



A LiDAR-less approach to autonomous hazard detection and avoidance systems based on semantic segmentation

Pelayo Peñarroya^{1,2} · Simone Centuori¹ · Manuel Sanjurjo² · Pablo Hermosín¹

Received: 14 November 2022 / Revised: 7 February 2023 / Accepted: 31 March 2023 /
Published online: 30 May 2023
© The Author(s) 2023

Abstract

In this paper, a passive hazard detection and avoidance (HDA) system is presented, relying only on images as observations. To process these images, convolutional neural networks (CNNs) are used to perform semantic segmentation and identify hazards corresponding to three different layers, namely feature detection, shadow detection, and slope estimation. The absence of active sensors such as light detection and ranging (LiDAR) makes it challenging to assess the surface geometry of a celestial body, and the training of the neural networks in this work is oriented towards coping with that drawback. The image data set for the training is generated using blender, and different body shape models (also referred to as meshes) are included, onto which stochastic feature populations and illumination conditions are imposed to produce a more diverse database. The CNNs are trained following a transfer learning approach to reduce the training effort and take advantage of previously trained networks. The results accurately predict the hazards in the images belonging to the data set, but struggle to yield successful predictions for the slope estimation, when images external to the data set are used, indicating that including the geometry of the target body in the training phase makes an impact on the quality of these predictions. The obtained predictions are composed to create

Simone Centuori, Manuel Sanjurjo and Pablo Hermosín have equally contributed to this work.

This article is part of the topical collection on Machine Learning in Celestial Mechanics and Dynamical Astronomy.

Guest Editors: Massimiliano Vasile, Xiyun Hou, Roberto Furfaro and Alessandra Celletti.

✉ Pelayo Peñarroya
pelayo.penarroya@deimos-space.com

Simone Centuori
simone.centuori@deimos-space.com

Manuel Sanjurjo
msanjurj@ing.uc3m.es

Pablo Hermosín
pablo.hermosin@deimos-space.com

¹ Mission Analysis and Navigation, Deimos Space S.L.U., Ronda de Poniente 19, 28760 Tres Cantos, Madrid, Spain

² Aerospace Engineering, Universidad Carlos III de Madrid, Avenida Universidad 30, 28911 Leganés, Madrid, Spain

safety maps, which are meant to be given as input to the guidance block of the spacecraft to evaluate the need for a manoeuvre to avoid hazardous areas. Additionally, preliminary hardware-in-the-loop (HIL) test results are included, in which the algorithms developed are confronted against images taken using real hardware.

Keywords Hazard detection and avoidance · Autonomous navigation · Asteroids · Image processing · Semantic segmentation

1 Introduction

Landing on small celestial bodies has become one of the most important challenges in the pursuit of space exploration and exploitation. Some missions in the past years have achieved huge milestones towards this goal, by taking advantage of different strategies. The Rosetta mission (Schwehm and Schulz 1999; Glassmeier et al. 2007) deployed a lander, Boehnhardt et al. (2017), whose objective was to obtain information about the comet's gravitational field and surface properties. Kawaguchi et al. (2008), OSIRIS-REx Lauretta et al. (2017), and Watanabe et al. (2017) used a different approach and performed Touch-and-Go (TaG) manoeuvres, which consist of a controlled touch-down on the surface of the celestial body, followed immediately by a powered ascent.

Interacting with the surface of the body of interest has, thus, gained prominence in the mission design and analysis component of the current space exploration activities, even if it poses some of the most challenging tasks. Landing sequences require very high degrees of autonomy due to the high latency with ground stations on Earth, so it is key to be able to rectify the original instructions given to the spacecraft based on the inputs that its on-board sensors receive.

One of the systems in charge of this is the HDA system, whose typical tasks include (but are not limited to) shadow detection, feature detection, slope estimation, or surface roughness estimation. Two main types of HDA systems can be distinguished: passive and active. Passive systems rely only on camera inputs, while active systems use, additionally, LiDAR. The difference lies in the way to interact with the environment. LiDARs, for instance, emit a beam that is then registered in the sensor once it bounces and comes back. This is considered an active sensor because it registers an observation originally sourced from the sensor itself. Passive systems, on the other hand, entail sensors that only receive some kind of input observation, without proactively emitting any kind of interaction with the environment. Cameras are passive sensors because they only take whatever information is available in the environment they work in.

