



**POLITECNICO**  
MILANO 1863

[RE.PUBLIC@POLIMI](mailto:RE.PUBLIC@POLIMI)

Research Publications at Politecnico di Milano

## Post-Print

This is the accepted version of:

M. Pugliatti, M. Maestrini

*Small-Body Segmentation Based on Morphological Features with a U-Shaped Network Architecture*

Journal of Spacecraft and Rockets, Published online 19/09/2022

doi:10.2514/1.A35447

The final publication is available at <https://doi.org/10.2514/1.A35447>

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/11311/1221269>

# Small-body segmentation based on morphological features with a UNet architecture

Mattia Pugliatti <sup>\*</sup>, Michele Maestrini. <sup>†</sup>  
*Politecnico di Milano, 20156, Milan, Italy*

Small-bodies such as asteroids and comets display great variability in terms of surface morphological features. These are often unknown beforehand but can be employed for hazard avoidance during landing, autonomous planning of scientific observations, and navigation purposes. Algorithms performing these tasks are often data-driven, which means they require realistic, sizeable, and annotated datasets which in turn may rely heavily on human intervention. This work develops a methodology to generate synthetic, automatically-labeled datasets which are used in conjunction with real, manually-labeled ones to train deep-learning architectures in the task of semantic segmentation. This functionality is achieved by designing UNet architectures trained with different strategies. These show good generalization capabilities, implement uncertainty quantification estimates and can be hybridized to exploit qualities from multiple networks.

## Nomenclature

C-1	=	Dataset of synthetic images for classification
CNN	=	Convolutional Neural Network architecture
$\gamma$	=	Weighting parameter used in the UNet <sub>H</sub>
$\Gamma$	=	Maximum size of the bounding box of the small-body
$D_0$	=	Saturation distance
D-1,2,3	=	Datasets of synthetic images
D-4	=	Dataset of real images
IoU	=	Intersection over Union
$FOV$	=	Field of view
$\hat{\mathbb{H}}[y x]$	=	Predictive entropy
$l_i^j$	=	i-th layer of the j network

---

<sup>\*</sup>PhD Student, Department of Aerospace Science and Technology, Via La Masa 34; mattia.pugliatti@polimi.it

<sup>†</sup>Postdoc Research Fellow, Department of Aerospace Science and Technology, Via La Masa 34; michele.maestrini@polimi.it

Part of this work was presented as paper No. AAS 21-378 at the 2021 AAS/AIAA 31 st Space Flight Mechanics Meeting, February 1 - 3, 2021, Charlotte , North Carolina.

mIoU	=	Mean Intersection over Union
P	=	Number of samples to compute the predictive entropy
SCCE	=	Sparse Categorical Cross Entropy loss metric
UNet <sub>S</sub>	=	UNet architecture trained with synthetic images
UNet <sub>S</sub> <sup>A</sup>	=	UNet architecture trained with real images from UNet <sub>S</sub>
UNet <sub>R</sub>	=	UNet architecture trained with real images from scratch
UNet <sub>H</sub>	=	Hybrid UNet architecture between UNet <sub>S</sub> and UNet <sub>S</sub> <sup>A</sup>
$x$	=	Input
$y$	=	True output
$\hat{y}$	=	Predicted output
$w_j$	=	Weight parameter of the j-th class
WSCCE	=	Weighted Sparse Categorical Cross Entropy loss metric
$\theta$	=	Weights and biases of the network

## I. Introduction

**S**MALL-BODIES such as asteroids and comets are characterized by their variety of shapes, physical properties, orbital characteristics, composition, and surface morphological features. The latter are often not characterized in detail or cannot be observed at all from ground-based measurements. These features, however, could enable important autonomous on-board capabilities. While a variety of algorithms have been developed to detect, match, and track image's features generated by morphological properties for optical navigation and shape reconstruction purposes around a small-body [1, 2], fewer methods have been developed to discern between the different classes these morphological features belong to. Semantic segmentation can be defined [3] as the capability to perform both object recognition and accurate boundary segmentation at pixel level. Performing semantic segmentation can be seen as a way to transform image content into a new space in which each pixel contains a meaning. In this work, this is the taxonomic connotation of the morphological feature described by that pixel: background, surface, crater, boulder, or the terminator region of a small-body.

Image segmentation has been used in previous space-related works as a means to classify geological properties of the terrain in [4] or as a way to identify between plumes and jets from comets and moons as in [5], which uses a combination of simple pixel-intensity based methods and geometric considerations. In [6], a series of advanced image processing techniques for enhanced flyby science around small-bodies are introduced. Amongst them, a methodology for autonomous features detection supported by simple filtering and statistical-based classification is introduced, alongside image segmentation to distinguish between features, surface, and background pixels. With the progress of artificial

intelligence, and in particular deep-learning, architectures for image-segmentation applications have boomed. Most notably, outside from the space domain, in [7] a new successful architecture, also called U "shaped" Network (UNet), is introduced which performs segmentation for biomedical applications with two distinct design choices: a symmetric U-shaped structure lacking the fully connected layer between encoder and decoder and the concatenation of copies of frozen trained encoder layers in the decoder to retain feature-extracting capabilities at different scales between the two encoding and decoding portions of the network. This architecture is extensively used for its design and implementation simplicity in works such as in [8, 9] to obtain hazard maps for safe lunar landing site selections or most notably in [10], which compare the UNet architecture amongst other ones for hazard detection for small-body landing applications. The accuracy of deep-learning-based methods is also highlighted in [11] with a set of 5 different Convolutional Neural Network (CNN) architectures to detect geological structures at varying scales for the Mars Reconnaissance Orbiter mission, demonstrating that their methodology outperforms other states of the art methods used previously for the same task.

Another critical element of interest for the segmentation task is uncertainty quantification. Pixel's classes can be associated with uncertainty metrics that quantify the robustness of the segmentation. Most notably, in [12] a methodology is described to generate predictive entropy as a way to quantify uncertainty for semantic segmentation networks based on Bayesian inference. The predictive entropy accounts for both aleatoric and epistemic uncertainty and is adopted in [13] to further extend the work performed in [9] to generate uncertainty-aware capabilities for the selection of feasible landing sites on the Moon.

The aforementioned works in the literature contain three major inconveniences which are addressed in this work for the specific task of image segmentation about small-bodies. First, they do not fully appreciate the complexity of the classes of features existing on a small-body surface, often not considering more than 3 meaningful layers or focusing only on safe-unsafe pixel classification. Secondly, most of them do not include uncertainty quantification metrics which, taking inspiration from [13], could be of paramount importance for a real operational scenario. Third, they often require extensive manual preparation of the data since (apart from [8, 9, 13] which use digital terrain maps and [10] which exploits ray-tracing capabilities) large, realistic, annotated datasets are difficultly available. This is ultimately a major deficiency for those high-performing data-driven methods such as deep-learning UNet architectures which could demand a large amount of data to produce a reliable, robust, and highly generalized architecture.

The work presented here pivots on the prior analysis performed by the same authors in [14], which is further extended to include real image annotated datasets, smaller and hybrid architectures, and uncertainty quantification capabilities. In particular, in this work, the following novelty elements are emphasized. First, an approach is proposed for the automated labeling of surface morphological features on a small-body which can be used to generate a considerable amount of synthetic image-mask pairs describing 5 different pixel categories. This dataset is accompanied by a smaller one of manually-labeled real images from previously flown missions to asteroids. In addition, a UNet segmentation

architecture is designed, trained, and tested with different strategies with the ultimate goal to develop a method that can autonomously operate on-board a spacecraft visiting an unknown body. Because of this, particular care is given to the ability of the architecture to not only predict an accurate segmentation map but also to generate an uncertainty map that can be operationally used for reliability. Within this context, we demonstrate the importance of the training approach as well as of the hybridization of the datasets to obtain a reliable technique that can also be expected to work in more realistic and generic circumstances.

Generating segmentation-uncertainty maps pairs from raw grayscale images enables a transformation to the segmented space in which pixel content is greatly simplified and assigned a specific meaning. The authors believe that such skill could unlock new capabilities that would have been previously impossible such as autonomous scientific operations, both in terms of planning and pointing, to perform advanced hazard detection and avoidance during landing, and to perform navigation tasks, as has been previously illustrated in [15], where a method based on CNN and Normalized Cross Correlation (NCC) is designed to work considering as input segmentation maps. Indeed, these maps have much more condensed information with respect to their grayscale counterparts. Since they are already partially processed, they are expected to put lower strain on the computational resources for further advanced processing.

The rest of the paper is arranged as follows. The methodology is presented in Sec. II which defines the framework to generate both the automated and the manually labeled datasets, the architectures of the neural networks considered as well as the algorithm to compute the segmentation uncertainty. In Sec. III the performance of the different networks are detailed both for the synthetic and real datasets in subsection III.A and subsection III.B respectively. The performance of the uncertainty assessment are remarked in subsection III.C while a new hybrid architecture is detailed in subsection III.D. In Sec. IV some final considerations are illustrated as well as future works. Finally, two appendices complete the discussion with the network's characteristics, hyper-parameters and training history, and mosaic views of the predicted masks and associated uncertainty maps.

## II. Methodology

### A. Datasets generation

Two families of datasets are used for training, validation, and testing of the segmentation method: synthetic and real. The former is obtained from an artificially modified version of existing small-body shape models enriched by geological features such as boulders and craters. The latter is obtained by processing imagery from previously flown missions. The steps involved in the generation of such datasets are illustrated in detail in this section.

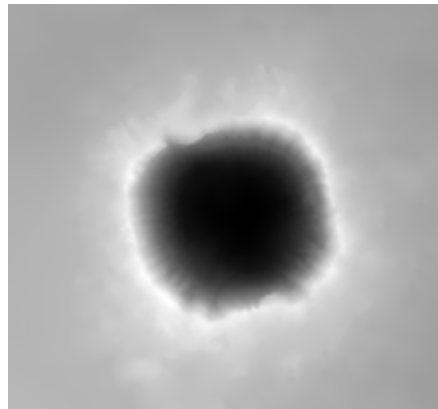
#### 1. Synthetic

To generate datasets of synthetic images with an abundant and diverse presence of morphological features, an approach has been developed in this work based on the artificial enhancement of existing shape models. This approach

represents a novel framework introduced in this work which can be further extended to other space-related applications. The approach can be divided into two phases. First, an enhanced shape model is generated from an existing "base" model of a known small-body. Second, said model and its byproducts are used to generate image-segmentation mask pairs.

Starting from the model enhancement portion, the first step consists in getting the rough shape models (i.e., the "base" models) of 9 real small-bodies from existing databases <sup>\*†</sup>. The models chosen are 67P/Churyumov–Gerasimenko, (101955) Bennu, (65803) Didymos, (6489) Golevka, 103P/Hartley, (8567) 1996 HW1, (10) Hygiea, (21) Lutetia, and (88) Thisbe which are referred in the rest of the paper by their short names.

Their low-resolution meshes are then modified to generate higher resolution ones and surface roughness is simulated with texture displacement. Artificial craters are then applied on the models as different objects by using the *shrinkwrap* modifier in Blender <sup>‡</sup>. Each crater is generated in Blender by extracting a height-map from a real texture-map of existing craters on Earth <sup>§</sup> and applying it as texture-displacement on a planar mesh. Fig. 1 shows the height-map of the Barringer crater [16] in Arizona as an example. Random scaling on all three axes of the crater's mesh is applied for each instance to generate multiple and diverse craters which are then manually stitched on the shape models. The number of craters added on each model following this procedure is summarized in Tab. 1.



**Fig. 1 Grayscale heightmap of Barringer crater used as texture for the craters on the bodies.**

Boulders generation follow a similar procedure. The *Rock Generator* add-on in Blender is used to generate a large, random, set of rocks with varying characteristics. Each element is grouped in one of three classes depending on their qualitative size: small, medium, and large. The number of boulders is illustrated in summarized in Tab. 1 for each model. The population of boulders is then applied to the shape models exploiting Blender's particle system. Differently than craters, boulders are positioned automatically by the particle system, whose randomization parameters can be adjusted to obtain the desired effects.

---

<sup>\*</sup><https://sbn.psi.edu/pds/shape-models/>

<sup>†</sup><https://3d-asteroids.space/>

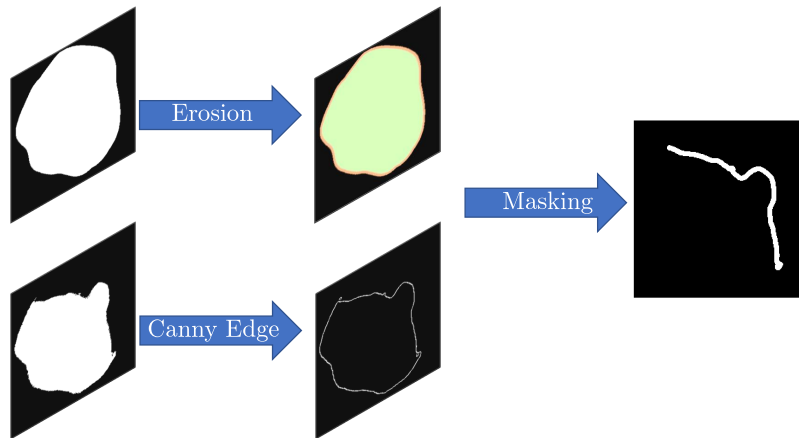
<sup>‡</sup><https://www.blender.org/>

<sup>§</sup><https://tangrams.github.io/heightmapper/>

Following this procedure, the model enhancement portion of the framework generates three models for each small-body:

- **Clean model:** it is a model with a simplified mesh, represented by the base models without surface texture.
- **Crater model:** it is a model with a refined mesh, texture, and craters. The craters are laid on the mesh but are not merged. This is important because it allows us to give each crater an individual identifier for the ray-tracing rendering engine.
- **Full model:** it is a model of the asteroid with textures, craters, and boulders. Differently than the previous model, the craters are now fully merged to the mesh. Boulders are not merged with the rest of the small-body, making it possible to associate to them a unique identifier for the ray-tracing rendering engine.

