BENNUNET - APPLYING MACHINE LEARNING TECHNIQUES FOR AUTONOMOUS OPTICAL RELATIVE NAVIGATION OF AN ASTEROID

Alfredo Escalante López[(1)], **Pablo Ghiglino**[(2)], **Manuel Sanjurjo-Rivo**[(1)]

[(1)] *Universidad Carlos III de Madrid, 28911 Leganes, Madrid, Spain*
[(2)] *Klepsydra Technologies AG, Brugglenstrasse 2a, 8604 Volketswil, Zurich, Switzerland*

ABSTRACT

This contribution presents Bennunet, a hybrid neural network-based method, devoted to on-board spacecraft relative position and attitude estimation in the vicinity of minor bodies using monocular vision. It is a follow-up investigation of Churinet, which set up the basis for using neural networks for pose estimation, offering a lightweight and robust solution. In this case the asteroid Bennu has been chosen as the target of the investigation given the extensive data derived from the Osiris-Rex mission. Multiple shape models of Bennu have been used to produce synthetic image training sets covering the whole range of camera position, attitude, illumination conditions, camera field-of-view, image resolution, and target albedo map variation, allowing to study the impact of different geometries and image effects in the network performance. Modern state-of-the-art architectures have been implemented for Bennunet, substantially improving its performance compared to the base Convolutional Neural Network (CNN) used in previous works. Multiple data augmentation techniques have been implemented to further extend the image sets during training. Finally, the trained networks have been tested with real images of Bennu, achieving to maintain the same accuracy as with synthetic images without any degradation.

1 INTRODUCTION

For decades, Small Solar System bodies have been the target of space missions. The low temperature and low gravity environment existing on comets and asteroids preserve its high volatile content, allowing scientist to investigate the origins of the Solar System. The need of taking in-situ measurements, has driven the exploration of various comets and asteroids in the last decades, the Vega 1 and 2 spacecrafts intercepted Comet Halley in March 1986 [1]; Rosetta International Mission which was the first to rendezvous with a comet, 67P/Churyumov-Gerasimenko, and to land on it [9] after performing fly-bys to asteroids Steins and Lutetia [5]; missions Hayabusa 1 and 2 explored asteroids Itokawa and Ryugu respectively, bringing samples from Ryugu back to the Earth. In order to operate the spacecraft in the vicinity of such small bodies, a combination of radio navigation ground support and autonomous relative optical navigation involving the use of navigation cameras is generally used. The main challenge of navigating around bodies with very small size and mass is that the ephemeris and physical properties of the target are typically not known with enough accuracy for using standard orbit determination techniques[4]. Due to the inaccuracies derived from ground observations of these objects, the only solution then is to rely on in-situ measurements for determining with the onboard computer the relative position of the spacecraft with respect to the target. Although other sensors as thermal cameras or LiDARs have also been used for navigation, monocular vision cameras are usually the main sensor used for performing optical relative navigation. These cameras can be used to

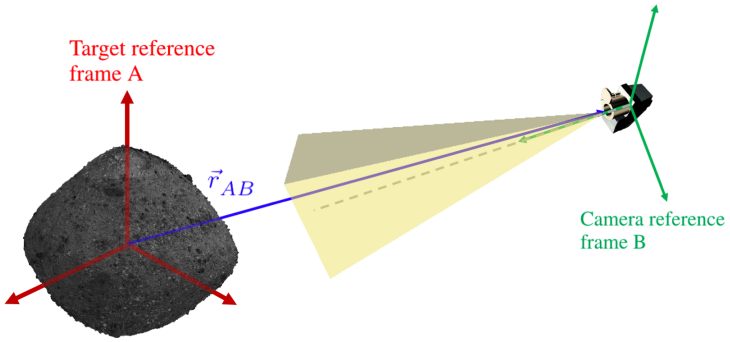

Figure 1: Depiction of camera pose estimation with respect to target body fixed frame.

estimate the relative position of the spacecraft with lower hardware complexity, mass, size, and power requirements.

The state-of-the-art monocular pose determination methods for optical relative navigation rely on identifying a pre-defined subset of landmarks in the image with classical image processing algorithms and comparing its location with all the known landmarks database [6]. For the Rosetta mission, automatic landmark tracking was applied using the image database, the landmark coordinates and the preliminary shape model as input parameters, with the major disadvantage that the image had to be downlinked to Earth for manual visual inspection and orbit determination. Moreover, the robustness of this method is strongly dependant on the illumination conditions and its accuracy degrades at very low and very high phase angles.

The usage of Convolutional Neural Networks (CNNs) is spreading in many industries as the main computer vision solution due to its precision, robustness, efficient performance in changing scenarios and lightweight. Instead of relaying on features matching as landmark-based methods, machine learning algorithms could be trained to learn the nonlinear transformation from the 2-D input image space (for grayscale) to the 6-D pose vector space (3 position coordinates plus 3 Euler angles). The main drawback of implementing this direct nonlinear transformation, is that the Neural Network behaves like a black box, so they should be extensively tested and validated in order to identify possible failure cases, otherwise very difficult to detect. The pose vector can be estimated as a discrete or continouos variable using two methods. The first, called multiclass-classification, consists on solving a classification problem after discretizing and labelling the pose space [12], [10], consequently the maximum achievable accuracy depends on the resolution of the discretization. The second method applies regression such that the output of the Neural Network is directly the pose vector coordinates [8], [16].

In this work, a set of Convolutional Neural Network (CNN) organised in two levels, high-level multiclass-classification and low-level regression, is presented, capable of estimating the relative pose of a camera with respect to a target body. The pose estimation problem is shown in Fig. 1, where $\vec{r}_{AB}$ represents the position vector of the camera focal point with respect to the asteroid centered body-fixed frame to be estimated by the CNNs. This approach was first used for the case of comet 67P/Churyumov-Gerasimenko with promising results. Data augmentation techniques applied during training have been investigated in order to generalise the CNNs estimating capabilities as much as possible, accounting for image shift, rotation, and distortions, while maintaining a reasonable accuracy in the pose estimation. In this case, Bennu has been selected as the main target to investigate the applicability of CNNs for pose estimation using data derived from the Osiris-Rex mission. The OCAMS instrument onboard Osiris-Rex counted with three cameras with different field-of-views,

Figure 2: Comparison of different Bennu models at different spatial resolutions.

which allows to investigate the performance of the CNN at different altitudes and with different camera configurations. In addition, the multiple spatial resolution shape models available for Bennu can be used to evaluate the impact of the model resolution used to generate the training sets.
As it is well known, training CNNs requires a large amount of data, and even the large archive of the Osiris-Rex mission is not enough to directly train the neural networks. For this purpose, a Python package named SPyRender, developed in previous works, has been used to generate large sets of synthetic images suitable for training the CNNs. The geometry of the target and observer, illumination source, camera model, and target shape and texture can be configured and modified during runtime when rendering the scene, allowing for the efficient production of different combinations of geometric conditions. These sets of random combinations of the elements defining the scene together with additional data augmentation methods, are the key to maintain the CNN accuracy when providing as input real images which are not included in the training sets.

