhaus, E. B., Bryan, B. H., Ballouz, R., Barnouin, O. S., Binzel, R. P., Bottke, W. F., Hamilton, V. E., Walsh, K. J., Chesley, S. R., Christensen, P. R., Clark, B. E., Connolly, H. C., Crombie, M. K., Daly, M. G., Emery, J. P., McCoy, T. J., McMahon, J. W., Scheeres, D. J., Messenger, S., Nakamura-Messenger, K., Righter, K., and Sandford, S. A., "OSIRIS-REx: sample return from asteroid (101955) Bennu," *Space Science Reviews*, Vol. 212, No. 1, 2017, pp. 925–984. doi:10.1007/s11214-017-0405-1.

[25] Watanabe, S. i., Tsuda, Y., Yoshikawa, M., Tanaka, S., Saiki, T., and Nakazawa, S., "Hayabusa2 mission overview," *Space Science Reviews*, Vol. 208, No. 1, 2017, pp. 3–16. doi:10.1007/s11214-017-0377-1.

[26] Ogawa, N., Terui, F., Yasuda, S., Matsushima, K., Masuda, T., Sano, J., Hihara, H., Matsuhisa, T., Danno, S., Yamada, M., Mimasu, Y., Yoshikawa, K., Ono, G., Yokota, Y., Saiki, T., and Tsuda, Y., "Image-based Autonomous Navigation of Hayabusa2 using Artificial Landmarks: Design and In-Flight Results in Landing Operations on Asteroid Ryugu," *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, 2020. doi:10.2514/6.2020-0225.

[27] Lucas, B. D., and Kanade., T., "An Iterative Image Registration Technique with an Application to Stereo Vision." *Proceedings of Imaging Understanding Workshop*, 1981, pp. 121–130.

[28] Krizhevsky, A., Ilya, S., and E., H. G., "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012, pp. 1097–1105. doi:10.1145/3065386.

[29] Ronneberger, O., Fischer, P., and Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," Springer International Publishing, 2015, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

[30] Badrinarayanan, V., Kendall, A., and Cipolla, R., "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 12, 2017, pp. 2481–2495. doi:10.1109/TPAMI.2016.2644615.

[31] Izzo, D., Märtens, M., and Pan, B., "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," *Astrodynamics*, Vol. 3, No. 4, 2019, pp. 287–299. doi:10.1007/s42064-018-0053-6.

[32] Quadrelli, M. B., Wood, L. J., Riedel, J. E., McHenry, M. C., Aung, M., Cangahuala, L. A., Volpe, R. A., Beauchamp, P. M., and Cutts, J. A., "Guidance, Navigation, and Control Technology Assessment for Future Planetary Science Missions," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 7, 2015, pp. 1165–1186. doi:10.2514/1.g000525.

[33] Pugliatti, M., and Topputo, F., "Small-Body Shape Recognition with Convolutional Neural Network and Comparison with Explicit Features Based Methods," *AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe*, 2020.

[34] Pugliatti, M., Maestrini, M., di Lizia, P., and Topputo, F., "On-board Small-Body Semantic Segmentation Based on Morphological Features with U-Net," *31st AAS/AIAA Space Flight Mechanics Meeting, Charlotte*, 2021.

[35] Pugliatti, M., and Topputo, F., "Navigation about irregular bodies through segmentation maps," *31st AAS/AIAA Space Flight Mechanics Meeting, Charlotte*, 2021.

[36] Teil, T., Bateman, S., and Schaub, H., "Autonomous On-orbit Optical Navigation Techniques For Robust Pose-Estimation," *Advances in the Astronautical Sciences AAS Guidance, Navigation, and Control*, Vol. 172, Univelt Inc., 2020.

[37] Sharma, S., Beierle, C., and D'Amico, S., "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," *2018 IEEE Aerospace Conference*, 2018, pp. 1–12. doi:10.1109/AERO.2018.8396425.

[38] Fuchs, T. J., Thompson, D. R., Bue, B. D., Castillo-Rogez, J., Chien, S. A., Gharibian, D., and Wagstaff, K. L., "Enhanced flyby science with onboard computer vision: Tracking and surface feature detection at small bodies," *Journal of Earth and Space Science*, Vol. 2, No. 10, 2015, pp. 417–434. doi:10.1002/2014ea000042.

[39] Furfaro, R., and Law, A. M., "Relative optical navigation around small bodies via extreme learning machines," *AAS/AIAA Astrodynamics Specialist Conference 2015*, Vol. 156, Univelt Inc., 2016, pp. 1959–1978.

[40] Gaudet, B., Linares, R., and Furfaro, R., "Deep reinforcement learning for six degree-of-freedom planetary landing," *Advances in Space Research*, Vol. 65, No. 7, 2020, pp. 1723–1741. doi:10.1016/j.asr.2019.12.030.

[41] Scorsoglio, A., D'Ambrosio, A., Ghilardi, L., Gaudet, B., Curti, F., and Furfaro, R., "Image-Based Deep Reinforcement Meta-Learning for Autonomous Lunar Landing," *Journal of Spacecraft and Rockets*, 2021, pp. 1–13. doi:10.2514/1.a35072.