These models are used in the second portion of the approach, the generation of the image-mask pairs, to produce labeled datasets. Each segmentation map is represented by a 5 layers mask representing arbitrary morphological features. From (0) to (4) these are: (0) Background, (1) Surface, (2) Craters, (3) Boulders, and (4) Terminator region. The *clean model* is used to generate the ground truth of the terminator region with a dedicated image processing pipeline in Matlab<sup>¶</sup>, illustrated in Fig. 2. First, a Canny edge extractor [17] is applied, providing both the sharp edge between small-body and background space and the gradual one visible in the terminator region. To avoid considering spurious edges, an acceptance mask is computed which exploits the asteroid pass index mask which does not account for shadows. However, this binary image would not exclude the body-space edges obtained via the Canny extractor. To avoid including them, the boolean acceptance image is eroded using a morphological operation with a circular structuring element [18].



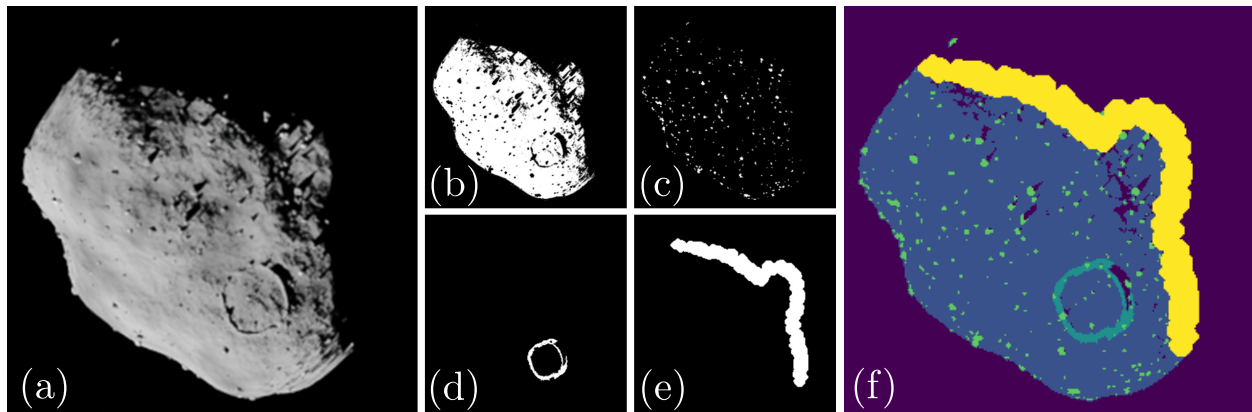
**Fig. 2** Extraction of terminator region from the *clean model*.

The background, surface, craters, and boulders masks are easily generated using the *Cycles* ray-tracing rendering engine in Blender and exploiting different identifiers assigned to the surface, craters, and boulders. This approach

<sup>¶</sup><https://it.mathworks.com/products/matlab.html>

has been inspired by the work of [19]. Craters masks are obtained from the *crater model* while surface and boulders are obtained from the *full model*. Using different pass indexes for each feature allows us to generate boolean maps illustrating for each pixel the identifier of the object which stimulates the sensor. Once all 5 raw masks are generated, they need to be hierarchically stacked together to avoid pixels overlapping between classes. Intuitively, the hierarchy used in order of decreasing priority is terminator, boulders, craters, surface, and background. Fig. 3 shows an example of the input, raw boolean masks, and final segmentation mask for the asteroid Lutetia.

Finally, the grayscale image, input of the segmentation architecture presented in this work, is a byproduct generated from the *full model*. The raw grayscale image obtained from rendering in Blender is further modified with the addition of artificial noise in Matlab. Gaussian noise with mean 0.1 and variance 0.0001 is added to each image, followed up by a 2D Gaussian smoothing kernel with a standard deviation set to 0.1. The purpose of this step is to introduce only a small variability in the pixel content of the images to allow the networks to generalize to noise, not to model any type of specific camera as the values were arbitrarily selected by the authors. This assumption should indeed be challenged by further studies on the topic by possibly estimating them from real data acquired by representative hardware [20].



**Fig. 3 Complete image of the small-body model of Lutetia (a), surface (b), boulders (c), craters (d), and terminator region (e) layers and true segmentation mask (f).**

The image-mask pairs are then generated with random camera positions around each body sampled uniformly in a spherical shell whose radius spans  $[0.4D_0, 1.3D_0]$ ,  $D_0$  being the approximate range at which the body completely fills the Field Of View (FOV) of the camera (assumed to be  $10 \times 10$  deg).  $D_0$  is computed as:

$$D_0 = \frac{\Gamma}{2 \tan \frac{FOV}{2}} \quad (1)$$

where  $\Gamma$  represents the maximum length of the shape model. For simplicity, an ideal pointing to the Center of Mass (CoM) of the body is assumed. This assumption is expected to have no significant impact on the performance of the network at inference time. In fact, real images acquired by the on-board sensor could go through a pre-processing



centering step before being fed to the neural network. For each acquisition, the sun’s direction is selected randomly in an angular range from the camera boresight of  $\pm 90$  deg. In such a way, a variety of illumination conditions are considered for a realistic case scenario in which a small-body is seen from full to partial illumination conditions.

The described methodology allows to virtually generate any arbitrary amount of annotated image-mask pairs, limited only by the available rendering resources and time, which in turn enable the training of data-driven methods.

**Table 1 Summary of the craters and boulders added to each model.**

Base Model	Craters no.	Boulders no.		
		<i>small</i>	<i>medium</i>	<i>large</i>
67P	2	500	-	5
Bennu	3	1000	250	10
Didymos	4	800	30	5
Golevka	2	800	-	40
103P	3	5000	30	5
HW1	2	2000	40	5
Hygiea	5	1500	100	5
Lutetia	5	1000	350	-
Thisbe	5	1000	-	20

Four different synthetic datasets are generated, summarized in Tab. 2. D-1 is the only one used for training, validation, and testing and it is made by 7 out of the 9 available models. Indeed two bodies, HW1 and Thisbe, are only tested in D-2 and D-3 to understand the generalization capabilities of the segmentation method. In this way, it is possible to isolate in D-1 the capability of the method to generalize on images never seen during training, while in D-2 and D-3 it is possible to test wider generalization capabilities with images from models never seen during training. Also, D-3 represents a flyby scenario, which is of interest for the application of the method presented in this work. In particular, this scenario is characterized by a large excursion in the FOV occupancy of the target, even outside the envelope used during training. The characteristic of all synthetic datasets are summarized in Tab. 2 while the ones of D-1 are detailed in Tab. 3.

**Table 2 Summary of the synthetic image-mask pairs split of D-1, D-2, and D-3.**

Dataset	Models	Train	Validation	Test
<b>D-1</b>	67P, Bennu, Didymos, Golevka, 103P, Hygiea, and Lutetia	11500	1050	1050
<b>D-2a</b>	HW1	-	-	1500
<b>D-2b</b>	Thisbe	-	-	1500
<b>D-3</b>	Thisbe	-	-	56

**Table 3 Summary of the synthetic image-mask pairs constituting the training, validation and test set of D-1.**

<b>Models</b>	<b>Train</b>	<b>Validation</b>	<b>Test</b>
<b>67P</b>	1350	150	150
<b>Bennu</b>	1850	150	150
<b>Didymos</b>	1750	150	150
<b>Golevka</b>	1500	150	150
<b>103P</b>	1500	150	150
<b>Hygiea</b>	1850	150	150
<b>Lutetia</b>	1700	150	150
<b>Total</b>	<b>11500</b>	<b>1050</b>	<b>1050</b>

## 2. Real dataset

In order to assess the applicability of the method presented in this work for real mission conditions a small dataset of real images from previously flown missions is also generated. This is comprised of 200 images randomly selected from Hayabusa I [21], Hayabusa II [22], Osiris-Rex [23], Dawn [24], NEAR Shoemaker [25] respectively of (25143) Itokawa, (162173) Ryugu, (101955) Bennu, (4) Vesta, and (433) Eros. A selected set of 50 images from these missions is downloaded, cropped, or resized to snippets of 256 x 256 grayscale images. These have been manually labeled by the authors using the *labelbox* online tool <sup>‡</sup>. In order to cross-validate the labeling and simplify it, some common rules have been established: when it came to boulders, only the largest and more meaningful ones are labeled. Each author labeled a split of 60% of the original dataset, thus creating an overlap which is used as a test-bench to assess possible biases introduced by different individuals. Finally, each of the 50 manually-labeled image-mask pairs is flipped and rotated thus creating the final set of 200 pairs. As there is no consensus about the number of images to be used in a dataset, this number is deemed to be sufficient to provide the proof of concept for our work. Other databases of space images (despite the different application) only relied on 400 images [26], whereas the SPEED dataset, which represents the state-of-the-art in image datasets for space application, only provided  $\sim 300$  real images [20] while being able to rely on a much simpler annotation procedure. The split used for training, validation, and test is illustrated in Tab. 4.

**Table 4 Summary of the real image-mask pairs split of D-4.**

<b>Dataset</b>	<b>Models</b>	<b>Train</b>	<b>Validation</b>	<b>Test</b>
<b>D-4</b>	Eros, Itokawa, Vesta, Bennu	140	40	20

## 3. Classification dataset

For completeness, another dataset used to train the encoder of the segmentation method used is briefly described. Such dataset, referred to as C-1, is composed of the same images of D-1 associated with different labels but instead

<sup>‡</sup> <https://labelbox.com/>, last time accessed: 15th of March 2022.

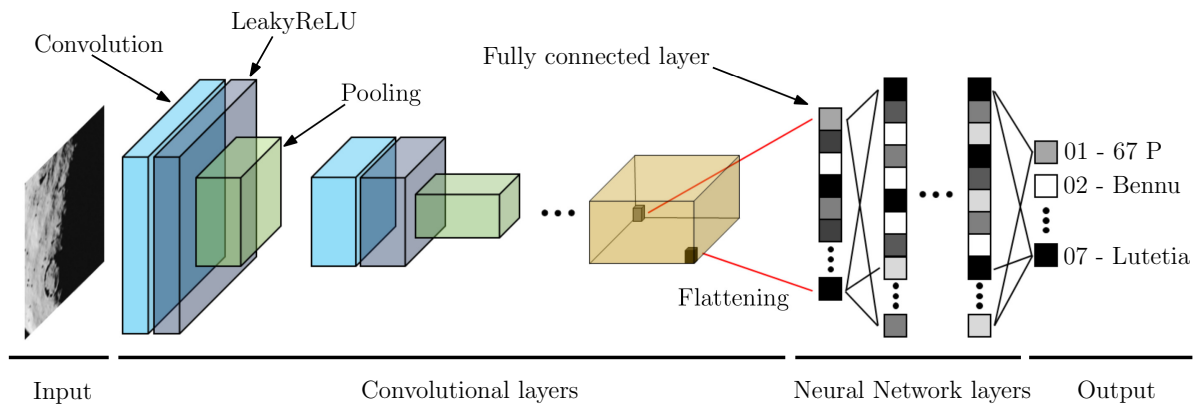
of having the segmentation masks as labels, the small-body names are considered as labels. 7 different classes are considered. The dataset split is illustrated in Tab. 5.

**Table 5 Summary of the synthetic dataset C-1.**

Dataset	Models	Train	Validation	Test
C-1	67P, Bennu, Didymos, Golevka, 103P, Hygiea, and Lutetia	10880	2720	-

## B. Convolutional Neural Network architectures

In this section, the network architectures used in this work are discussed. First, a CNN architecture is trained on C-1 to develop an encoder. The schematic architecture of the CNN is illustrated in Fig. 4 while its design and hyper-parameters are reported in Tab. 8 and Tab. 10 of the appendix for conciseness. The input is represented by a grayscale image while the output is a vector of 7 elements, each representing the *softmax* probability of belonging to that specific class.