2 SYNTHETIC IMAGES GENERATION METHODS

In order to have large enough training sets and aiming to study the impact of the different geometric configurations of the scenario on the CNN accuracy, multiple synthetic image data sets have been generated. Different image effects have been implemented via configuration like illumination source position, type and intensity, camera and target position and rotation, camera field-of-view aperture angles, image resolution, and textures of the target model. For the study case of asteroid Bennu, multiple models are available at the Osiris-Rex SPICE archive PDS4 collection according to model coverage, production technique and spatial resolution. In this work, two shape models have been used to evaluate the effect of the spatial resolution of the input model and scale of the surface features in the CNN accuracy. Fig. 2 shows the difference between different 3D models used for generating the synthetic images. From left to right, the preliminary model based on PolyCam images during approach phase at a spatial resolution of 6 meters, the more detailed model after close proximity operations with a spatial resolution of 880mm, and the same detailed model but adding albedo map to the shape model. The texture used for the albedo map is based on the normalized one derived from the PolyCam images at low phase angles (less than 8 degrees) with a spatial resolution of approximately 6 cm and available at the USGS [17]. Because this normalized map does not have coverage out of the $\pm$55 degrees latitude, the texture map has been completed with the normalized global mosaic produced with images at phase angles up to 30 degrees. The resulting albedo map is shown in Fig. 3. The similarity between the synthetic and real images can be appreciated in Fig. 4, displaying some examples of synthetic MapCam images generated with SPyRender (first row) compared to the corresponding real images (second row).
For rendering the images, the aperture of the field of view (FoV) of the camera is defined in terms of reference and cross angles for a rectangular shape following the SPICE format. These angles have been taken from the OCAMS Instrument Kernel [15]. For MapCam, an angle of 3.97 degrees is used

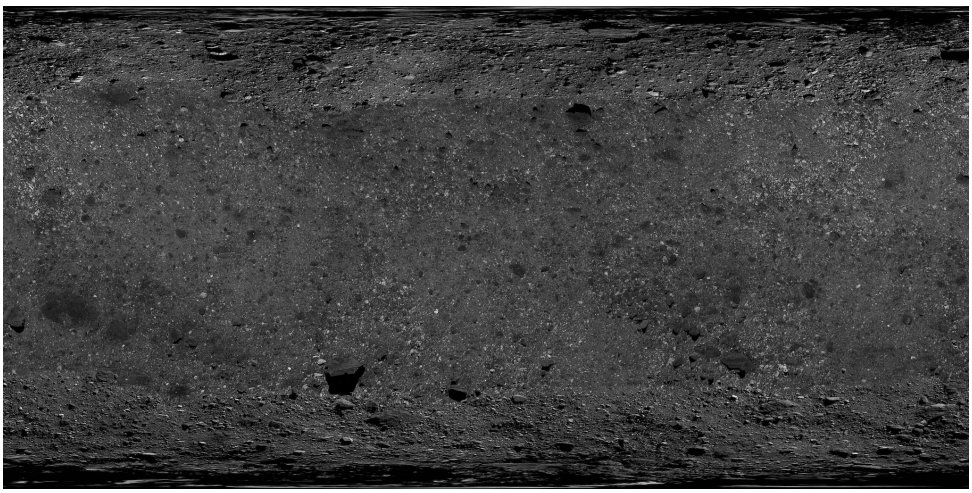

Figure 3: Albedo map for the surface of asteroid Bennu.

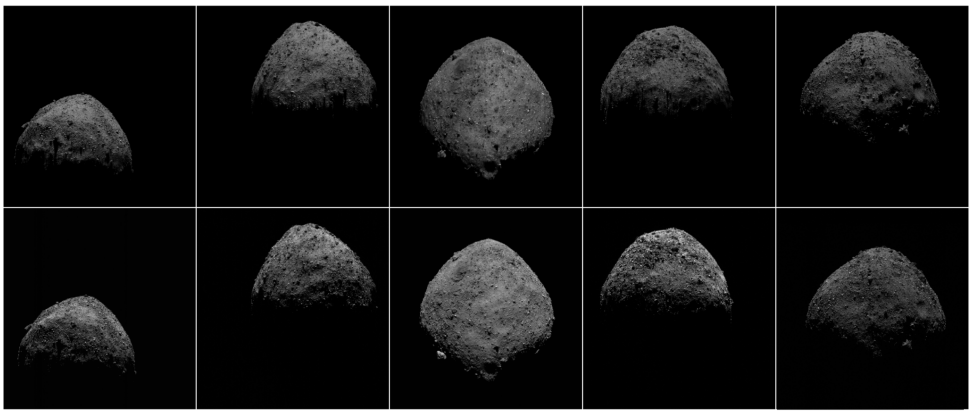

Figure 4: Comparison of real vs synthetic OCAMS MapCam images.

for a squared FoV, while for SamCam, a larger angle of 20.44 degrees is used. This difference in the FoV aperture allows to produce images with the full asteroid in the camera FoV when being far from the asteroid (using MapCam) or when closer to its surface (using SamCam), enabling to train and evaluate the capability of the CNN to achieve field-of-view invariance. Regarding the resolution in terms of pixel lines and pixels samples of .png images output by the offscreen renderer, 224x224 and 480x480 have been used. 224x224 is a standard resolution used in computer-vision CNN but in later works, the usage of larger resolutions has shown to improve the accuracy of the CNN depending on the use case. Because very high resolution models are available in this case, it is expected than increasing the image resolution could indeed improve the CNN accuracy.

Two main types of image data sets have been produced to evaluate the performance of both classification and regression CNNs. The first group of sets contains images generated from the whole range of camera positions around the target body and is mainly devoted to analyse the training of CNNs capable of global position estimation. The second group consists of regional sets of images containing images of one single sector of the target body. For the latter group, the space around the target body has been divided in sectors of 45 degrees longitude and latitude yielding a total of 32 sectors. For each of these sectors, regional train and test data sets have been produced for analysing the training of CNNs which could estimate position and attitude of the camera with higher accuracy.