[42] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62–66. doi:10.1109/TSMC.1979.4310076.

[43] Lundberg, S. M., and Lee, S.-I., "A unified approach to interpreting model predictions," *Proceedings of the 31st international conference on neural information processing systems*, 2017, pp. 4768–4777.

Appendix - A

In this appendix the 3 function that generates the PA, Magnitude and Orientation estimate are detailed.

Phase angle function

The output of the $PA(x)$ function is in deg, the input x is adimensional and is object eccentricity.

$$PA(x) = p_2 x^2 + p_1 x + p_0 \tag{9}$$

$p_0 = 10.31$; $p_1 = 12.42$; $p_2 = 92.15$;

Magnitude function

The output of the $M(x, y)$ function is in pxl, the input x and y are respectively in deg and pxl and are the phase angle and the major-axis length

$$M(x, y) = \sum_{\substack{i=0,\cdots,5 \\ j=0,\cdots,5 \\ i \cdot j \leq 6}} p_{ij} x^i y^j \tag{10}$$

$p_{00} = 120.8$; $p_{10} = -7.647$; $p_{01} = -1.013$; $p_{20} = -144.9$; $p_{11} = 0.3552$; $p_{02} = 0.003179$; $p_{30} = 144.2$; $p_{21} = 0.3701$; $p_{12} = -0.001161$; $p_{03} = -4.667e-06$; $p_{40} = -59.36$; $p_{31} = -0.2142$; $p_{22} = 4.841e-05$; $p_{13} = 1.133e-06$; $p_{04} = 3.292e-09$; $p_{50} = 11.18$; $p_{41} = 0.02029$; $p_{32} = 8.412e-05$; $p_{23} = -1.278e-07$; $p_{14} = -3.471e-10$; $p_{05} = -9.042e-13$;

Orientation function

The output of the $\Phi(x)$ function is in deg, the input x is also in deg and represent an orientation angle between the CoB and the eCoB.

$$\Phi(x) = a_1 sin(b_1 x + c_1) + a_2 sin(b_2 x + c_2) + a_3 sin(b_3 x + c_3) + a_4 sin(b_4 x + c_4) \tag{11}$$

$a_1 = 553.6$; $b_1 = 0.008099$; $c_1 = -0.5278$; $a_2 = 313.3$; $b_2 = 0.0112$; $c_2 = 2.154$; $a_3 = 9.686$; $b_3 = 0.06434$; $c_3 = 1.644$; $a_4 = 16.5$; $b_4 = 0.03558$; $c_4 = 2.855$;

Appendix - B

The NN and CNN architectures used in this work are reported in this section using the Tensorflow 2.5.0 standard together with the training and validation performance.

Table 4 Architecture of the NN_1 considered in this work. The total number of parameters is $3,139$, all of which are trainable.

Layer (type)	Output Shape	Param #
input (InputLayer)	(None, 14)	0
dense_1 (Dense)	(None, 64)	960
drop (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dense_3 (Dense)	(None, 3)	99

Table 5 Architecture of the NN_2 considered in this work. The total number of parameters is $21,347$, all of which are trainable.

Layer (type)	Output Shape	Param #
input (InputLayer)	(None, 14)	0
dense_1 (Dense)	(None, 32)	480
dense_2 (Dense)	(None, 64)	2112
dense_3 (Dense)	(None, 128)	8320
drop (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8256
dense_5 (Dense)	(None, 32)	2080
dense_6 (Dense)	(None, 3)	99

Table 6 Architecture of the NN_3 considered in this work. The total number of parameters is $1,002,947$, all of which are trainable.

Layer (type)	Output Shape	Param #
input (InputLayer)	(None, 14)	0
dense_1 (Dense)	(None, 64)	960
dense_2 (Dense)	(None, 128)	8320
dense_3 (Dense)	(None, 256)	33024
dense_4 (Dense)	(None, 512)	131584
dense_5 (Dense)	(None, 1024)	525312
drop (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 256)	262400
dense_7 (Dense)	(None, 128)	32896
dense_8 (Dense)	(None, 64)	8256
dense_9 (Dense)	(None, 3)	195

Table 7 Architecture of the CNN considered in this work. The total number of parameters is $1,438,659$, all of which are trainable.

Layer (type)	Output Shape	Param #
input (InputLayer)	(None, 128, 128, 1)	0
conv_1 (Conv2D)	(None, 128, 128, 32)	320
pool_1 (MaxPooling2D)	(None, 64, 64, 32)	0
conv_2 (Conv2D)	(None, 64, 64, 64)	18496
pool_2 (MaxPooling2D)	(None, 32, 32, 64)	0
conv_3 (Conv2D)	(None, 32, 32, 128)	73856
pool_3 (MaxPooling2D)	(None, 16, 16, 128)	0
conv_4 (Conv2D)	(None, 16, 16, 256)	295168
pool_4 (MaxPooling2D)	(None, 12, 12, 256)	0
flat (Flattern)	(None, 16384)	0
dense_1 (Dense)	(None, 64)	1048640
drop (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dense_3 (Dense)	(None, 3)	99

Training and validation

In Figure 24 the Mean Squared Error is reported for NN_1, NN_2, NN_3, and CNN together with the smallest validation loss achieved during training (green point). These occur respectively at 459, 476, 358, and 53 epochs. To address overfitting, the sets of weights and biases of the networks at these epochs are the ones used in inference on the test sets.