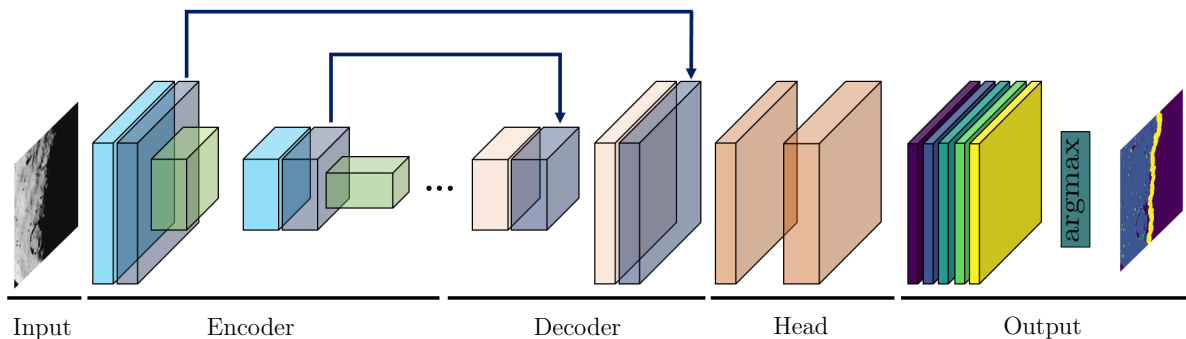
**Fig. 4 Schematic architecture of the CNN used for classification.**

The classification task is formulated as the work in [27], but with the specific purpose to develop an encoder that is capable to extract features of interest from small-body images. Such encoder is then transferred to the UNet for the final task of image segmentation. Note that in [14] this task has been exploited from a well-known architecture [28] that had been previously trained for different tasks and types of imagery than the one associated with small-bodies. Instead, in this work, the authors want to investigate whether a specific and custom-made encoder has the potential to improve the performance illustrated in [14].

In order to predict segmentation masks from grayscale images a network architecture inspired by the UNet design is developed. The efficacy and simplicity of this architecture has already been proven both in the broader computer vision domain [7] but also for specific space applications [8–10, 13, 14]. The schematic of the UNet architecture used in this work is illustrated in Fig. 5 while its design and hyper-parameters are illustrated in detail in the appendix in Tab. 9 and

Tab. 11.

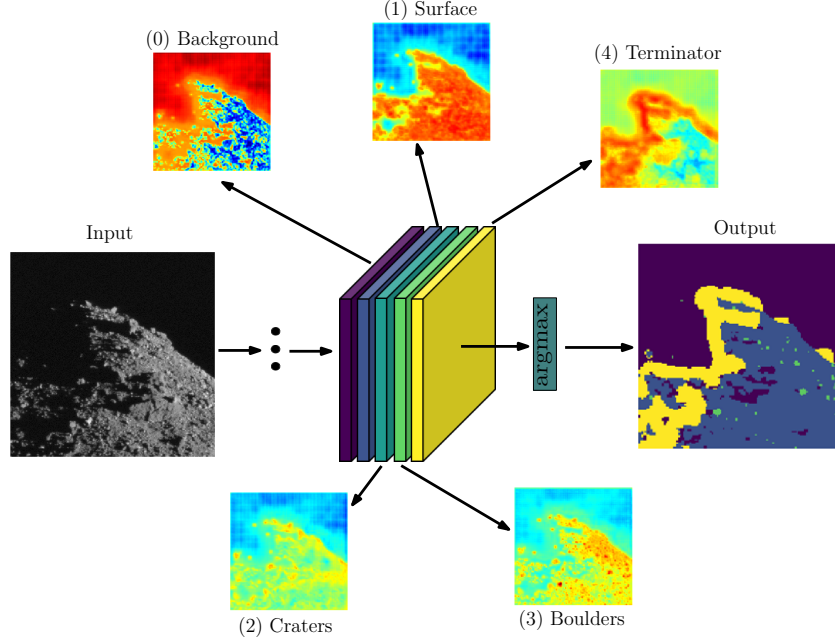
The architecture is conceptually divided into 5 portions. The input is represented by a grayscale image, the output by a preliminary  $128 \times 128 \times 5$  tensor which is processed as final output to be a  $128 \times 128$  image whose pixel values span from 0 to 4, each corresponding to a specific layer of the small-body. In Fig. 6 it is possible to see an exploded view of the output portion of the UNet. Each of the 5 layers that makes the output tensor is a  $128 \times 128$  matrix containing unbounded float values. Each pixel of the matrix represents a scalar score for that pixel belonging to that specific layer, the score being visualized with a *jet* colormap in Fig. 6; the higher the score (red) the higher the chance of that pixel belonging to that class, vice-versa for lower scores (blue). By applying the *argmax* function, the  $128 \times 128 \times 5$  tensor is reduced to a  $128 \times 128$  matrix where each pixel can assume a value from 0 to 4 (i.e., the values representing the morphological classes) so that the output matrix represents the predicted segmentation mask. Note that the colors used for each class, from the *viridis* colormap which are visible in Fig. 6, are the same that used for the remainder of this work when illustrating the segmentation masks.



**Fig. 5 Custom UNet architecture. The input is the image, the output is the semantic segmentation map.**

The contracting portion of the network (encoder) is composed of a succession of convolution, Leaky Rectified Linear Unit (LeakyReLU) activation functions, and max-pooling layers which progressively increase in depth and reduce in size (i.e. height and width). The expansive portion (decoder) is made by a combination of transpose convolution (light-red and light-pink blocks in Fig. 5), LeakyReLU, and upsampling layers of reducing the depth and increasing size. LeakyReLU are extensively used in this work since they have been observed to perform better than Rectified Linear Unit (ReLU) for space images, as observed also in [29]. This symmetric nature is what gives the network its characteristic "U" shape and name. The encoding layers of the network are responsible for extracting low-level features from the image, which is in this work is a task achieved with the encoder architecture of the CNN trained on classification. Also, note that the output of the convolutional layers of the encoder are copied and stacked in the corresponding layers of the decoder and that the network lacks a fully connected layer in the middle. The former is represented in Fig. 5 by the blue arrows linking specific layers of the encoder with the decoder.

Three instances of the UNet architecture are used throughout this work, referred to as: UNet Synthetic (UNet<sub>S</sub>),



**Fig. 6 Exploded view of the  $128 \times 128 \times 5$  output tensor before it is processed to generate the output mask.**

UNet Synthetic Augmented (UNet<sub>S</sub><sup>A</sup>), and UNet Real (UNet<sub>R</sub>). UNet<sub>S</sub> is trained over the training set of D-1. Both UNet<sub>S</sub><sup>A</sup> and UNet<sub>R</sub> are instead trained over the training set of D-4. At the beginning of training, UNet<sub>S</sub><sup>A</sup> shares the weights and biases of UNet<sub>S</sub>, thus implementing a form of transfer learning from D-1 to D-4 between the two networks. On the other hand, UNet<sub>R</sub> has the weights and biases initialized randomly at the beginning of the training. It is remarked that all three networks share the same architecture, which is just used with three different sets of weights, biases, and hyper-parameters. This is done as a proof of concept to remove architectural differences and isolate the contribution generated by transfer learning.

To train an architecture, metrics need to be defined to assess its performance. For the CNN, a simple Sparse Categorical Cross Entropy (SCCE) metric is used, defined as:

$$SSCE = - \sum_{i=1}^N y_i \log \hat{y}_i \quad (2)$$

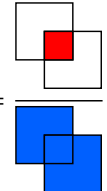
where  $\hat{y}_i$  represents the  $i$ -th output of the *softmax* output,  $y_i$  is the corresponding target value and  $N$  is the number of values in the model output (i.e., one per asteroid class). On the other hand, the UNet is trained using as loss metric the Weighted Sparse Categorical Cross Entropy (WSCCE), defined as:

$$WSSCE = - \frac{1}{M} \sum_{j=1}^M \sum_{i=1}^K w_j y_{i,j} \log \hat{y}_{i,j} \quad (3)$$

where  $w_j$  represents the  $j$ -th component of the weight vector associated with each layer in the mask. This weight serves

to re-balance the number of pixels of those less represented morphological features (i.e., craters and boulders). In this work the values of  $w_j$  have been initially determined from statistical analysis of the true masks in the training set and have been modified as part of the hyper-parameter tuning during training. The value of WSCCE can therefore be obtained as the weighted average of SCCE computed for each of the  $K$  pixels which has to be classified as one of the  $M$  geological features.

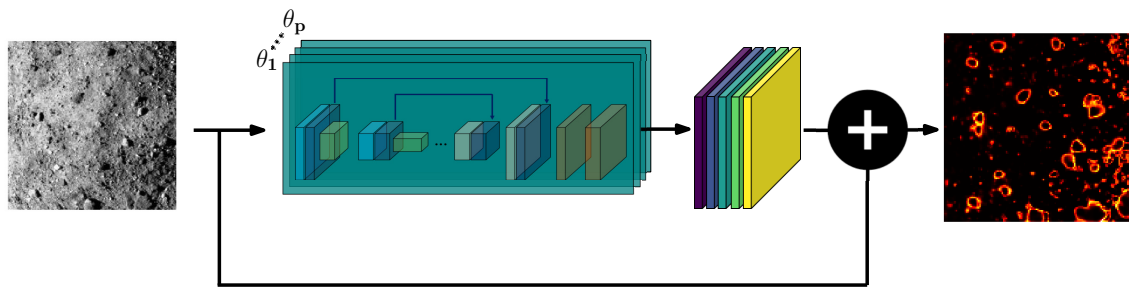
To assess the accuracy of the segmentation, additional metrics are introduced. First, the Intersection over Union (IoU) is defined as a simple ratio between two areas:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Area of Red Square}}{\text{Area of Blue and Red Squares}} \quad (4)$$


where the numerator of this ratio is the overlap between the predicted and ground truth masks. On the other hand, the denominator is the total area encompassed by these two masks. We then define the mean Intersection over Union (mIoU) as the average of the IoU computed for each class. The former represents a global segmentation performance metric for each image or dataset, the latter a metric for each specific layer.

### C. Uncertainty quantification

In this paper a preliminary step towards the inclusion of uncertainty quantification in the pixel class prediction by the UNet is attempted. In doing so, this work leverages prior efforts on the topic, most notably of [12], which introduces a methodology to quantify uncertainty from predictive entropy, and the one of [13] which showcase how such uncertainty could be operationally used for robust safe landing-site selection.



**Fig. 7 Architecture to generate the uncertainty maps.**

Predictive entropy is capable to model both aleatoric and epistemic uncertainty [12], the first being the one caused by environmental variability, the second being the one caused by the model's uncertainty. The approach described in [12] exploits Bayesian inference and the non-deterministic nature of the network architecture achieved by incorporating dropout to quantify the uncertainty as predicted entropy. The approach is schematized in Fig. 7. The same instance of

the architecture, determined by a fixed set of weights and biases  $\theta$ , runs  $P$  times (in this work  $P = 20$ ) across the same input image, producing multiple raw output tensors, each of which represents the pixel-by-pixel score for belonging to that layer before the *argmax* function is applied. Because dropout is extensively used in the architectures, the output of the image flow at each iteration is not the same, as the dropout changes the active connections across the network. Because of this,  $P$  multiple raw output tensors are generated. From these multiple sets of raw outputs, the uncertainty is computed as pixel-wise predictive entropy as [12]:

$$\hat{\mathbb{H}}[\mathbf{y}|\mathbf{x}] = - \sum_{i=1}^K \left[ \frac{1}{P} \sum_{p=1}^P \text{softmax}(y_{i,p}|\mathbf{x}, \theta_p) \right] \cdot \log \left[ \frac{1}{P} \sum_{p=1}^P \text{softmax}(y_{i,p}|\mathbf{x}, \theta_p) \right] \quad (5)$$

where  $\mathbf{x}$  and  $\mathbf{y}$  are respectively the vector of input and output of the network for each pixel,  $K$  is the number of classes over which the predictive entropy is computed ( $K = 5$  in this work),  $p$  is referred to the  $p$ -th sample considered and finally  $\text{softmax}(y_{i,p}|\mathbf{x}, \theta_p)$  is the probability that the pixel is assigned by the network to the  $i$ -th class during  $p$ -th sample given the input  $\mathbf{x}$  and obtained with a set of weights, biases, and dropout combination provided by  $\theta_p$ . Note that the weights and biases are the same but the dropout randomly nullifies the connections both during training and testing, hence  $\theta_p$  represents a specific instance of the network. Also note that nominally the network prediction is generated by using the *argmax* function as illustrated Fig. 5 but that the softmax function is used instead to generate  $\hat{\mathbb{H}}[\mathbf{y}|\mathbf{x}]$ . By applying this approach for each pixel, an uncertainty map can be generated. The uncertainty map represented in Fig. 7 using the *inferno* colormap shows how low (black) and high (yellow) uncertainties are scattered across the predicted mask.

By setting a global threshold (quantified as a scalar between 0 and 1, which scales between the minimum and maximum values of predicted entropy for each image), pixels from the predicted mask associated with uncertainties higher than this threshold are assigned to a  $6^{th}$  additional layer, specific for uncertain pixels. To select an appropriate value for such a threshold, the validation datasets of D-1 and D-4 are used for all architectures. The threshold is selected as that maximizing the mIoU for all images in the validation sets, which is then used in inference only on images from the test sets. The threshold is found to be equal to 0.91, 0.54, 0.48, and 0.23 respectively for the UNets<sub>S</sub>/D-1, UNets<sub>S</sub>/D-4, UNet<sub>S</sub><sup>A</sup>/D-4, and UNet<sub>R</sub>/D-4 pairs.