Data augmentation techniques are applied to the data sets, either during production or directly during

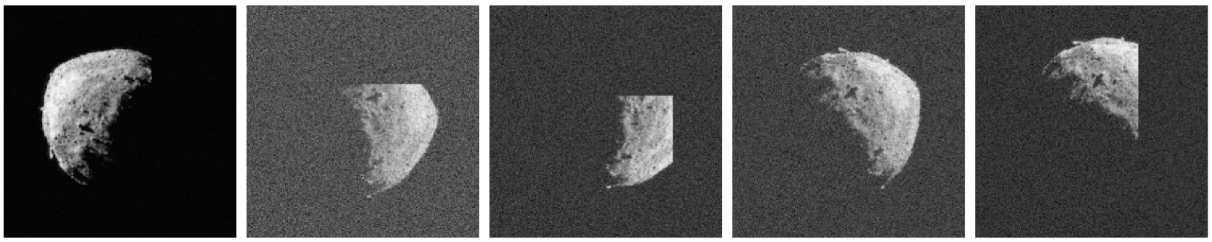

Figure 5: Example of data augmentation combined effects on a single image.

the training process, in order to extend the features of the images used for training, seeking rotational invariance, translational invariance and noise invariance. In addition to these basic modifications, a type of cutout erase technique has been applied. This method randomly removes up to half of an image from the side to the center of the image for either one or two continuous sides. This allows the CNN to interpret images in which a large part of the target was outside of the FoV. The effect of applying these data augmentation techniques on a single image can be appreciated in Fig. 5. In addition, a set of real MapCam and SamCam images obtained from the Osiris-Rex archive have been compiled to validate the trained models and compare the performance when ingesting real images instead of synthetic ones. While the different camera orientations can be extended during training by rotating and applying a shift to the images (as deviating from nadir pointing), illumination conditions have to be generalised during the generation of the data sets. As concluded in previous works, illumination conditions play a key role in computer vision and optical navigation, so it should be properly configured to achieve illumination invariance in terms of intensity and direction. It is true that depending on the mission, the illumination conditions could be constrained to a given range. For example, the spin axis of Bennu as well as its orbit angular momentum are quasi-perpendicular to the ecliptic plane, so the Sun position would be almost restricted to the XY plane in the body-fixed frame. However, for the sake of assessing a general solution, this fact is not taken into account in this work.

2.1 Training Sets Generation

The produced image data sets intended for global pose estimation are listed in Table 1, including the main features of the data sets. Each data set is composed by 40000 images, and its corresponding label files, for which 80% correspond to train set and 20% to validation set. The first and most simple set "simTrain224BE_sr" covers the whole range of camera positions around the target, with attitude fixed to Nadir pointing, meaning the target is centered in the images and there is no image rotation around the boresight direction (pixel lines aligned with asteroid North). In addition, random illumination source direction has been introduced during image rendering. For the rest of the image sets, the different combinations of around boresight rotation, target shift, illumination intensity, Gaussian noise, and shift cutout erase, have been applied. The same geometric conditions have been used to create the equivalent sets but using variable FoVs associated to MapCam, SamCam and 2xSamCam (an artifical one double the size of SamCam), and altitude ranges: 6 to 20 kilometres for MapCam, 1 to 3 kilometres for SamCam, and 500 to 1000 metres for 2xSamCam. Similarly, two sets have been created for each of the previous combinations, at 224x224 and 480x480 image resolution.

3 CONVOLUTIONAL NEURAL NETWORK ARCHITECTURES AND TRAINING

As an starting point to evaluate the impact of the different geometric combinations on the CNN performance a simple architecture based on AlexNet was selected [7], which has been proven to

Table 1: Description of the global synthetic image datasets generated for this work

Dataset	Description	Images
simTrain224BE_sr	Centered, No boresight rotation, Fixed brightness	40000
simTrain224BE_sr_rr	Centered, boresight rotation [0, 360], Fixed brightness	40000
simTrain224BE_sr_br	Centered, No boresight rotation, brightness [-98%, 260%]	40000
simTrain224BE_sr_rr_br	Centered, boresight rotation [0, 360], brightness [-98%, 260%]	40000
simTrain224BE_sr_rr_br_o10	shift, boresight rotation [0, 360], brightness [-98%, 260%]	40000
simTrain224BE_sr_rr_br_o10_ng	shift, boresight rotation [0, 360], brightness [-98%, 260%], Gaussian noise	40000
simTrain224BE_sr_rr_br_o10_se_ng	shift, boresight rotation [0, 360], brightness [-98%, 260%], shift eraser 50%, Gaussian noise	40000
simTrain224BE_mapcam	Real OCAMS MapCam images	40000

perform adequately in similar neural-network based applications for relative navigation like non-cooperative spacecraft rendezvous [14] or asteroid centroiding for autonomous attitude navigation [19]. In this case the selected architecture consists on just two convolutional layers followed by three fully-connected layers, the last one being the output layer. With such a simple architecture, it is easier to have convergence during training and evaluate the relation between the architecture hyperparameters and the performance for each study case. Once the base architecture CNN has been trained against the multiple image sets presented in Table 1 and the trends in the training process have been identified, it was decided to test modern state-of-the-art architectures for the most relevant cases. These modern architectures require longer training times, more memory for the same batch sizes due to the larger number of coefficients, and convergence may not be reached, however, they shown a great improvement in accuracy for other computer vision tasks. In this work, VGG-19, DenseNet and ResNet50v2 architectures have been tested for some of the training sets achieving great performance in the position estimation compared to the base architecture.

In this work, two types of CNNs (independent of the core architecture) have been produced differing on the last layer. The first type are Multiclass-classification CNNs, for which the last fully-connected layer has dimension 32 (same dimension as the labels vector for the camera position space sectors, each sector covering 45 degrees longitude and latitude). The Softmax [2] activation function is applied to the last layer as it normalizes the last layer output vector into a probability distribution over sector labels, meaning the element in the function output with the highest probability represents the estimated sector. In (1), z_i refers to the i-th element of the vector provided as output by the last fully-connected layer, and L_i are the elements of the resulting probability distribution.

$$L_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{1}$$

The second type are Regression CNNs, for which the last fully-connected layer implements the linear activation function providing directly as output the target variable values to be estimated, having dimension 3 for estimating camera position vector in Cartesian coordinates, 2 for Yaw and Pitch angles, and 1 for Roll (around boresight direction) angle. The decision to have three different Regression CNNs for estimating position vector, pitch and yaw angles, and roll angle is supported by the difference in scale between the output variables and the impact this has on the weights back-propagation and estimation accuracy. A spread range of values in the target variables causes weights to abruptly change, introducing instabilities in the training process [3]. This behaviour could be overcome by having a multi-branch network to control the different scale of the variables, each of them provided by a different branch of the same CNN, however, it was decided to only use sequential CNNs at this stage of the development.