Some of the earliest works on HDA systems were presented by Epp and Smith (2007); Johnson et al. (2008), where the use of active HDA systems for landing on the lunar surface is discussed and analysed. There, different parametric analyses were performed including sensor performance, trajectory angle, or vehicle hazard tolerance. Although the use of passive systems has also been discussed previously by Huertas et al. (2006), it lacked performance when it came to the estimation of the slopes, as was also concluded in the posterior analysis performed by Neveu et al. (2015).

Having access to the digital elevation maps (DEMs) generated by LiDAR sensors makes the slope and roughness estimation processes much easier and accurate, but shapes the mission design around these active sensors that could weigh up to 25kg and require up to 200W of power (Huertas et al. 2006). In addition, they offer very restrictive field-of-view (FoV) angles

and resolutions and take considerable volume inside the spacecraft's bus. Cameras, on the other hand, offer much more flexibility from a mission design perspective in terms of mass, volume, and power consumption, which is key for missions whose scientific return is highly valuable.

All of the aforementioned pieces of work related to HDA relied on traditional image processing (IP) techniques, which pose the additional challenge of having to adapt computationally expensive techniques to the limited space-proven hardware available for space missions. It was only recently that the use of artificial intelligence (AI) techniques (in particular artificial neural networks (ANNs)) was proposed to tackle this problem (Lunghi et al. 2015), because of their great generalisation properties, which become very relevant in problems where it is impossible to determine in advance the different hazards present in the terminal landing area.

Additionally, ANN techniques are highly computationally efficient due to their working principle being based on polynomial operations, which is a desirable feature when an on-board implementation is considered. In Lunghi et al. (2016), two different architectures are compared: multilayer feedforward and cascade. In that work, information about the altitude and attitude of the spacecraft are also provided as additional input to the HDA algorithm that, then, generates the hazard map.

Following the same research line, in Silburt et al. (2019), a CNN was trained to recognise and count lunar craters. The networks also showed promising results regarding the transferability of its predictions after testing it on a database with images corresponding to Mercury, instead of the Moon.

In Pasqualetto Cassinis et al. (2020); Sharma and D'Amico (2020), CNNs were trained to estimate the position and pose of a non-cooperative spacecraft, using only monocular vision observations. In Sharma and D'Amico (2020), in particular, the authors also detail how a database generation functionality needed to be developed, due to the scarcity of resources available for the training. In that same year, in Tomita et al. (2020), the superiority of CNN-based algorithms over traditional techniques was demonstrated.

More recently, Silvestrini et al. (2022); Pugliatti et al. (2022) have shown how these networks can be used to perform orbital navigation using lunar craters and binary asteroid system geometry, respectively. The authors show how the great generalisation capabilities of such algorithms greatly improve the quality of the predictions made by the networks. The applicability of these algorithms to landing sequences and hazard detection is discussed and, in Pugliatti et al. (2022), a U-shaped network is developed and trained to detect features on small celestial bodies, using an ad hoc data set generation tool.

Because of the benefits they offer, CNNs have been used in many different projects in the last few years. However, one of the main drawbacks of deep learning (DL) architectures is that they require a relatively large amount of data to be trained (Al-Moosawi et al. 2021; Gao et al. 2021). On top of that, since the training scheme follows a supervised approach, the labelling of the available data is an effort required to ensure the quality of the training. These two factors make it very expensive to properly train such a network from scratch.

Considering all the above, and in spite of the amount of research put towards HDA systems and IP techniques, very few researchers have tackled the problem of combining both fields. In particular, developing algorithms capable of processing passive observations and still accounting for the geometry or roughness of the surface of a celestial body is a task that still remains unexplored. AI-based passive HDA systems are a very challenging problem mainly because they involve the prediction of a 3D geometry from visual, 2D inputs. However, finding a solution would greatly benefit future space missions to planetary bodies by providing a low-

impact solution (in terms of both cost and mission design) that would increase the autonomy of the spacecraft and its capability to interact with the bodies it would explore.