### III. Results

In this section, the results are illustrated and discussed in four separate sections. First, the performance of UNets over the test sets of the D-1, D-2, and D-3 datasets are illustrated in subsection III.A. Then, the performance of UNets, UNet<sub>S</sub><sup>A</sup>, and UNet<sub>R</sub> are illustrated for the test set of D-4 in subsection III.B. The application of the uncertainty maps is discussed in subsection III.C. Finally, in subsection III.D an interesting operative mode for a qualitatively more robust mask prediction is briefly discussed.

### A. Test with synthetic datasets

The performance of UNets over the synthetic datasets are summarized in Tab. 6. It is remarked that the mIoU over D-1 drops from  $\sim 60\%$  on the validation set (see Fig. 22) to  $\sim 56\%$  on the test set, which is also the highest value of mIoU achieved across all synthetic test sets. Albeit this drop in performance, the network shows good generalization performance when it is tested with new small-bodies never seen during training (D-2a and D-2b) and during a flyby case with a small-body never seen during training which also happens outside the training envelope considered.

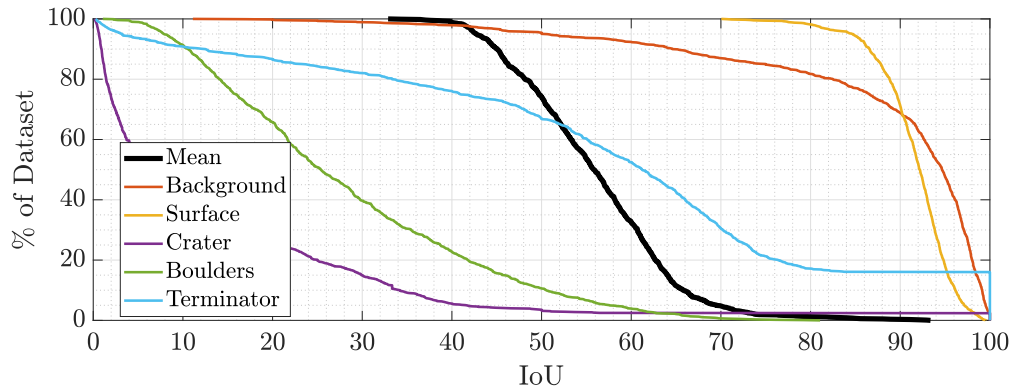
**Table 6** Table with a summary of the mIoU of the UNets for the different test cases expressed as a percentage.

<i>Test case</i>	<b>D-1</b>			<b>D-2a</b>			<b>D-2b</b>			<b>D-3</b>		
	<i>min</i>	<i>mean</i>	<i>max</i>	<i>min</i>	<i>mean</i>	<i>max</i>	<i>min</i>	<i>mean</i>	<i>max</i>	<i>min</i>	<i>mean</i>	<i>max</i>
<b>Background</b>	11.11	<b>88.16</b>	100.00	71.36	<b>95.91</b>	99.82	6.25	<b>88.05</b>	100.00	30.61	<b>94.08</b>	100.00
<b>Surface</b>	70.05	<b>91.57</b>	99.54	43.01	<b>91.63</b>	97.31	71.09	<b>91.60</b>	99.18	1.11	<b>73.63</b>	98.88
<b>Craters</b>	0.16	<b>14.09</b>	100.00	0.17	<b>10.68</b>	100.00	0.03	<b>9.20</b>	100.00	0.48	<b>11.91</b>	47.62
<b>Boulders</b>	1.08	<b>28.34</b>	81.06	1.16	<b>19.97</b>	58.46	2.79	<b>23.35</b>	59.28	2.13	<b>16.53</b>	50.21
<b>Terminator</b>	0.27	<b>57.52</b>	100.00	0.10	<b>44.67</b>	100.00	0.09	<b>60.50</b>	100.00	0.41	<b>58.19</b>	100.00
<b>Mean</b>		<b>55.93</b>			<b>52.57</b>			<b>54.54</b>			<b>50.87</b>	

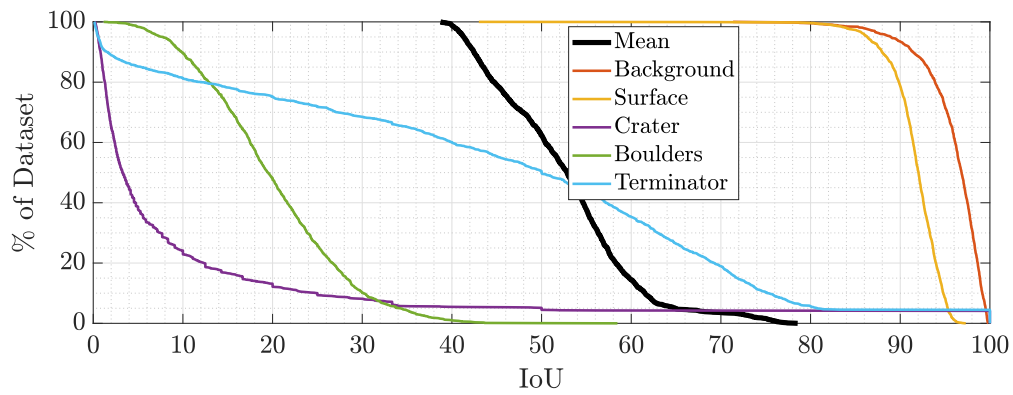
It is also noted that the performance presented in this work is slightly lower with respect to the one illustrated in [14]. Being the datasets the same between these two works, the difference is explained by two key differences. First, in this work, a much smaller encoder and UNet architecture are used with respect to the one in [14]. The former is about 5 times smaller, passing from a model made of 1842944 parameters to one of 392320 parameters. The latter is about 4 times smaller, passing from 4450757 to 1225413 parameters. Second, in this work, the authors wanted to investigate what level of performance could be achieved with an encoder specifically trained on small-body imagery in order to assess the real benefit of having a large pre-trained model specialized in detecting features from another visual domain. Being the performance slightly worse or similar, it is concluded that the specialization of the encoder for asteroid features does not seem to bring any particular advantage (apart from the smaller architecture) in terms of overall network performance.

In Fig. 8, Fig. 9, and Fig. 10 it is possible to see the cumulative values of IoU and mIoU across all datasets. In particular, the capability of the UNet to accurately detect background, surface, and terminator regions boost the mIoU performance. On the other hand, the network's worse performances in robustly detecting craters and boulders seem to be the key element in dragging down the mIoU.

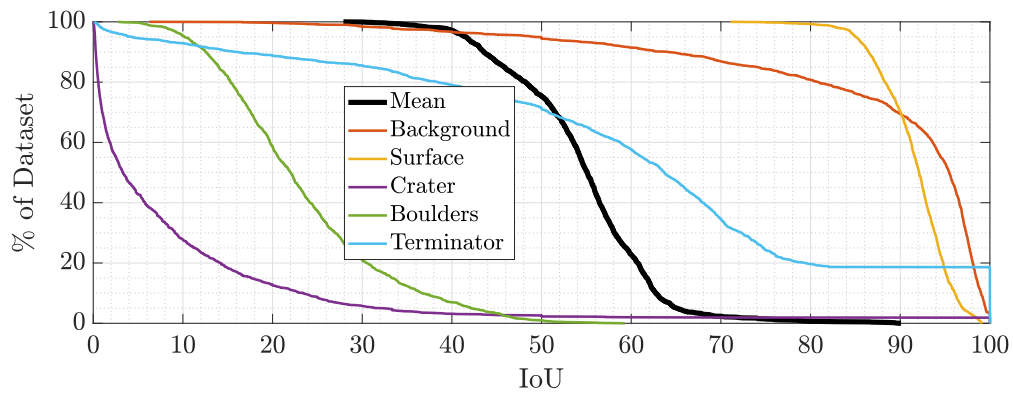




**Fig. 8** Cumulative mIoU and IoU (coloured) for each class of the UNetS on the test set of D-1.

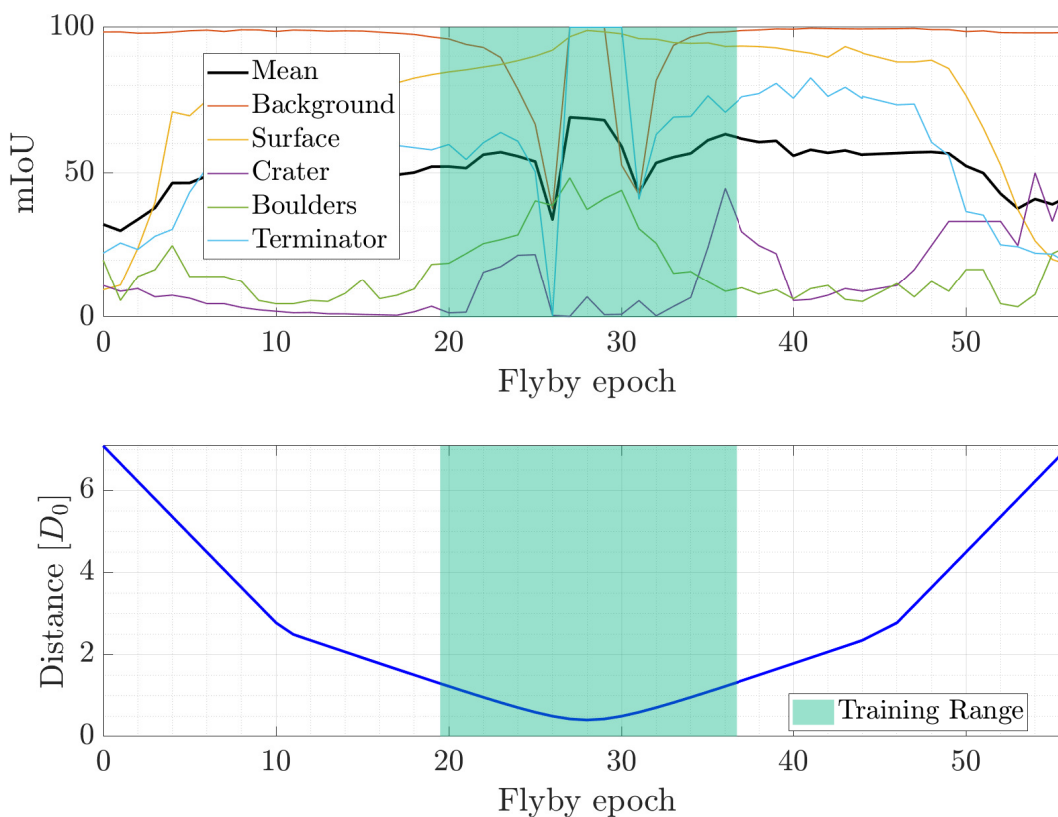


**Fig. 9** Cumulative mIoU (black) and IoU (coloured) for each class of the UNetS on the test set of D-2a.



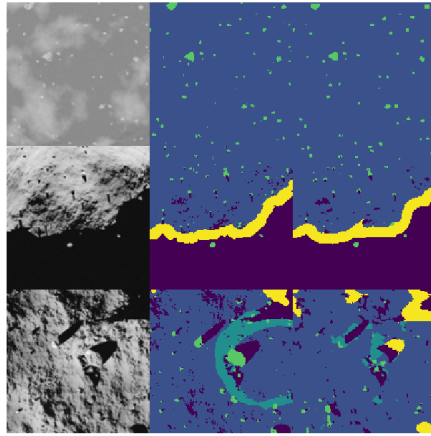
**Fig. 10** Cumulative mIoU (black) and IoU (coloured) for each class of the UNetS on the test set of D-2b.

In Fig. 11 is possible to see the network performance in a flyby scenario, which can be of interest for the application of the UNet. It is possible to appreciate the increased accuracy in the detection of boulders as the camera gets closer to the body, as well as the incapability of the network to accurately predict craters. It is also remarked that the shape model of Thisbe for D-2b and D-3 has been specifically designed to account for two nested craters, which is a feature never seen in training, to assess the capability of the network to unusual features. As it is possible to see both in Fig. 11 and Fig. 13, this setup seems to deeply affect the capability of the network to generate an accurate prediction. It is also noted that the network suffers in the prediction of smaller features (craters and boulders) observed from higher distances during the flyby. This behavior is expected since these features only occupy a few pixels and are very difficult to predict.