In the training process, the most relevant parameters to configure are the Loss function, the optimization algorithm, and the training epochs. The error or loss function, is used to estimate the loss of the model at the current iteration of the optimization algorithm, such that the weights are updated

accordingly to reduce the defined loss at the next iteration. The chosen loss function depends on the neural network to be trained and the predictive problem for which it will be applied. For Multiclass-classification CNNs, the Sparse Categorical Cross-Entropy loss function has been selected as it has been proven competitive in most domains and the preferred default option [11]. The expression for the Categorical Cross-Entropy is shown in (2), where y_i represents the target value, and ${y'}_i$ represents the i-th element in the model output.

$$L = -\sum y_i \log({y'}_i) \tag{2}$$

For the Regression CNNs, the typical loss functions to be used are the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). Because the roll, pitch and yaw angles can be considered zero mean Gaussian distributions, MSE has been used as the loss function for the CNNs estimating these quantities. As the MSE penalizes larger mistakes in the estimation, it is more likely that the weights are updated so to avoid producing outliers as output [13]. In addition, a custom loss function has been defined implementing the Mean Translation Error between the ground truth and the estimated position for a given image. Using the MAE or MSE for position estimation may result on one or two coordinates of the 3-dimensional position vector having a very small error while the other shows a large deviation. On the other hand, using the translation error, the magnitude of the position error vector is minimized. The equations for MAE, MSE and Translation Error are shown in (3), (4) and (5) for n samples, where y_i is the ground truth and y'_i the predicted value.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - y'_i| \tag{3}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - y'_i)^2 \tag{4}$$

$$E_T = \frac{1}{n}\sum_{i=1}^{n} |\vec{y}_i - \vec{y'}_i| \tag{5}$$

For the optimizer, SGD (Stochastic Gradient Descent with momentum) has been selected as optimization algorithm in charge of minimizing the loss function. While it may have slower convergence compared to Adam (Adaptive Moment), a better converged result has been achieved with SGD. The only drawback of SGD compared to Adam is that before training, both the input (images) and output (pose vector) variables have to be scaled to the 0 to 1 range. The training epochs (number of passes through the whole train set) have been set to ensure that validation loss does not change if epochs are increased further.

3.1 Time-Distributed Neural Networks

Recent computer vision algorithms for autonomous navigation implementing Neural Networks use as input, sequences of images instead of a single image. This approach enables the CNN to learn the dynamic behaviour of the specific use case improving the accuracy of the estimation by using accumulated input data, and even opening the door to directly estimating time derived quantities like linear and angular velocities. Time-Distributed Neural Networks or TdCNNs are one type of CNNs which allows the ingestion of sequencies of images by adding a Time-distributed layer like GRU (Gate Recurrent Units) in the architecture. In this architecture, each frame of the input image sequence is provided to the base CNN (note that in this case the same weights are used for the CNN applied to each image), the output of the CNN for each image is then combined in the GRU layer. Finally, a decision network consisting on several fully-connected layers is stacked on top of the GRU to provide

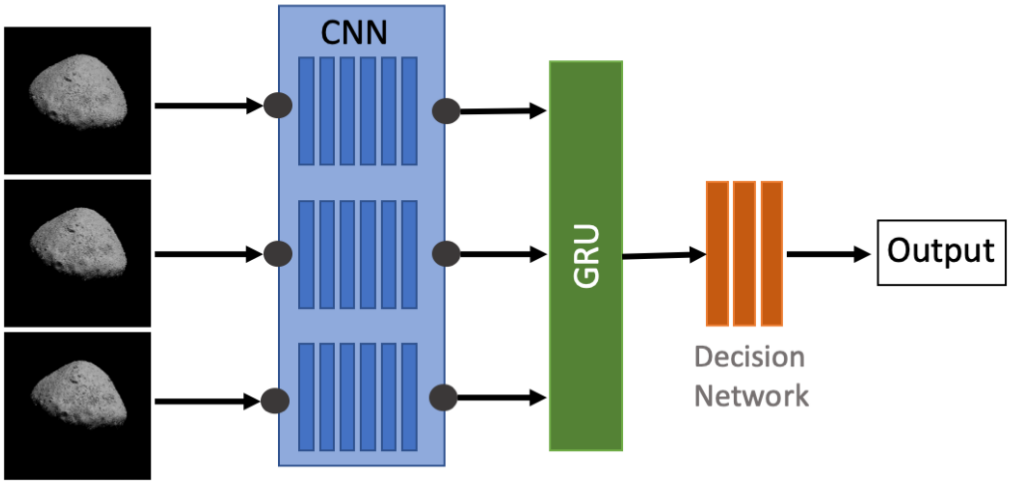

Figure 6: Time Distributed Convolutional Neural Network (TdCNN) architecture diagram.

the final output of the neural network, either space sector for multiclass classification or pose vector for regression. The global architecture for the TdCNN is depicted in Fig. 6. With the objective of testing this approach for the relative pose estimation, a dedicated training set consisting of sequences of 4 images has been produced. For each sequence, the geometry of the first frame is computed as for the base training sets, and then a random perturbation of the lattitude, longitude, altitude and euler angles is iterated to produce the remaining images in the sequence. This means that in order to get training sets of the same size as for the base CNNs, the training set for the TdCNN are four times larger, making the training process substantially more computationally inttensive.

3.2 Hybrid Neural Network Solution

The main drawback of using Multiclass-classification CNNs for position estimation is that the maximum accuracy of the estimated output directly depends on the number of sectors or labels in which the 3D space has been discretized. Nevertheless, as the number of sectors increases, so does the instabilities in the training process and the accuracy in the sector estimation. The main reason for this is that Classification problems take the different possible values of the output variable as independent discrete values without accounting for ordering or underlying continuous relations between them. In general, continuous variables such as the camera position or the Euler angles are better estimated by regression CNNs. However, after trying to train for a global position regression solution, the optimizer was not able to successfully minimize the loss function, most likely due to the large non-linearities in the underlying transformation for the input image to the 6D pose vector space. This lead to the hybrid two-level architecture, consisting of a High-level Multiclass-classification CNN in charge of estimating the local region or sector of the 3D space, and a set of Low-level Regression CNNs, each trained for one specific sector and capable of accurately computing camera position in Cartesian coordinates and camera angles. This approach took advantage of the strengths of both methods interconnecting both types of CNNs. Fig. 7 depicts the flow of this two-levels global approach for the pose estimation. In addition, a pre-processing step was added to perform a de-shifting operation to the input image centering the target in the image. This step is applied to improve the accuracy of the pose estimation, which in previous works was found to be strongly impacted by the position of the target centroid with respect to the center of the image.