Thus, in this work, passive HDA systems based on CNNs are developed and tested against different image data sets. CNNs offer the benefit of achieving high accuracies, at the cost of a more expensive training phase. Three separate layers are proposed for the complete system: shadow detection, feature detection, and slope estimation. The input to the algorithm is the raw image as it would be taken by a camera, without any additional information about the spacecraft's state. The algorithm developed under this research is named *astroHda* and has been included in the Astrodynamics Simulator (AstroSim) suite (Peñarroya et al. 2022). Unlike previously mentioned works, *astroHda* is intended to be capable of predicting hazards for any small body it could encounter, taking advantage of the generalisation capabilities that deep learning techniques such as CNN offer.

In the following, Sect. 2 talks about the architecture chosen for the neural networks and the techniques exploited for its development. After that, Sect. 3 introduces the followed training scheme and explains how the images used for the training and included in the data sets are generated. Section 4 shows the results obtained with the developed networks and analyses the different elements involved in the HDA system. It also explains how safety maps are created and includes network predictions for additionally rendered images not used for the training, and images taken by real sensors. Finally, Sect. 5 gathers some conclusions about the results obtained and discusses strengths and weaknesses of the developed techniques.

2 Architecture

CNNs are multilayer feedforward neural networks that apply image analysis filters as convolutions. This architecture is a type of DL method with the capability of very accurately classifying images and detecting patterns or other features in pictures. By stacking several layers, a pixel-by-pixel classification can be obtained as output. In each of these layers, multiple filters are convoluted over the input to the layer to obtain the so-called feature maps. Thus, CNNs are composed of mainly three types of layers:

1. Convolutional layers, where the image filters are applied in order to identify a certain pattern or feature.
2. Pooling layers, where the feature maps are reduced to their main attributes to reduce the computational cost while maintaining the main feature characteristics of the map.
3. Fully connected layers, which are responsible for the final classification and take into consideration all the previously stacked layers.

As it has been introduced in Sect. 1, training a DL network from zero could become quite expensive computationally. An alternative, proposed in Akçay et al. (2016), is transfer learning. This method is based on taking advantage of pre-trained networks as a baseline for the training of a neural network destined for a different task (or maybe a similar one). The weights obtained after training such a network are used as initial values for the training phase of a particular project. Some of the layers are initially frozen so that the first steps of the training do not modify the core hidden layers of the network. The reason why these core layers are frozen is that they contain the key feature maps that were previously learned by the network, which consist of more basic features that can be easily applied to many different image classification types. These key features include straight lines, circles, directionalities, or other basic geometric entities. On the other hand, the last layers of a network learn features that become much more case-specific, and are, thus, not frozen from the beginning to allow

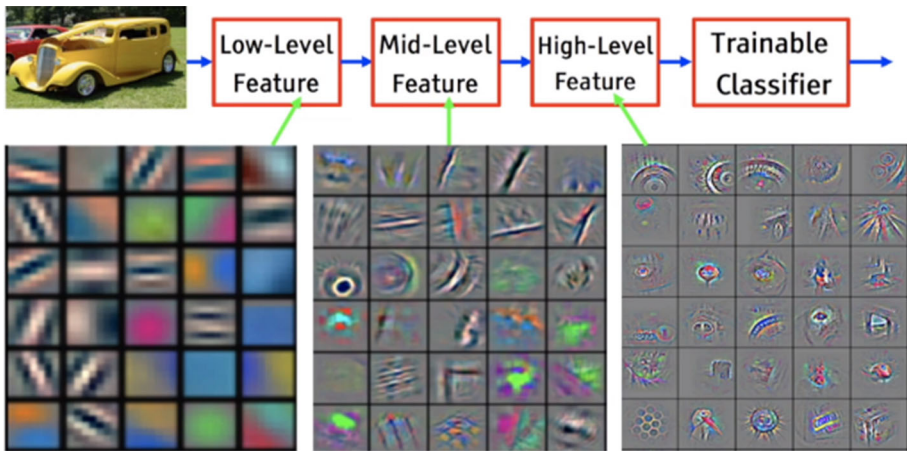


Fig. 1 Feature maps at different stages of a network. The feature maps at the initial layers encode edges and direction, i.e. horizontal or vertical lines, the feature maps obtained at the middle of the network visualise textures and patterns, and the feature maps at the last layers depict parts of objects and objects in the image (Vignesh 2020)