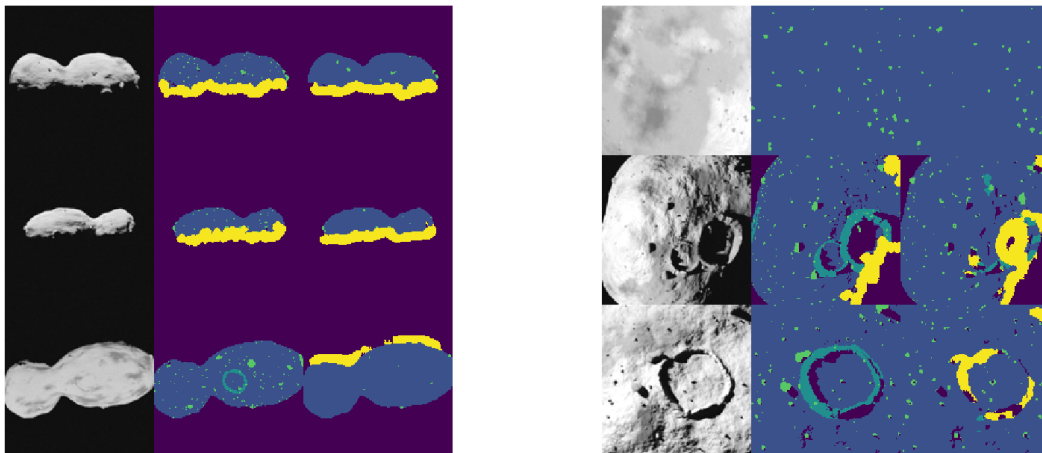


**Fig. 11 Cumulative mIoU (black) and IoU (coloured) for each class of the UNet<sub>s</sub> in test set of D-3 as function of time (top). Distance from the asteroid as function of time (bottom).**

Fig. 12 and Fig. 13 showcase small  $3 \times 3$  mosaics of the best, mean and worst values of mIoU for each dataset. More comprehensive mosaics are illustrated in the appendix for the interested readers, respectively in Fig. 23 and Fig. 24 for the D-1 and D-3 test sets.



**Fig. 12** input (left), true (center), and predicted (right) masks by UNet<sub>s</sub> for the best (top), average (middle) and worst (bottom) values of mIoU in the test set of D-1.



**Fig. 13** input (left), ground truth (center), and predicted (right) masks by UNet<sub>s</sub> in the best (top), average (middle) and worst (bottom) values of mIoU in the test set of D-2a (left mosaic) and D-2b (right mosaic).

## B. Test with real datasets

In the assessment performed in this section, the capability to design a network that can be robustly deployed for on-board applications is investigated by three different instances of the UNet on the same test set of the D-4 dataset. The performance of the architectures is summarized in Tab. 7.

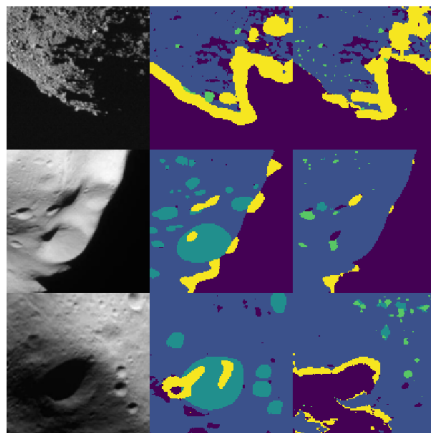
The worst performing network is UNet<sub>s</sub>, which is expected given the fact that this is the only network that has never been trained on real images. This showcases that the synthetic dataset could not be photo-realistic enough to allow the network to generalize from a synthetic to a real environment and that a relevant domain gap exists between the two. It is

also noted that this is particularly relevant for craters and boulders, while only in a smaller portion also for the terminator detection. This issue has been already illustrated qualitatively in [14] and is substantiated quantitatively in this work.

Performance is recovered with  $\text{UNet}_S^A$  and  $\text{UNet}_R$ , with the former performing slightly better than the latter. This is illustrated by the samples of predictions in Fig. 14 for  $\text{UNet}_S$  and Fig. 15 for  $\text{UNet}_S^A$  and  $\text{UNet}_R$ . In particular, from these images, it is possible to appreciate the incapability of  $\text{UNet}_S$  to detect craters, which can signal that the actual crater modeling performed on the shape models for the synthetic datasets of D-1 is too far from the real appearance of a crater.

**Table 7** Table with a summary of the mIoU for the different test cases expressed as a percentage.

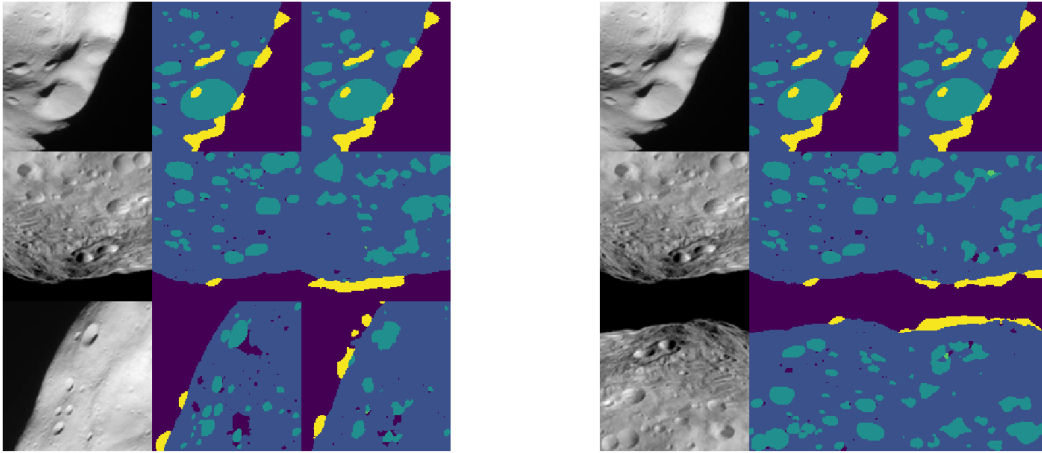
<i>Test case</i>	$\text{UNet}_S$			$\text{UNet}_S^A$			$\text{UNet}_R$		
	<i>min</i>	<i>mean</i>	<i>max</i>	<i>min</i>	<i>mean</i>	<i>max</i>	<i>min</i>	<i>mean</i>	<i>max</i>
<b>Background</b>	1.16	<b>69.84</b>	92.19	1.14	<b>76.79</b>	98.53	15.95	<b>74.00</b>	98.28
<b>Surface</b>	59.18	<b>82.00</b>	93.00	73.90	<b>88.33</b>	96.84	67.71	<b>83.58</b>	96.21
<b>Craters</b>	0.04	<b>12.63</b>	100.00	1.06	<b>56.94</b>	100.00	0.30	<b>49.47</b>	100.00
<b>Boulders</b>	0.09	<b>4.59</b>	20.00	2.78	<b>67.42</b>	100.00	1.65	<b>65.31</b>	100.00
<b>Terminator</b>	1.28	<b>24.64</b>	100.00	3.33	<b>55.68</b>	100.00	6.04	<b>52.68</b>	100.00
<b>Mean</b>		<b>38.74</b>			<b>69.03</b>			<b>65.01</b>	



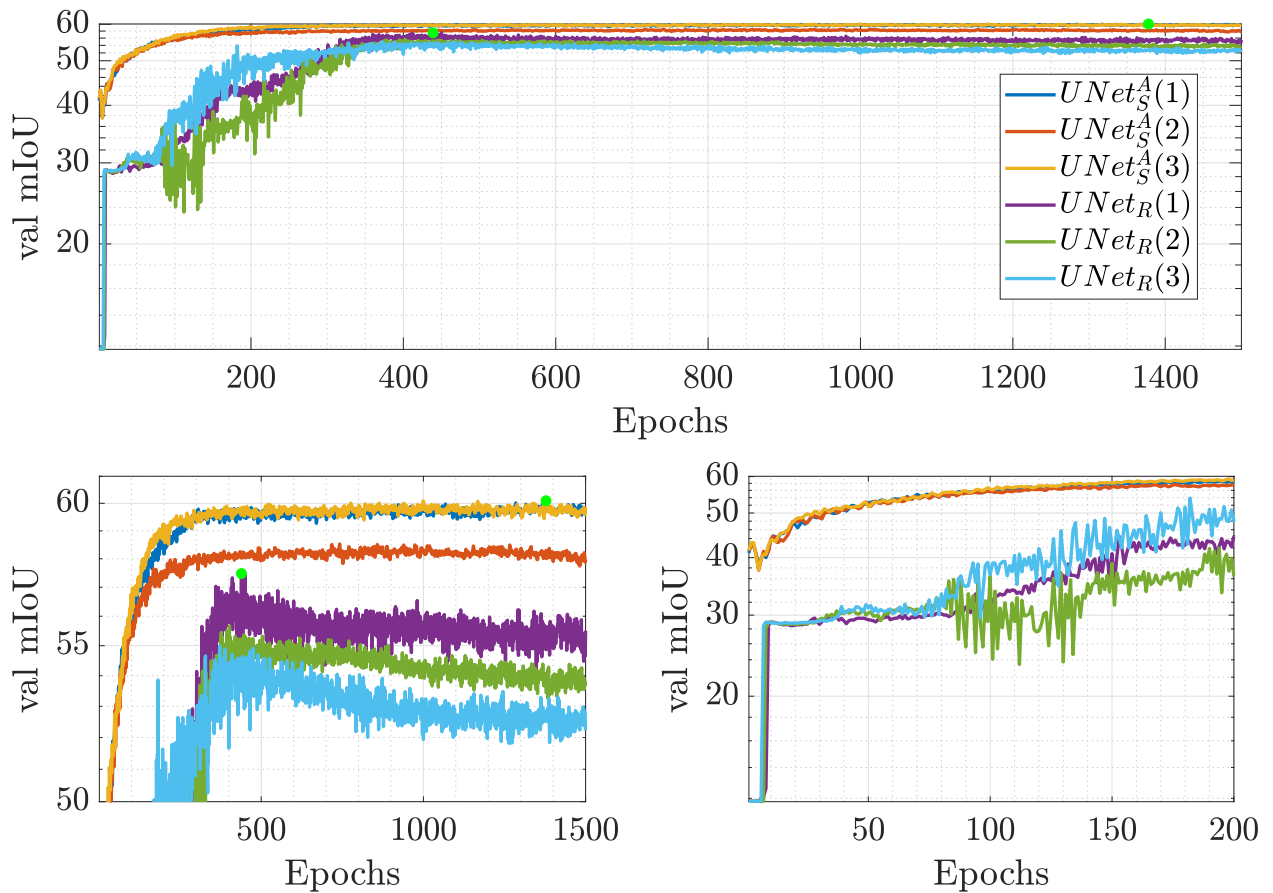
**Fig. 14** input (left), ground truth (center), and predicted (right) masks by  $\text{UNet}_S$  in the best (top), average (middle) and worst (bottom) cases of the test set of D-4.

The poor performance of  $\text{UNet}_S$  justify the need for  $\text{UNet}_S^A$  and  $\text{UNet}_R$  to exist. Indeed, these two networks are designed with an operative purpose in mind: their scope is to work with real images. The remaining point to answer is whether such training shall occur from scratch and entirely with real images ( $\text{UNet}_R$ ) or with boot-strap learning with both synthetic and real images ( $\text{UNet}_S^A$ ). From the performance in Tab. 7 the latter seems to be a preferable strategy. To support this, Fig. 16 displays the mIoU of the validation curve during training of the three best final instances of the architectures setup of  $\text{UNet}_S^A$  and  $\text{UNet}_R$ .

It is possible to appreciate the faster convergence of  $\text{UNet}_S^A$  over  $\text{UNet}_R$  as well as the higher value of the plateau



**Fig. 15** input (left), ground truth (center), and predicted (right) masks in the best (top), average (middle) and worst (bottom) cases of the test set of D-4 by  $UNet_S^A$  (left mosaic) and  $UNet_R$  (right mosaic).

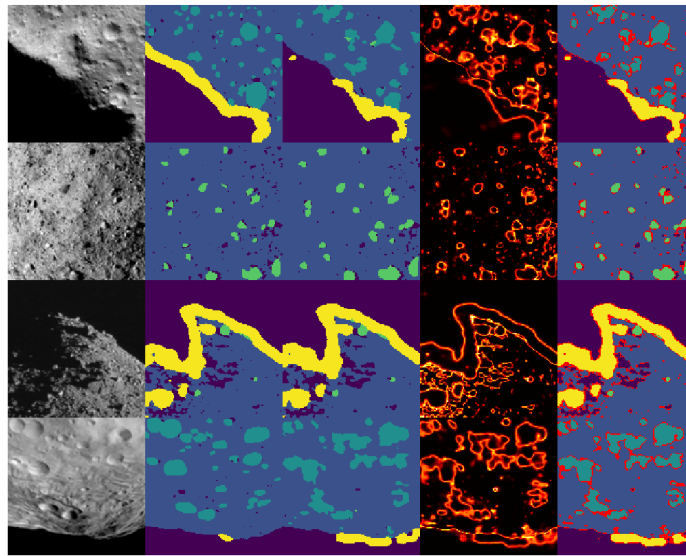


**Fig. 16** Validation mIoU for the 3 best training of  $UNet_S^A$  and  $UNet_R$ . (Top) global view, (bottom-left) zoom on the higher values of mIoU, (bottom-right) zoom early on in the training.

reached by the former over the latter. In particular, it is noted how early on during training UNet<sup>A</sup> is advantaged by the set of weights and biases inherited from the training experience of UNet<sub>S</sub>. These results justify the choice of boot-strap training, the importance of defining a synthetic dataset, and ultimately how a small, manually-labeled dataset of real images combined with it benefits the UNet for an operational application.