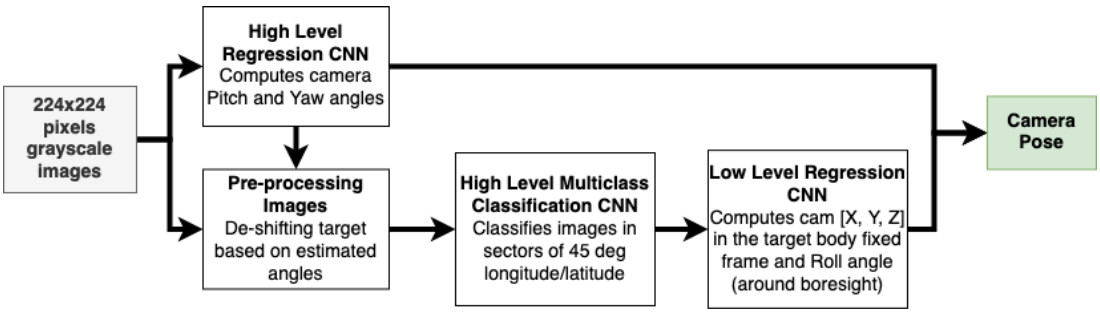

Figure 7: Two levels neural network flowchart.

3.3 Size of Trained Neural Networks

An important aspect of the neural networks is its lightweight and efficiency, characteristics that make them suitable to be executed onboard for autonomous navigation. Regarding the disk usage of the neural network, it is directly related to the size or number of coefficients composing the network. In general, the number of coefficients increases with the wider layers and deeper architectures. The base CNN architecture studied in this work consisting on two convolution and two fully-connected layers has 30 million parameters, resulting on a disk space of 120MB. On the other hand, the deeper VGG-19 architecture trained for the same image set has 20 million parameters, resulting on just 80MB of disk space. Similarly, other architectures as ResNet and DenseNet have less coefficients, yielding a substantial reduction in the disk space. Independent of the chosen architecture, other techniques can be applied to reduce the size of the neural network. The most common one is dynamic range quantization. This method statically quantizes only the weights from floating point to integer at conversion time, which provides 8-bits of precision and reduces the size of the network to approximately 1/4 of the original size. This technique has been applied to the trained models with no significant impact on accuracy, meaning the size of the CNN can be as small as 20MB (even less for efficiency-oriented architectures like MobileNet). Considering 32 CNNs are required for a global pose estimation solution (one per sector of the discretized space), the maximum size required would be just 640MB. The global hybrid model has been tested in a ARM64 Dual Core with 8GB RAM getting a data rate of 12 frames per second and a CPU usage of 62%.

4 RESULTS

4.1 High-Level Regression

The pitch and yaw angles of the camera reference frame are computed by the High-level Shift Regression CNN by estimating the target centroid displacement in pixels with respect to the center of the image. Knowing the model of the camera field-of-view, the pixels are easily converted to angles. This pixel shift is also used to add a de-shifting pre-processing and center the target in the image that will be provided to the pose estimation CNNs. Therefore, the parameter to be estimated by this CNN results in a 2-components vector containing vertical and horizontal shift in pixels, meaning the last fully-connected layer of this CNN has dimension 2. In order to train this CNN, starting from nadir pointing images, a random target shift up to 120 pixels (slightly more than half of the image) has been introduced in order to produce the un-centered images. As introduced in previous sections, data augmentation techniques were used when producing this image set in order to improve the generalization of the shift estimation. These are, random image rotation, random illumination intensity, Gaussian noise, variable field-of-view, and variable asteroid shape models. The advantage of using a CNN for

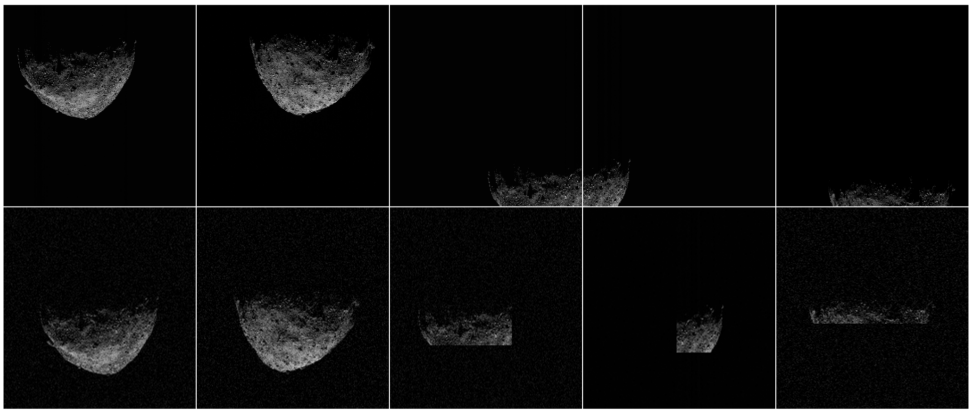

Figure 8: Results of de-shifting real OCAMS MapCam images.

estimating the shift, is that it can be taught to rely on asteroid features independent of shadow length (when illumination direction changes) and independent to small scale changes between multiple asteroid shape models (or due to activity of the surface), therein estimating the actual geometrical center of the target in the image instead of the illumination center.
For this de-shifting CNN, Mean Squared Error (MSE) has been used as the loss function to be minimized during training. However, for results visualization the Mean Absolute Error (MAE) has been plotted in Fig. 9 for a more straightforward reading of the CNN performance evolution during training. Note that because the output of the CNN has dimension 2 (horizontal and vertical shift), the loss is computed as the average of the MAE for the two output variables. It can be observed that while the train loss (dashed line) keeps decreasing with the number of epochs at a higher rare, the validation loss decreases much slower to a higher MAE of 12 pixels. For a 224x224 pixels image with a field-of-view of 3.97 degrees as MapCam, this pixel error is equivalent to an approximate pitch/yaw angle error of 0.21 degrees. The trained high-level regression CNN has been tested with real images from MapCam captured during the preliminary survey and consecutive orbit phases. For most of these images, the camera boresight was not aligned with the nadir direction, so the asteroid is in general not centered in the image. Moreover, the combined effect of illumination conditions and nadir off-pointing results in extreme cases with just a small illuminated part of the asteroid being visible but for which the CNN can still accurately estimate the centroid shift. Fig. 8 shows some examples of real images obtained from the OCAMS PDS Bundle for the specified time period, compared with the corresponding de-shifted images based on the CNN estimated shift.