the pre-trained network to adapt to the new data set. For instance, a network that has been trained to differentiate and classify different object images, as the one shown in Liu et al. (2014), would offer more detailed features as the layers go deeper. Figure 1 shows how, for deeper layers, the detail and case specificity of the feature maps increase (Zeiler et al. 2014).

Making use of such a technique is paramount when training a neural network for space applications, where the amount of available real data is relatively limited and not pre-processed for supervised learning. The next section will explain how this problem is tackled using both transfer learning and a pipeline for the generation of an artificial data set for the specific study case of a landing sequence in an asteroid environment.

Another of the main problems of deep neural networks is what is called vanishing gradients. In plane architectures, it could happen that the derivatives for the weights of each layer are small and that, when multiplying them (following the chain rule) for the back-propagation, they become even smaller to a point where an update of the initial layers could lose its effectiveness. In He et al. (2015), the idea of using residual layers is proposed, where the vanishing gradients problem is minimised by using skip connections, where a number of convolutional layers are skipped, at every basic block of the network. Figure 2 shows how a skip connection looks like in a basic convolution block. The input, x_I , is bypassed downstream to the output of the convolution filter. This allows for much deeper architectures, without overloading the training with the computational effort of excess parameters.

Taking advantage of this structure, a novel architecture was proposed in He et al. (2015), which was named residual network (ResNet)-34, referring to the 34 layers it is composed of. (Table 1 gives a summary of the structure of the network.) The network was trained using the ImageNet data set (Deng et al. 2009), which consists of 1000 classes (therefore, the last layer of the network in Table 1), distributed around 1.28 million training images and 50.000 validation images. After more than 500.000 epochs of training, the model was found to considerably reduce the top-1 error (by 3.5%) with respect to its non-residual counterpart, due to the improvement in the degradation problem.

For the results shown in this work, ResNet-34 has been chosen as the backbone architecture, because, as has been shown above, it is very effective when applied to semantic

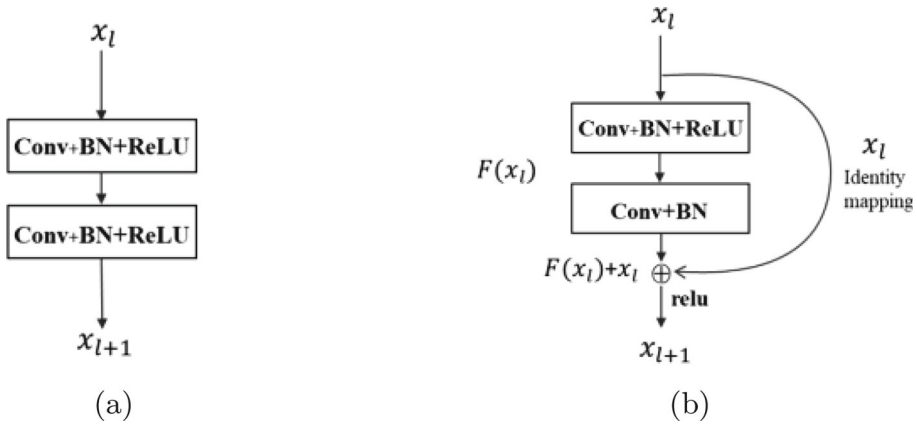


Fig. 2 Comparison between a traditional convolutional block (left) and a residual block (right)

Table 1 ResNet-34 architecture

Layer	Output size	34-Layer
Conv1	112×112	$7 \times 7, 64, stride2$ $3 \times 3 maxpool, stride2$
Conv2-x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$
Conv3-x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$
Conv4-x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$
Conv5-x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$
	1×1	Average pool, 1000-d, softmax

segmentation tasks. The higher depths that are achievable when using residual networks greatly benefit the capability of the network to detect certain patterns and shapes, without raising the computational burden of the training phase. Several ResNet architectures are available, the most-used ones ranging from 18 to 101. ResNet-34, which was introduced in Table 1, offers a great trade-off for the problem at stake and it is also available in the fastai pre-trained networks collection, which makes its implementation much simpler. The weights with which the network is initialised correspond to those obtained by training it with the ImageNet data set, as introduced in He et al. (2015). This transfer learning approach reduces the computational load needed to fit the network to the particular needs of the study case posed in this paper.