Table 8 Training hyper-parameters for the NN and CNN.

Hyper-parameters	NN_1	NN_2	NN_3	CNN
Batch size	32	32	32	32
Optimizer	Adam	Adam	Adam	Adam
Dropout	0.05	0.02	0.01	0.05
Activation function	ReLu/Sigmoid	ReLu/Sigmoid	ReLu/Sigmoid	ReLu/Sigmoid
Error metric	MSE	MSE	MSE	MSE
Epochs	500	500	500	60

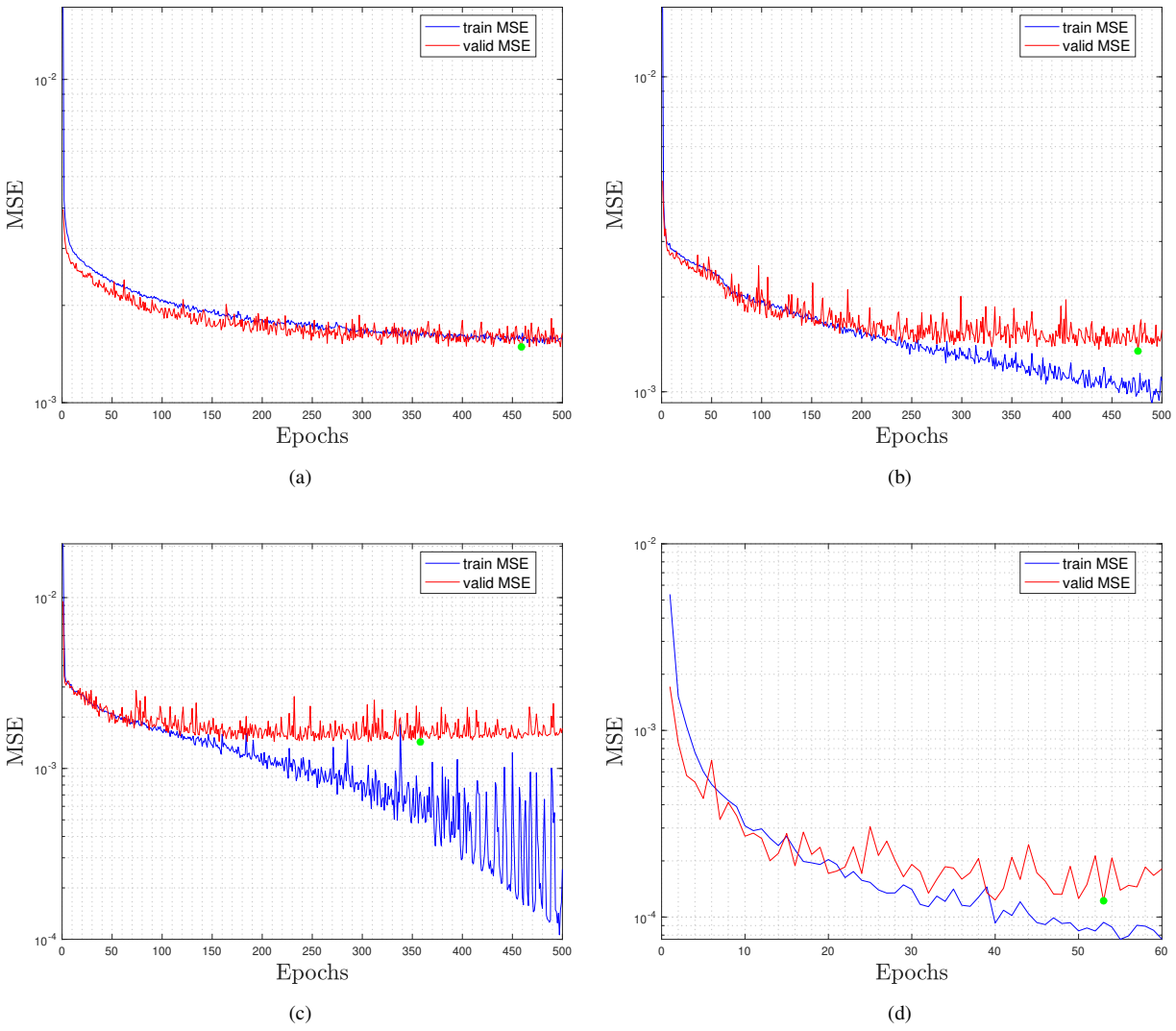


Figure 24 Training (blue) and Validation (red) history of the Mean Squared Error metric for NN_1 (a), NN_2 (b), NN_3 (c), CNN (d).