### C. Uncertainty maps

The uncertainty maps created with the methodology described in Sec. II with UNet<sup>A</sup> are illustrated in Fig. 17, while a larger sample of those generated by UNet<sub>S</sub>, UNet<sup>A</sup>, and UNet<sub>R</sub> are illustrated in Fig. 25. The red pixels represent those in the 6<sup>th</sup> layer, corresponding to the uncertain class.



**Fig. 17** Mosaic view of UNet<sup>A</sup> input (1st column), true and predicted masks (2nd and 3rd), uncertainty map (4th) and enhanced predicted mask (5th) for four samples of D-4.

From Fig. 17 it is possible to see that the enhanced masks contain uncertain pixels only at the boundaries between different classes. As the pixel classification is only being challenged on the borders, the uncertainty does not bring relevant changes in the enhanced maps. It is also reported that the overall performance of the networks improves only marginally when the mIoU is evaluated on the enhanced maps of the test sets. The maximum change is given to a positive variation up to 1.4% on the value of the mIoU. Finally, it is also noted that a test case where the uncertain pixels are substituted by the second most probable class has also been conducted with the same methodology but that it provided inconclusive results in terms of performance.

#### D. Hybrid segmentation

An interesting behavior has been observed by comparing the performance of  $UNet_S$  with the ones of  $UNet_S^A$  when predicting boulders. Test data showed that the automatically-labeled boulders in D-1 make  $UNet_S$  robust in detecting small to medium-size boulders. On the other hand, the manually-labeled dataset of D-4 only shows the biggest boulders. The latter has been a design choice to avoid excessive manual labor in the labeling of the dataset. These two effects are clearly visible in Fig. 18.

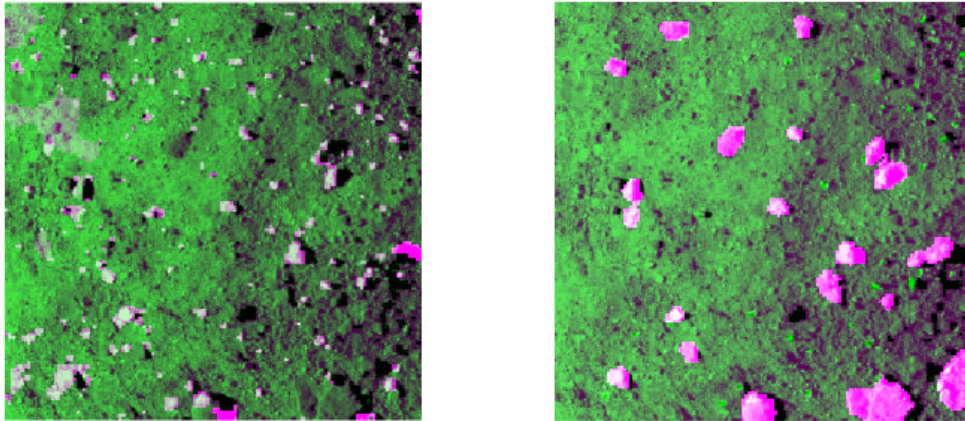


Fig. 18 Same scene of Benu surface with an overlay of the boulders detected by  $UNet_S$  (left) and  $UNet_S^A$  (right).

For these reasons, an additional architecture which makes use both of  $UNet_S$  and  $UNet_S^A$  predictions has been also implemented. Such hybrid architecture, referred to as UNet Hybrid ( $UNet_H$ ), is illustrated in Fig. 19.

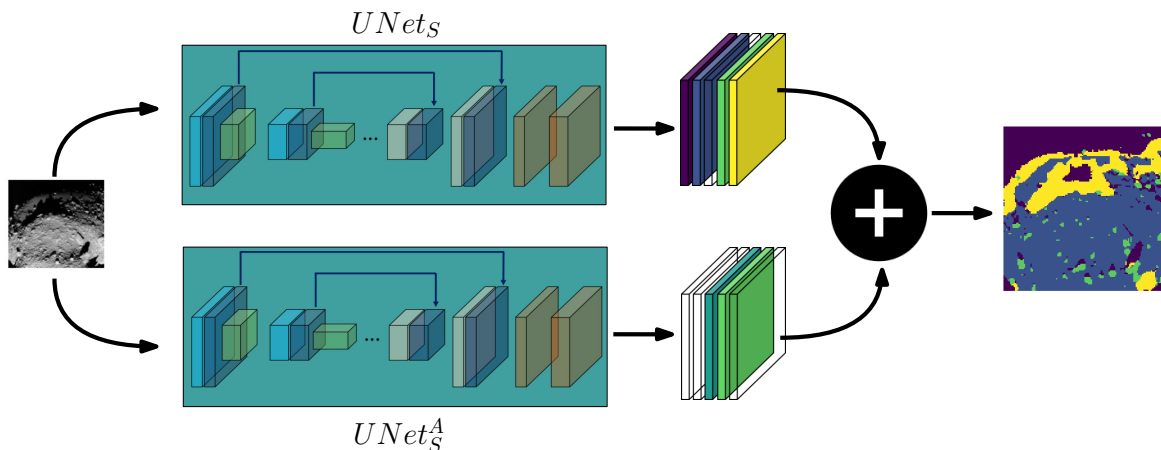


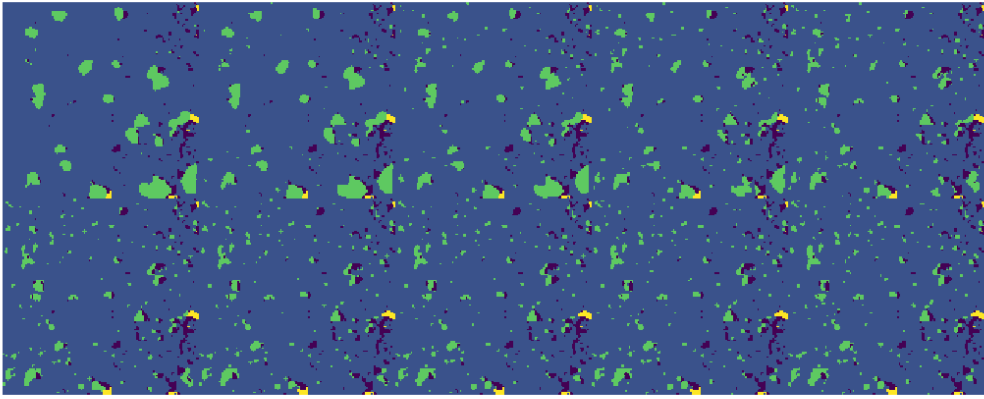
Fig. 19 Hybrid architecture  $UNet_H$  which uses the contribution of multiple networks for a qualitatively more realistic segmentation.

$UNet_H$  is realized by running the same input image twice through the same architecture instances by different sets of

weights and biases, respectively the one of UNet<sub>S</sub> and UNet<sub>S</sub><sup>A</sup>. This generates two sets of  $128 \times 128 \times 5$  raw output tensors. The first, second, and last layers (the ones corresponding to the background, surface, and terminator) are predicted entirely from UNet<sub>S</sub>. The crater's layer is predicted entirely from UNet<sub>S</sub><sup>A</sup>, which is better in detecting craters from real images than UNet<sub>S</sub>. Finally, the boulder's layer is computed by mixing the prediction from UNet<sub>S</sub> and UNet<sub>S</sub><sup>A</sup> using the following equation:

$$I_3^{\text{UNet}_H} = \log(\gamma) \cdot I_3^{\text{UNet}_S} + (1 - \log(\gamma)) \cdot I_3^{\text{UNet}_S^A} \quad (6)$$

where  $\gamma$  is a weighting parameter set to vary between 1 and 10 to mix the contribution between UNet<sub>S</sub> and UNet<sub>S</sub><sup>A</sup>. Since no reference ground truth mask is generated for this scenario, its performance are assessed only qualitatively from Fig. 20 and Fig. 21. UNet<sub>H</sub> could fuse the simplicity of training with only synthetic images with some dedicated fine-tuning performed only on craters and boulders, producing a qualitatively richer prediction. It is also noted that  $\gamma$  can be varied operationally based on the model confidence or expected envelope around the body.



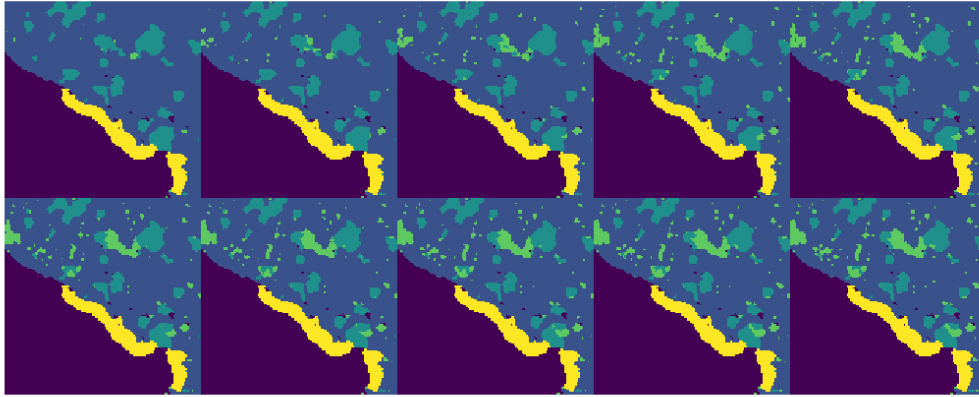
**Fig. 20** Example of hybrid predictions from UNet<sub>H</sub> with different weights. From left to right, top to bottom  $\gamma$  changes from 1 to 10.

## IV. Conclusions

A new methodology is presented to generate automatically labelled datasets for the semantic segmentation of small-bodies exploiting the ray-tracing capabilities of Blender and using simple image processing methods. Such a new framework allowed the training of deep-learning networks to perform the task of image segmentation. The architecture chosen for this task is the one of the UNet which generates for each input image a segmentation mask in which each pixel is classified in one of five different classes: background, surface, craters, boulders, and terminator region.

A UNet architecture is designed, trained, and tested with different approaches with the ultimate goal of developing a





**Fig. 21** Example of hybrid predictions from  $\text{UNet}_H$  with different weights. From left to right, top to bottom  $\gamma$  changes from 1 to 10.

robust network that can be deployed on-board to operate with real images. Because of this, particular consideration is also given to the ability of the UNet to not only predict an accurate segmentation map but also to accompany it with an uncertainty map, which can be used operationally to weight the reliability of the prediction.

To start, the UNet is trained only with synthetic images. This network, also referred to as  $\text{UNet}_S$ , displays adequate generalization capabilities, but at the same time performs poorly over the real images of D-4. To enhance performances, two different approaches are put in place. The first one consists of a boot-strap training of the UNet from the best set of weights and biases found during the training of  $\text{UNet}_S$ . The network implemented with this strategy is referred to as  $\text{UNet}_S^A$ . On the other hand, in the second strategy, a  $\text{UNet}_R$  is trained from scratch with the set of weights and biases selected randomly at the beginning of the training. The first strategy proved to be successful in improving the overall performances, showcasing the importance that hybridization of the dataset may play during training. A hybrid synthetic-real dataset has the potential to improve overall performance while lessening the cost of manual labeling of real images, which can be just a slice of a bigger dataset.

The ability of the network to predict its uncertainty metric is also explored as an additional skill. This is a useful instrument to eliminate uncertain regions, which however are usually found to be located around the boundaries between features. The uncertain pixels are labeled as part of a sixth additional class (uncertain) which can be discarded from the predicted mask. This skill brings only marginal improvements on the network performances, nonetheless, it is retained as an interesting capability to be further exploited in future works.

By comparing some of the results obtained in this work with the ones in [14], it is also possible to see that smaller encoder and UNet architectures have an unfavorable impact on the performances. Another factor that may have contributed to this is given by the fact that the encoder used in [14] has been trained for images completely different

than the ones used in this work. For this reason, it has been decided to change this setup to understand whether small-body features could be embedded in the encoder with a different task (i.e., small-body shape classification). It has been observed that a comparable level of performance can be achieved with these two strategies when comparing the validation sets, while on the test set the network trained in this work seems to perform slightly worse than the one in [14]. This is nevertheless compensated by a lighter architecture, whose computational advantages shall be taken into account.

Finally, the possibility to use a hybrid architecture (UNet<sub>H</sub>) that mixes the prediction between different trained networks is shortly illustrated. Future works may focus on this strategy to optimize the value of the weighting parameter while addressing challenges related to datasets annotation. Also, in this work several real-world camera effects have not been modelled, as this analysis is mainly a proof of concept that can be further extended in future iterations. On the other hand, different architectures may be explored, a larger more exhaustive dataset of real images could be considered as well as better modeling of craters in the synthetic images to fill the existing domain gap with the real images.

## Acknowledgments

M.P. 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.