4.2 High-Level Multi-class classification

In previous works, the impact of the different geometric effects on the sector estimation performance was carefully investigated [18] in order to achieve the best performance possible for a real mission scenario. In this contribution it was decided to explore the usage of the Time Distributed CNNs (TdCNNs) to improve further the sector estimation achieved with the base CNN. The Loss and Accuracy evolution of the classification during the training process is shown in Fig. 10 for both the base CNN and the TdCNN using the same architecture connected to the GRU layer. It can be appreciated that a boost in accuracy of about 20% has been achieved by using the TdCNN with sequences of 4 images. This promising result will be investigated more in detail in the future, implementing also advanced CNN architecture below the GRU layer to try to further improve the classification performance. In addition, it was observed that the classification error accumulates in the boundaries between sectors. This is because for variable illumination conditions, it is hard for the CNN to distinguish between

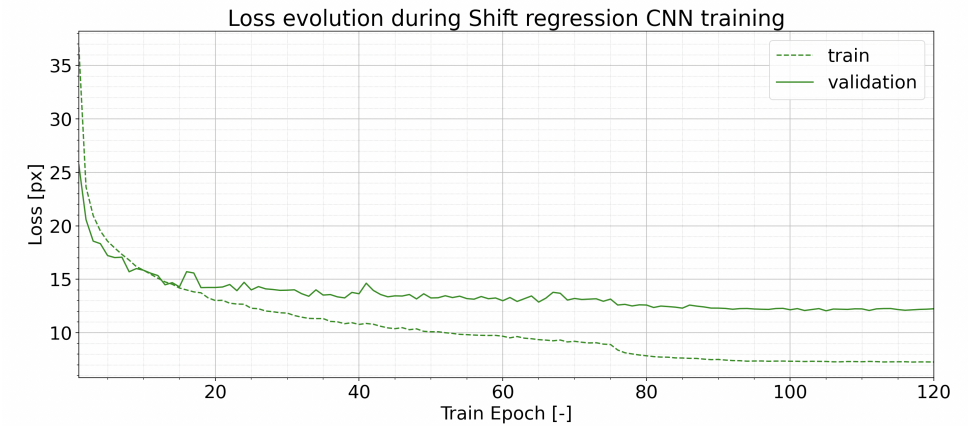

Figure 9: Train and Loss evolution for pixel shift regression.

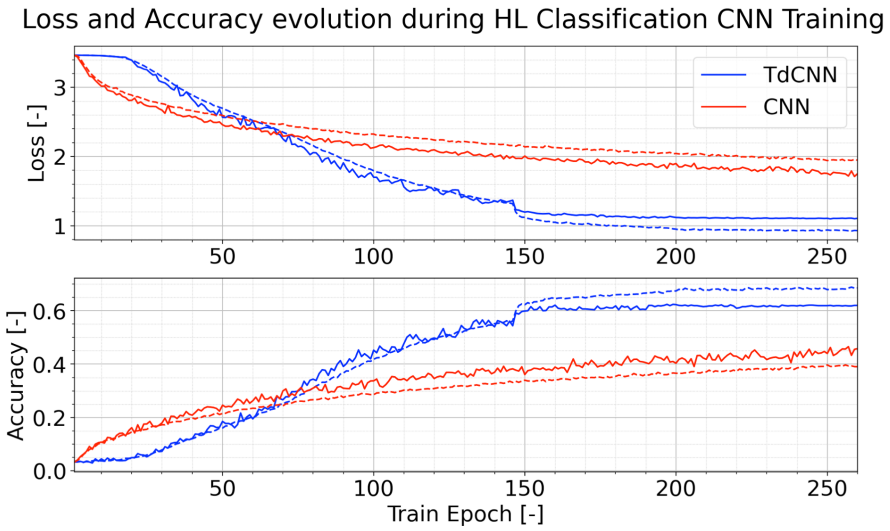

Figure 10: Train and Loss evolution for simple CNN and TdCNN for Multiclass-classification.

to neighbour sectors when the camera was over the border between them. To minimize the impact of this, the low-level regression CNNs are trained with images from extended sectors, meaning the discretized space sectors are extended by a 20% in order to overlap. This leads to CNN corresponding to neighbour sectors, both being able to accurately estimate the position vector close to the boundary between them.

4.3 Low-Level Regression

The low-level regression CNNs have been trained each for one specific sector of the 3D space and are capable of estimating camera position vector in Cartesian coordinates and roll angle (around camera boresight rotation). Note that two independent sequential networks have been trained, one for position estimation having last fully-connected layer of dimension 3, and other for roll angle estimation with dimension 1. In order to train these CNNs, one image set has been produced per sector. All the previously introduced data augmentation effects have been used for training this CNNs, including the shift erase cutout, which enables the CNN to deal with pre-processed de-shifted images. In the case of the position regression CNN, the used loss fuction to be minimized is the Translational Loss

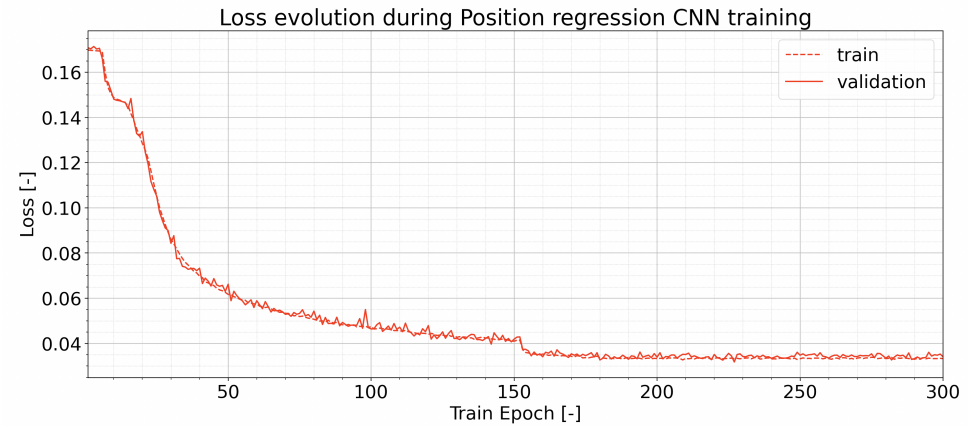

Figure 11: Train and Loss evolution for Position regression.