3 Data set generation and training

According to Ripley (2007); Ghilardi et al. (2022), CNN-based approaches require the assessment of the network in three stages:

- Training set: a set of examples used for learning, that is to fit the parameters of the classifier.

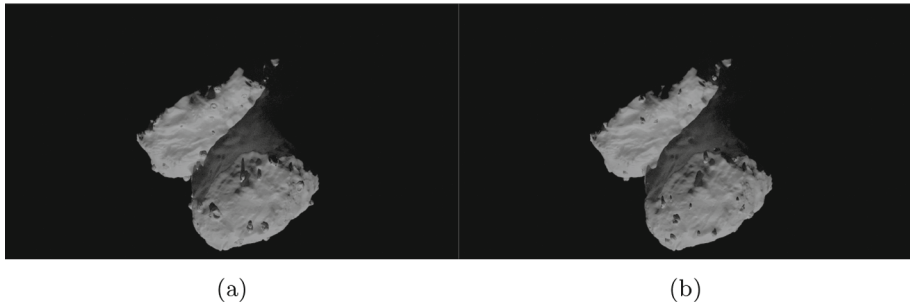


Fig. 3 Arbitrary rendered images for comet 67P/Churyumov-Gerasimenko generated using AstroSim. Notice how features (craters and boulders) have been randomly distributed around the surface of the comet to enrich the data set

- Validation set: a set of examples used to tune the parameters of a classifier, for example, to choose the number of hidden units in a neural network.
- Test set: a set of examples used only to assess the performance of a fully specified classifier.

Since the training scheme follows a supervised approach, the generation of ground truth images is fundamental for the creation of the database. The images used as input to the CNN should be the raw input as they would appear on the camera, i.e. the original mesh, with shadows and features (this will be referred to as *featured mesh*).

Thus, the ground truth information required for the *astroHda* database must include labelling about shadowed areas, feature location, and slope information. The former can be obtained by pixel subtraction between the featured body and the mesh observed from the same position and with the same orientation but once all of its features have been removed (this will be the *naked mesh*). For the latter, i.e. the slope estimation, the slope at each facelet of the mesh is computed, as depicted in Fig. 4, using the local gravity vector as a vertical reference for the computation of the slope. To compute the gravity vector, and exploiting the fact that a shape model is available, AstroSim is used to compute the gravitational potential at the centre of each facet using a polyhedral model-based gravity model, implemented following (Werner 1994). Then, the slope estimation can be tackled in two different manners: with a regression algorithm or a classification method.

The first option implies predicting the numerical value of the slope at each pixel corresponding to the body's surface. Since the pixels need to be ultimately translated into "safe" or "dangerous" areas, a simplified approach was chosen, where a threshold is defined to categorise facelets between safe and unsafe. Transforming these regression problems into a classification problem also allows for the use of a similar neural network structure to the one used for shadow or feature detection.

The slope threshold was set to 15° in this work based on Wang et al. (2020). Then, the equivalent image for the featured mesh is generated colouring the mesh accordingly (this will be called *sloped mesh*).

Images for the training database were generated using four different body meshes, one crater model, and three boulder models in three random combinations and distributions. To render the images, ten different orbits were simulated for each of the obtained bodies (the naked mesh, the three featured bodies, and the sloped mesh) and twenty pictures were taken from the spacecraft using a nadir-pointing attitude profile.