## References

- [1] Panicucci, P., "Autonomous vision-based navigation and shape reconstruction of an unknown asteroid during approach phase," Ph.D. thesis, Institut Supérieur de l'Aéronautique et de l'Espace, 2021.
- [2] Jarvis, B., Choi, G. P. T., Hockman, B., Morrell, B., Bandopadhyay, S., Lubey, D., Villa, J., Bhaskaran, S., Bayard, D., and Nesnas, I. A., "3D Shape Reconstruction of Small Bodies From Sparse Features," *IEEE Robotics and Automation Letters*, Vol. 6, No. 4, 2021, pp. 7089–7096. <https://doi.org/10.1109/LRA.2021.3097273>.
- [3] Szeliski, R., *Computer Vision*, 2<sup>nd</sup> ed., Springer International Publishing, 2022. <https://doi.org/10.1007/978-3-030-34372-9>, URL <https://doi.org/10.1007/978-3-030-34372-9>.
- [4] Thompson, D., Niekum, S., Smith, T., and Wettergreen, D., "Automatic detection and classification of features of geologic interest," *Proceedings of IEEE Aerospace Conference*, 2005, pp. 366–377. <https://doi.org/https://doi.org/10.1109/AERO.2005.1559329>.
- [5] Wagstaff, K. L., Thompson, D. R., Bue, B. D., and Fuchs, T. J., "Autonomous Real-time Detection of Plumes and Jets from Moons and Comets," *The Astrophysical Journal*, Vol. 794, No. 1, 2014, 43. <https://doi.org/https://doi.org/10.1088/0004-637X/794/1/43>.
- [6] 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," *Earth and Space Science*, Vol. 2, No. 10, 2015, pp. 417–434. <https://doi.org/https://doi.org/10.1002/2014ea000042>, URL <https://doi.org/10.1002/2014ea000042>.

- [7] Ronneberger, O., Fischer, P., and Brox, T., “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, LNCS, Vol. 9351, 2015, pp. 234–241. [https://doi.org/https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/https://doi.org/10.1007/978-3-319-24574-4_28).
- [8] Scorsoglio, A., D’Ambrosio, A., Ghilardi, L., Furfaro, R., Gaudet, B., Linares, R., and Curti, F., “Safe Lunar landing via images: A Reinforcement Meta-Learning application to autonomous hazard avoidance and landing,” *AAS/AIAA Astrodynamics Specialist Conference 2020*, Vol. 175, Univelt Inc., San Diego, CA, 2020, pp. 91–110.
- [9] Iiyama, K., Tomita, K., Bhavi A. Jagatiaz and, T. N., and Ho, K., “Deep Reinforcement Learning for safe landing site selection with concurrent consideration of divert maneuvers,” *AAS/AIAA Astrodynamics Specialist Conference 2020*, Vol. 175, Univelt Inc., San Diego, CA, 2020, pp. 111–130.
- [10] Caroselli, E., Belien, F., Falke, A., Curti, F., and Forstner, R., “Deep learning-based passive hazard detection for asteroid landing in unexplored environment,” *44th AAS GN&C conference, Colorado, Breckenridge, 2022*, pp. 1–16.
- [11] Palafox, L. F., Hamilton, C. W., Scheidt, S. P., and Alvarez, A. M., “Automated detection of geological landforms on Mars using Convolutional Neural Networks,” *Computers & Geosciences*, Vol. 101, 2017, pp. 48–56. <https://doi.org/https://doi.org/10.1016/j.cageo.2016.12.015>.
- [12] Mukhoti, J., and Gal, Y., “Evaluating Bayesian Deep Learning Methods for Semantic Segmentation,” , 2018. URL <http://arxiv.org/abs/1811.12709>, pre-print.
- [13] Tomita, K., Skinner, K. A., and Ho, K., “Uncertainty-Aware Deep Learning for Autonomous Safe Landing Site Selection,” , 2021. <https://doi.org/10.13140/RG.2.2.15224.98564>, URL <http://rgdoi.net/10.13140/RG.2.2.15224.98564>, pre-print.
- [14] Pugliatti, M., Maestrini, M., Lizia, P. D., and Topputo, F., “On-board small-body semantic segmentation based on morphological features with U-Net,” *31st Space Flight Mechanics Meeting, Charlotte, NC, 2021*, pp. 1–20.
- [15] Pugliatti, M., and Topputo, F., “Navigation about irregular bodies through segmentation maps,” *31st Space Flight Mechanics Meeting, Charlotte, NC, 2021*, pp. 1–20.
- [16] Barringer, D. M., “Further Notes on Meteor Crater, Arizona,” *Proceedings of the Academy of Natural Sciences of Philadelphia*, Vol. 66, No. 3, 1914, pp. 556–565. URL <https://www.jstor.org/stable/4063595>.
- [17] Canny, J., “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, pp. 679–698. <https://doi.org/https://doi.org/10.1109/TPAMI.1986.4767851>.
- [18] Shih, F., *Image Processing and Mathematical Morphology: Fundamentals and Applications*, CRC Press, 2011, Chap. 3, pp. 25–35. URL <https://doi.org/10.1201/9781420089448>.
- [19] Faraco, N., “Instance segmentation for features recognition on non-cooperative resident space objects,” Master’s thesis, Politecnico di Milano, 07 2020. URL <https://www.politesi.polimi.it/handle/10589/167418>.

- [20] Sharma, S., and D'Amico, S., "Pose Estimation for Non-Cooperative Rendezvous Using Neural Networks," *CoRR*, Vol. abs/1906.09868, 2019. URL <http://arxiv.org/abs/1906.09868>.
- [21] Yoshikawa, M., Kawaguchi, J., Fujiwara, A., and Tsuchiyama, A., "Hayabusa sample return mission," *Asteroids IV*, Vol. 1, 2015, pp. 397–418. [https://doi.org/10.2458/azu\\_uapress\\_9780816532131-ch021](https://doi.org/10.2458/azu_uapress_9780816532131-ch021).
- [22] 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. <https://doi.org/10.1007/s11214-017-0377-1>.
- [23] 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. <https://doi.org/10.1007/s11214-017-0405-1>.
- [24] Russell, C. T., and Raymond, C. A., *The Dawn Mission to Vesta and Ceres*, Springer New York, New York, NY, 2012, Chap. The Dawn Mission to Vesta and Ceres, pp. 3–23. [https://doi.org/10.1007/978-1-4614-4903-4\\_2](https://doi.org/10.1007/978-1-4614-4903-4_2), URL [https://doi.org/10.1007/978-1-4614-4903-4\\_2](https://doi.org/10.1007/978-1-4614-4903-4_2).
- [25] Prockter, L., Murchie, S., Cheng, A., Krimigis, S., Farquhar, R., Santo, A., and Trombka, J., "The NEAR shoemaker mission to asteroid 433 eros," *Acta Astronautica*, Vol. 51, No. 1, 2002, pp. 491–500. [https://doi.org/https://doi.org/10.1016/S0094-5765\(02\)00098-X](https://doi.org/https://doi.org/10.1016/S0094-5765(02)00098-X), URL <https://www.sciencedirect.com/science/article/pii/S009457650200098X>.
- [26] Shi, J.-F., Ulrich, S., and Ruel, S., "CubeSat Simulation and Detection using Monocular Camera Images and Convolutional Neural Networks," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, pp. 1–23. <https://doi.org/10.2514/6.2018-1604>.
- [27] 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 2020*, Vol. 175, Univelt Inc., San Diego, CA, 2020, pp. 2539–2558.
- [28] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. <https://doi.org/https://doi.org/10.1109/CVPR.2018.00474>.
- [29] Teil, T. F., "Optical Navigation using Near Celestial Bodies for Spacecraft Autonomy," Ph.D. thesis, University of Colorado Boulder, 2020.

## Appendix - Networks architecture and training

The architectures used in this work are reported in this section ( using the Tensorflow 2.5.0 notation) together with their hyper-parameters, training, and validation performance.

Tab. 8 illustrates the architecture of the CNN used in the classification task to train the UNet encoder, represented by its convolutional portion.

**Table 8 Architecture of the CNN considered in this work. The total number of parameters is 1,474,951, all of which are trainable.**

Layer name	Layer type	Output Shape	Param #
I	InputLayer	(None, 128, 128, 1)	0
C1	Conv2D	(None, 128, 128, 16)	160
LR1	LeakyReLU	(None, 128, 128, 16)	0
P1	MaxPooling2D	(None, 64, 64, 32)	0
C2	Conv2D	(None, 64, 64, 32)	4640
LR2	LeakyReLU	(None, 64, 64, 32)	0
P2	MaxPooling2D	(None, 32, 32, 64)	0
C3	Conv2D	(None, 32, 32, 64)	18496
LR3	LeakyReLU	(None, 32, 32, 64)	0
P3	MaxPooling2D	(None, 16, 16, 128)	0
C4	Conv2D	(None, 16, 16, 128)	73856
LR4	LeakyReLU	(None, 16, 16, 128)	0
P4	MaxPooling2D	(None, 8, 8, 256)	0
C5	Conv2D	(None, 8, 8, 256)	295168
LR5	LeakyReLU	(None, 8, 8, 256)	0
P5	MaxPooling2D	(None, 4, 4, 256)	0
DO1	Dropout	(None, 4, 4, 256)	0
F	Flattern	(None, 4096)	0
D1	Dense	(None, 256)	1048832
DO2	Dropout	(None,256)	0
D2	Dense	(None, 128)	32896
O	Output	(None,7)	903

The CNN’s architecture in Tab. 8 is divided into four portions, respectively from top to bottom: input, convolutional layers, neural networks layers, and output. The results of convolution and activation at each depth of the CNN are copied in the encoding layers of the UNet while stacked in the decoding layers of the same. These are referred to as  $E_i$  layers, where  $i$  indicate their reference depth.

The architecture in Tab. 9 is divided into 5 portions, from top to bottom: input, encoder, decoder, head, and output. The encoder is constituted by the frozen convolution layers of the CNN architecture in Tab. 8 while the decoder is generated by stacking such layers with new upsampling layers taken from the pix2pix\*\* architecture in TensorFlow (TF). Note that  $UNet_S$ ,  $UNet_S^A$ , and  $UNet_R$  share the same architecture described in Tab. 9, with different sets of weights and biases as consequence of different training strategies. Finally, the hyper-parameters of both types of architectures are illustrated in Tab. 10 and Tab. 11. Note that a thorough hyper-parameter search based on a refined grid-search approach has been used to find out the best set of parameters defining the networks. The LeakyReLU is used in all the convolutional layers of the CNN and UNet, the ReLU is used in all the layers of the neural network portion of the CNN apart from the last one, the output layer, which uses the *softmax* activation function.

\*\*<https://www.tensorflow.org/tutorials/generative/pix2pix>, last accessed on 15th of March 2022.

**Table 9 Architecture of the UNet considered in this work. The total number of parameters is 1,225,413, 832,613 of which are trainable and 392,320 are not trainable.**

Layer name	Layer type	Output Shape	Param #
I	InputLayer	(None, 128, 128, 1)	0
E1	Encoder	(None, 128, 128, 16)	160
E2	Encoder	(None, 64, 64, 32)	4640
E3	Encoder	(None, 32, 32, 64)	18496
E4	Encoder	(None, 16, 16, 128)	73856
E5	Encoder	(None, 8, 8, 256)	295168
UP1	Sequential	(None, 16, 16, 128)	295424
CC1	Concatenate	(None, 16, 16, 256)	0
DO1	Dropout	(None, 16, 16, 256)	0
LR1	LeakyReLU	(None, 16, 16, 256)	0
UP3	Sequential	(None, 32, 32, 64)	147712
CC2	Concatenate	(None, 32, 32, 128)	0
DO2	Dropout	(None, 32, 32, 128)	0
LR2	LeakyReLU	(None, 32, 32, 128)	0
UP3	Sequential	(None, 64, 64, 32)	36992
CC3	Concatenate	(None, 64, 64, 64)	0
DO3	Dropout	(None, 64, 64, 64)	0
LR3	LeakyReLU	(None, 64, 64, 64)	0
UP4	Sequential	(None, 128, 128, 16)	9280
CC4	Concatenate	(None, 128, 128, 32)	0
DO4	Dropout	(None, 128, 128, 64)	0
LR4	LeakyReLU	(None, 128, 128, 128)	0
CT1	Conv2DTranspose	(None, 128, 128, 128)	36992
DO5	Dropout	(None, 128, 128, 128)	0
CT2	Conv2DTranspose	(None, 128, 128, 256)	295168
DO5	Dropout	(None, 128, 128, 256)	0
O	Conv2DTranspose	(None, 128, 128, 5)	11525