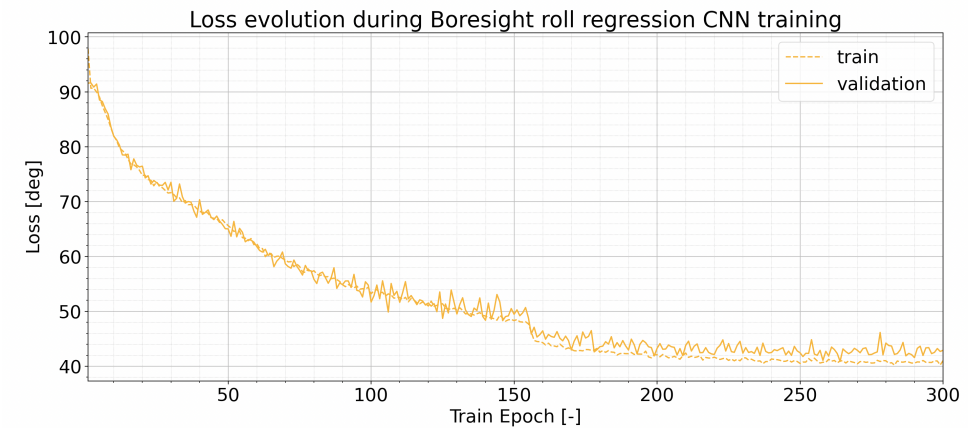

Figure 12: Train and Loss evolution for Boresight roll angle regression.

Error, compared to the MSE used for the other cases. The evolution of the loss function for position estimation during training of base CNN architecture is shown in Fig. 11. It can be appreciated that validation and training loss follow a similar evolution for most of the training epochs. It is also important to note the further decrease on the loss function after epoch 150, showing the importance of properly scheduling the learning rate, to achieve the minimum loss and keep improving loss even when convergence seemed to be reached. In the case of the roll angle evolution shown in Fig. 12, the validation loss is more noisy and converges to a higher value than the training loss. The hyperparameters for training the roll angle, like learning rate or dropout could be fine tuned to improve the validation loss.

In Fig. 13 the final value for the loss after training with the different training sets are summarised for the altitude range 500 to 1000 meters with the 2xSamCam field-of-view. As it can be observed, superimposing multiple geometric effects as noise, cutout erase and target shift (off-nadir) have similar resulting loss. It is actually reduced slightly due to the regularization introduced by the data augmentation and the reduced over-fitting. However, when introducing rotation around boresight (roll angle), the loss substantially increases. Therefore, if attitude of the spacecraft could be estimated from other sources like star trackers and the dynamics of the target are also known to some degree, the accuracy can be substantially improved by constraining the roll angle range. For example, a deviation of pixel lines direction of up to 50 degrees from the North direction, reduced approximately a 20 percent the loss with respect to the 360 degrees random rotation, and in many cases, at least the spin axis of the

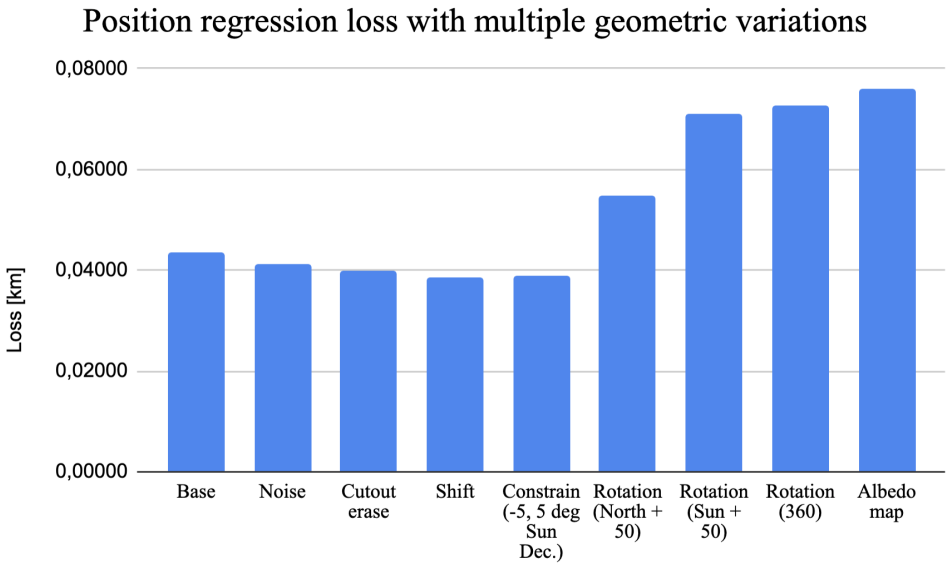

Figure 13: Effects on position estimation loss of multiple geometric conditions.

asteroid is well characterized. Knowing the spin axis direction and the spacecraft attitude with respect to the International Celestial Reference System, could potentially improve the accuracy of the CNN estimation. Regarding the introduction of an albedo map compared to the plain color shape model, the impact on loss is negligible, however, it is required to avoid degraded accuracy when providing real images for estimation, for which surface color is clearly influenced by the surface albedo over the surface incidence illumination angle.

Finally, the altitude-normalized loss resulting from training with multiple shape models and with multiple image resolution is shown in Fig. 14 for three altitude ranges 500 to 1000 metres, 1.4 to 3 kilometres, and 7 to 20 kilometres. When assessing the spatial resolution of the asteroid shape model, it is clear that the higher resolution (red) results in reduced loss compared to the low resolution model (blue) at any altitude. The more detailed surface features make easier for the CNN to located and estimate the position of the camera. Moreover, when using the higher resolution model, the normalized loss is reduced when moving farther from the asteroid as with a smaller field-of-view, the details on the asteroid limb gain more relevance on the image. Regarding the image resolution, the higher 480x480 resolution yields a substantial decrease of the loss compared to the standard 224x224 both for the low (yellow) and high (green) resolution models. Last, the result of training a more complex architecture like VGG-19 with the high resolution model and the standard image resolution of 224x224 is shown (orange), decreasing the loss to less than a 50% of that of the base architecture. This trained model has been tested with real images from MapCam showing no sort of degradation of accuracy compared with the synthetic images used for training, meaning that the gap between synthetic and current images has been successfully overcame.