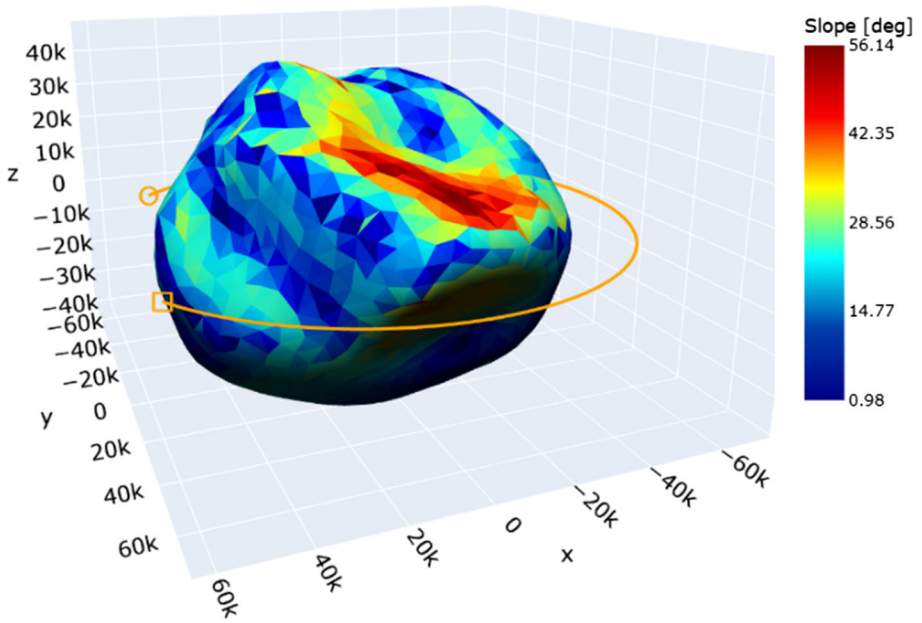
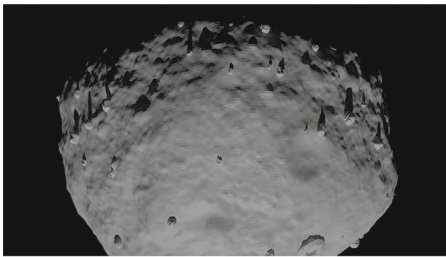
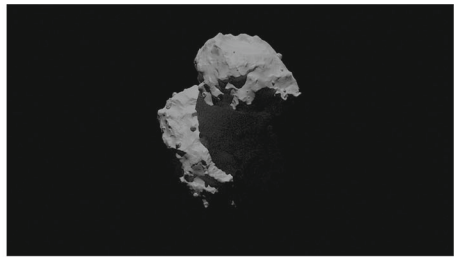


Fig. 4 Example of the slope computation result for asteroid Lutetia. Around the body, an arbitrary orbit is displayed in orange



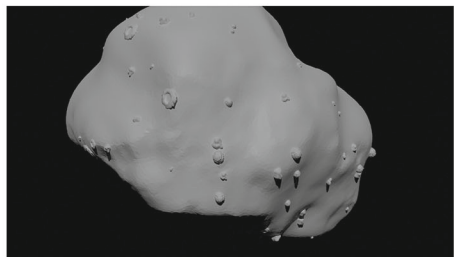
(a) Bennu.



(b) 67P/Churyumov-Gerasimenko.



(c) Eros.



(d) Lutetia.

Fig. 5 Examples of the different base meshes and feature distributions for the training and validation data set

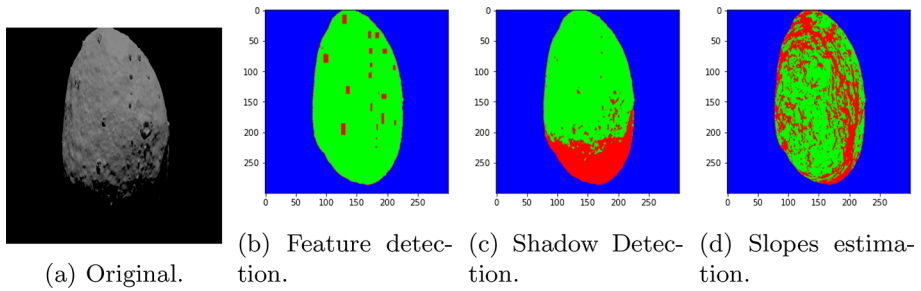


Fig. 6 Examples of the masks generated for an arbitrary rendering of Benu. Blue represents deep space, red represents dangerous areas, and green represents safe areas