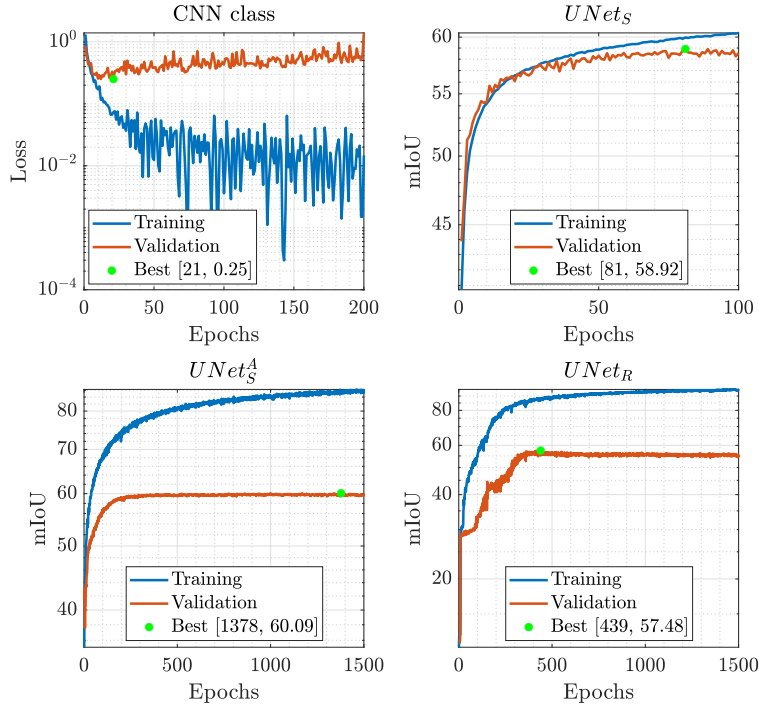
**Table 10 Hyper-parameters of the CNN.**

Parameter	Value
Batch size	200
Optimizer	Adam
Activation function	ReLU, LeakyReLU, and Softmax
$\alpha$ parameter of the LeakyReLU	0.3
Convolution kernel size	3x3
Pooling kernel size	2x2
Dropout value (DO1)	0.2
Dropout value (DO2)	0.2
Loss metric	SCCE
Accuracy metric	accuracy
Epochs	100

**Table 11 Hyper-parameters of the UNet.**

Parameter	UNet <sub>S</sub>	UNet <sub>S</sub> <sup>A</sup>	UNet <sub>R</sub>
Batch size	50	70	70
Optimizer	Adam		
Activation function	ReLU, LeakyReLU		
$\alpha$ parameter of the LeakyReLU	0.3		
Convolution kernel size	3x3		
Pooling kernel size	2x2		
Dropout values (DO1-DO4)	0.2	0.6	0.1
Dropout values (DO5-DO6)	0.4	0.6	0.1
Weight Background	0.09	0.09	0.09
Weight Surface	0.09	0.09	0.09
Weight Craters	0.45	0.35	0.35
Weight Boulders	0.23	0.33	0.33
Weight Terminator	0.14	0.14	0.14
Loss metric	WSCCE		
Accuracy metric	mIoU		
Epochs	100	1500	1500
Learning rate	$1.3 e^{-3}$	$1.0 e^{-6}$	$1.0 e^{-3}$

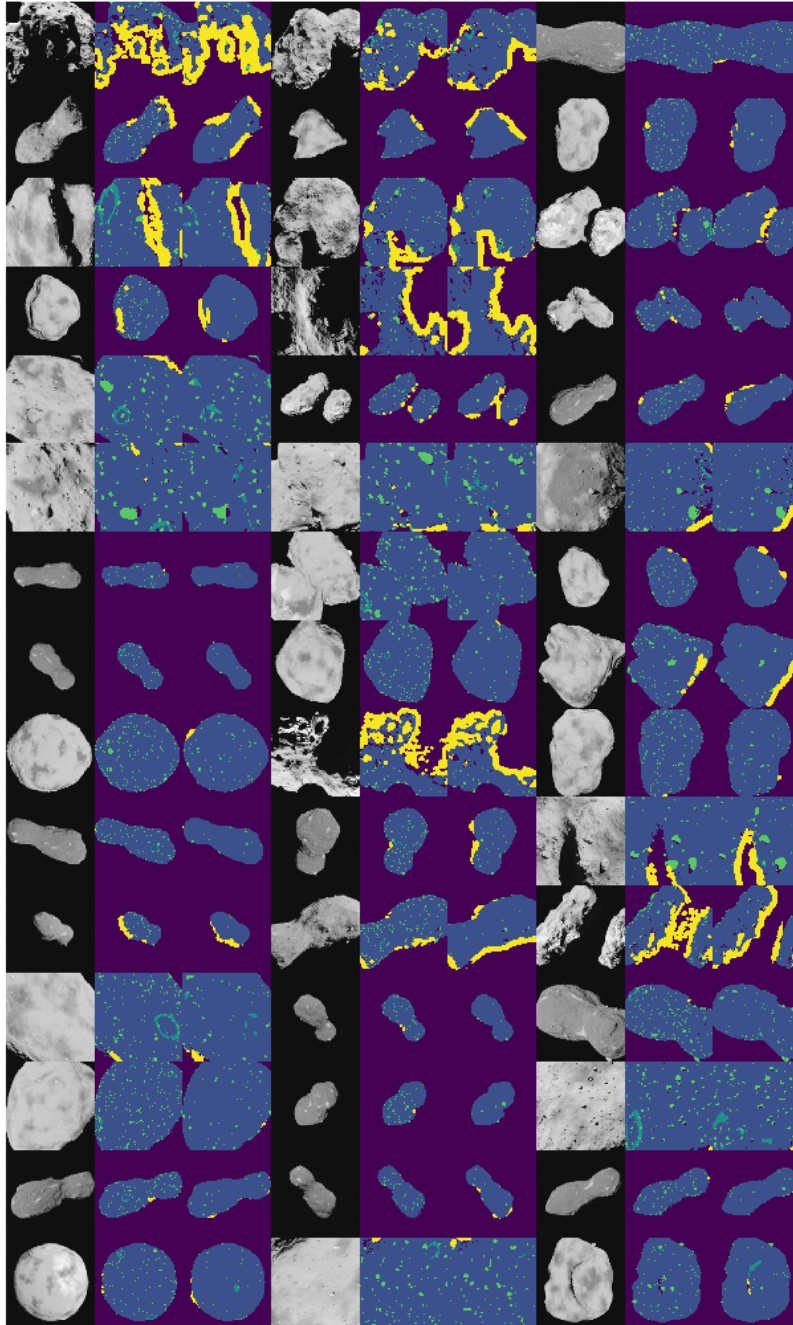
Fig. 22 shows training and validation curves of the four networks used in inference in this work. Note that each architecture is instantiated with the set of weights and biases for inference at the epoch at which its validation curve reaches its minimum (CNN) or maximum (UNet) value.



**Fig. 22 Training and Validation curves of the architectures used in this work.**

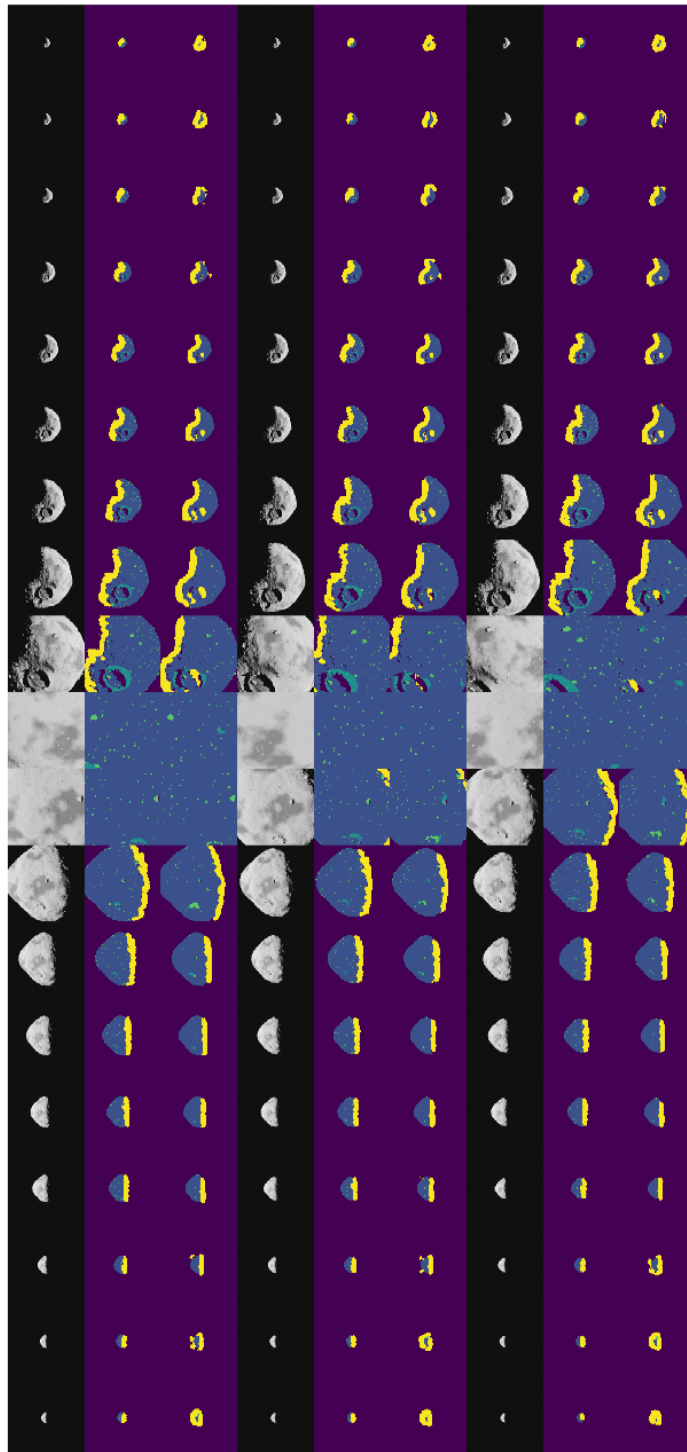
## Appendix - Mosaic views

This appendix contains mosaic views of the prediction of various UNet architectures in different datasets.

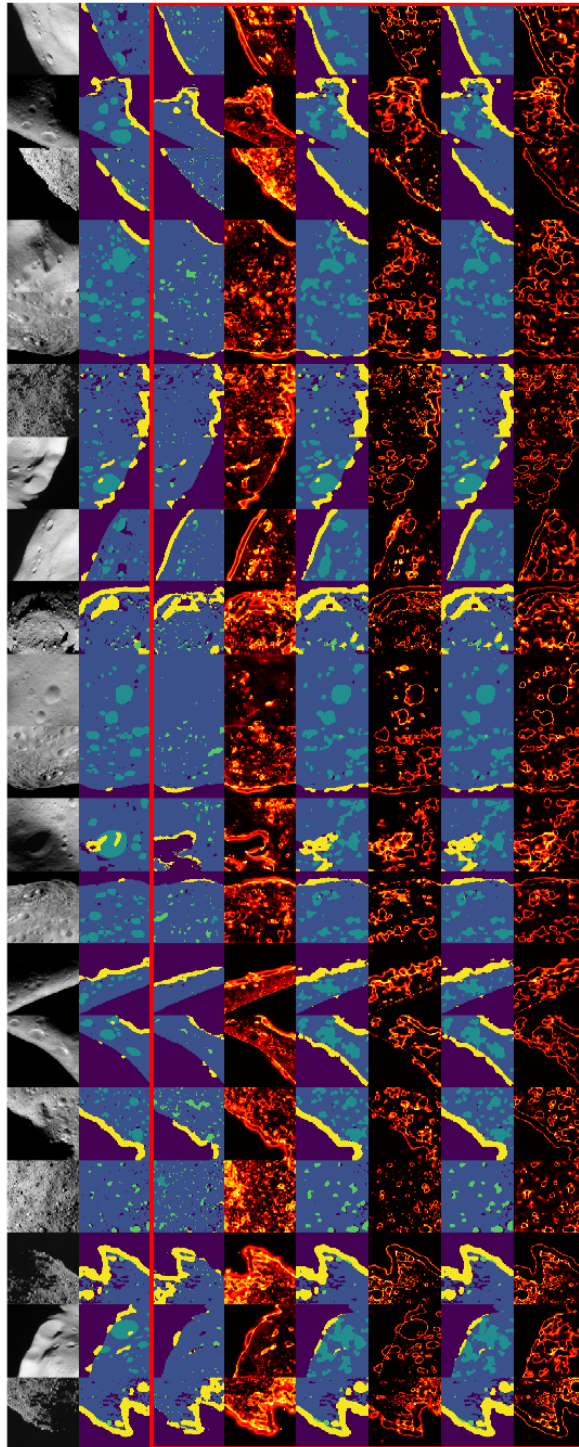


**Fig. 23** Mosaic of  $15 \times 3$  triplets showing input (left), true mask (center) and predicted mask (right) for test dataset of D-1 by UNets.





**Fig. 24** Mosaic of  $19 \times 3$  triplets showing input (left), true mask (center) and predicted mask (right) for test dataset of D-3 by UNets.



**Fig. 25** Mosaic of 20 samples from the test set of D-4 showing input (1st column), true mask (2nd column) together with predicted mask and uncertainty map pairs of UNet<sub>S</sub>, UNet<sub>S</sub><sup>A</sup>, and UNet<sub>R</sub> (from left to right in the red rectangle).