5 CONCLUSION

In this contribution, the training of CNNs applied to monocular vision navigation has been extended, assessing variations in target model and camera parameters which were not accounted in previous works. The main result is the successful testing of real images without any accuracy degradation, using CNNs trained only with synthetic images. This continues setting up the basis for developing feasible deep learning navigation algorithms for orbiting minor bodies. The usage of multiple shape

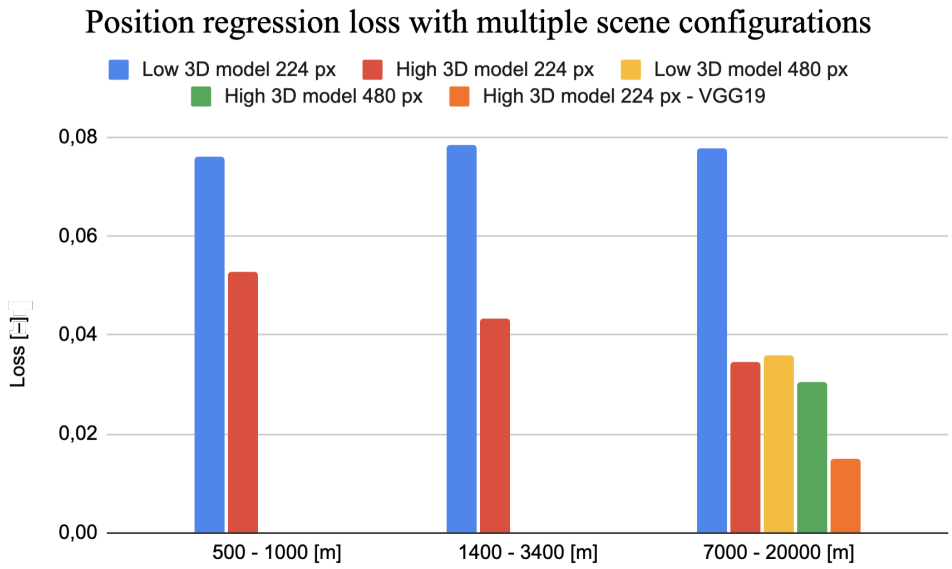

Figure 14: Effects on position estimation loss of shape model and image resolution.

models with variable spatial resolution together with the addition of a detail albedo map was key to filling the gap between synthetic and real images. It is true that high-detail shape models of future missions targets required for generating the synthetic training sets, may not be available ahead of the spacecraft arrival and in-situ measurements have taken place. Nonetheless, there are some cases of shape models based on punctual observations of past and current missions, like for Phobos based on Mars Express HRSC observations, for Deimos based on Viking and more recently, Hope images, or for the binary asteroid system composed by Didymos and Dimorphos visited by DART and LICI-Acube which will be later on visited by Hera spacecraft. In addition, the implementation of more complex architectures like the Time Distributed CNN and state-of-the-art CNNs like VGG, ResNet or DenseNet have been proven to boost the achieved accuracy, which shall be further improved for more restrictive scenarios.

6 ACKNOWLEDGEMENT

This research is part of the R+D+i project TED2021-132099B-C31 funded by MCIN/AEI (10.13039 / 501100011033) and by the European Union NextGenerationEU PRTR. The authors also acknowledge the Principal Investigator(s) Rizk, B. (University of Arizona) of the OCAMS instrument onboard the Osiris-Rex mission for providing data sets in the data archive.

REFERENCES

[1] R. Sagdeev *et al.*, “Vega spacecraft encounters with comet halley,” in *Nature*, 1986, pp. 259–262.

[2] C. Bishop, “Neural networks for pattern recognition,” in *Neural Networks for Pattern Recognition*, 1995, p. 238.

[3] C. Bishop *et al.*, “Neural networks for pattern recognition,” Advanced Texts in Econometrics, 1995.

[4] B. Williams, "Technical challenges and results for navigation of near shoemaker," *Johns Hopkins APL Tech. Dig.*, vol. 23, Jan. 2002.

[5] M. Barucci *et al.*, "Rosetta asteroid targets: 2867 steins and 21 lutetia," vol. 128, no. 1-4, pp. 67–78, 2007. DOI: 10.1007/s11214-006-9029-6.

[6] N. Mastrodemos *et al.*, "Optical navigation for the dawn mission at vesta," vol. 140, pp. 1739–1754, 2011.

[7] A. Krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

[8] A. Kendall *et al.*, "Posenet: A convolutional network for real-time 6-dof camera relocalization," Dec. 2015, pp. 2938–2946. DOI: 10.1109/ICCV.2015.336.

[9] A. Accomazzo *et al.*, "The final year of the rosetta mission," *Acta Astronautica*, vol. 136, pp. 354–359, Jul. 2017. DOI: 10.1016/j.actaastro.2017.03.027.

[10] R. Linares *et al.*, "A deep learning approach for optical autonomous planetary relative terrain navigation," Feb. 2017.

[11] M. Lapin *et al.*, "Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1533–1554, 2018. DOI: 10.1109/TPAMI.2017.2751607.

[12] S. Sharma *et al.*, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," 2018.

[13] J. Qi *et al.*, "On mean absolute error for deep neural network based vector-to-vector regression," *IEEE Signal Processing Letters*, vol. 27, pp. 1485–1489, 2020. DOI: 10.1109/LSP.2020.3016837.

[14] S. Sharma and S. D'Amico, "Neural network-based pose estimation for noncooperative spacecraft rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 4638–4658, 2020.

[15] B. Semenov (NAIF), "Osiris-rex archived spice kernel dataset," 2021.

[16] A. Garcia *et al.*, "Lspnet: A 2d localization-oriented spacecraft pose estimation neural network," Apr. 2021.

[17] Lunar and Planetary Laboratory, University of Arizona, "Bennu osiris-rex ocams global albedo mosaic 6.25cm v6," 2021.

[18] A. Escalante *et al.*, "Churinet - applying deep learning for minor bodies optical navigation," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–14, 2022. DOI: 10.1109/TAES.2022.3227497.

[19] M. Pugliatti *et al.*, "Neural network-based pose estimation for noncooperative spacecraft rendezvous," *Data-Driven Image Processing for Onboard Optical Navigation Around a Binary Asteroid*, pp. 1–17, 2022.