Table 2 Label meaning for each of the layers developed for *astroHda*

	Shadow detection	Feature detection	Slope estimation	Colour
Safe area	Illuminated	Non-featured	Slope $< 15^\circ$	Green
Hazard	Shadowed	Featured	Slope $\geq 15^\circ$	Red
Deep space				Blue
Undetected				Black

The resulting set contains 2000 images that are then processed to generate the masks used as ground truth for the semantic segmentation training, using the pipeline described before. Examples of the different meshes used (with overlaid feature distributions) can be observed in Fig. 5. The corresponding masks for a rendering of Benu from the training set are also depicted in Fig. 6, where the labels for deep space, danger, and safe areas are depicted. Table 2 shows the different labels used and their colour codes.

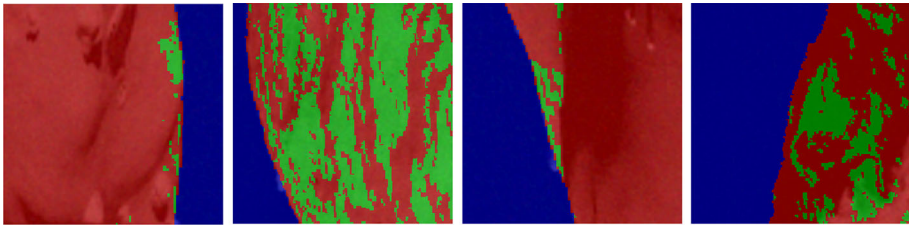
This training data set is divided into training and validation sets (as described at the beginning of Sect. 2) using a random split of 80–20, respectively. This means that 80% of the data set is being used to compute the optimal change in the internal weights of the network, and 20% as online cross-check to estimate whether the network is fitting too closely to the images it is being trained on. The rendered images are downsized to a 300x300 pixel resolution, and they are normalised using the parameters obtained from ImageNet (He et al. 2015). The reason is that the initial weights for the architectures used are extracted from the ImageNet data set, so the normalisation is also performed accordingly, exploiting transfer learning (Akçay et al. 2016) as a way to more efficiently train the networks developed for this work. Additionally, to increase the robustness of the training, the augmentations collected in Table 3 are implemented on the data set, and an exemplification of the resulting images is shown in Fig. 7, for the slope estimation layer. There, it becomes clear how the scaling for the image or the granularity added by the inclusion of the Gaussian noise impact the rendered images.

The accuracy metric used for the training relies on the dice loss from fastai Howard (2018), which builds on the traditional Intersection-over-Union methods. The optimiser used is Ranger, which is a combination of RAdam and Lookahead optimisers (Wright et al. 2021). As for the loss function, FlattenedLoss of CrossEntropyLoss is utilised, following what is common practice in semantic segmentation problems.

The training scheme, as mentioned before, is based on a transfer learning approach, where the network weights from ResNet-34 (used as the backbone) for ImageNet are taken as an

Table 3 Augmentation parameters for training data set

Random flip	True
Max rotation angle	10.0 deg
Min/Max zoom	1.0 / 5.0
Max lightning change	0.2
Max warping	0.2
Probability of affine transformation	0.75
Probability of brightness change	0.75
Gaussian noise mean/ σ	0.0 / 10.0

**Fig. 7** Example of the augmentations applied to the training set, for the slope estimation layer. In the images shown, the label colouring is superimposed over the original image

initial guess. A first stage of 10 epochs is performed, where only the last layer of the network is available for tuning, followed by a second phase of 12 epochs, where all the weights are unfrozen and freed to be modified by the optimiser to achieve a better fit to the data set used for this development.

The obtained networks are then tested using a new, smaller, data set, which is generated using the same pipeline described before, but with four new base meshes and features. This set consists of 100 images, and its role is to check the accuracy of the developed functionalities when used outside the data set they have been trained with. Examples of the used renderings can be observed in Fig. 8.

4 Results

The networks are tested on each of the aforementioned data sets to evaluate the predictions they are capable of doing. To understand how well the developed algorithms can be transferred to a HIL scenario, the networks are tested on a number of real-world images, even if the training data set included none of this type of images.

4.1 Shadow detection

Shadow detection was initially implemented following the Multi-Task Mean Teacher (MTMT) approach proposed by Chen et al. (2020) without further training. In their paper, they implement an MTMT training scheme where a CNN is trained to perform three different tasks simultaneously, namely, shadow count, shadow detection, and edge detection. Representative results of the obtained predictions can be observed in Fig. 9. The network was

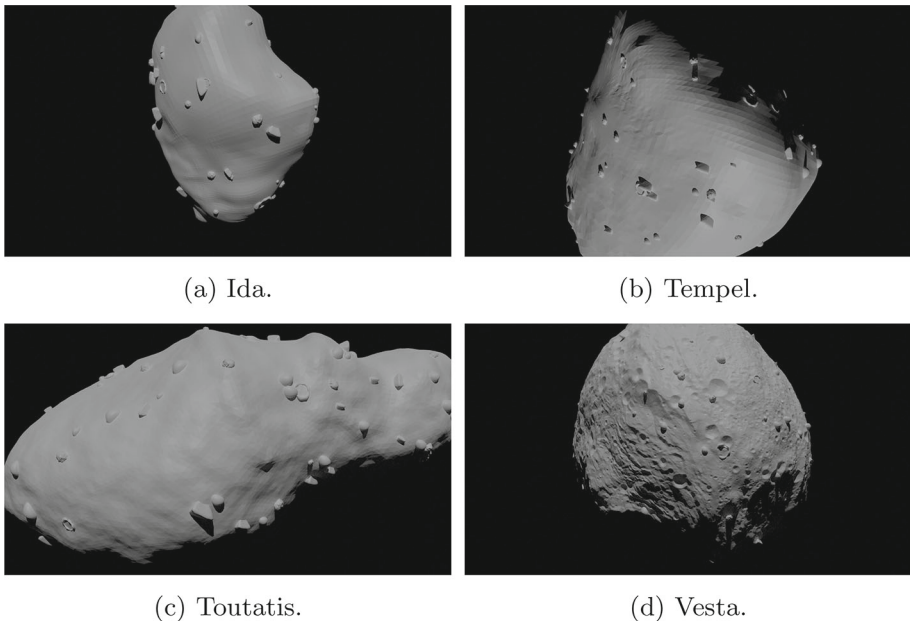


Fig. 8 Examples of the different base meshes and feature distributions for the test data set

trained using real-world images, and, in the results obtained, it can be seen how the prediction struggles to find a limit between shadowed and deep-space areas.

Trying to improve the results obtained using the MTMT network, a new network scheme (based on the approach introduced in Sect. 2) was developed. The architecture described in Table 1 was adopted, only with slight modifications to the last layer so that the output labels would be reduced to four labels (as introduced in Table 2): illuminated, shadowed, deep space, or undetected.

The labelling methodology chosen was to have a pixel-per-pixel label, as it is typically done in semantic segmentation tasks, where the last layers of the network are filtered through a softmax layer which assigns a label to each image coordinate. In the particular case of the shadow detection algorithm, safe areas correspond to zones where the illumination conditions allow the spacecraft to see the surface in detail, hazard to those shadowed or in weak illumination conditions, deep space corresponds to the image pixels outside the limb of the body, and a fourth label is given for those pixels, which the algorithm cannot classify as any of the previous types. Table 2 gathers the meaning of the different labels for each of the layers developed in this work.

Following the training scheme introduced in Sect. 3, the predictions shown in Fig. 10 are obtained. Its left-hand side panel shows two examples of the shadow predictions the algorithm is able to perform. The predicted shadows match very accurately the target (i.e. the ground truth), even in zones where the shape of the shadowed areas becomes highly irregular or even around features. On the other hand, the upper example on that figure shows how, when shadows are cast on the body limbs, it becomes difficult for the CNN to successfully distinguish between deep space and shadowed area of the body. In fact, when looking at the right-side panel, where the worst three predictions are gathered, it is clear that most of the

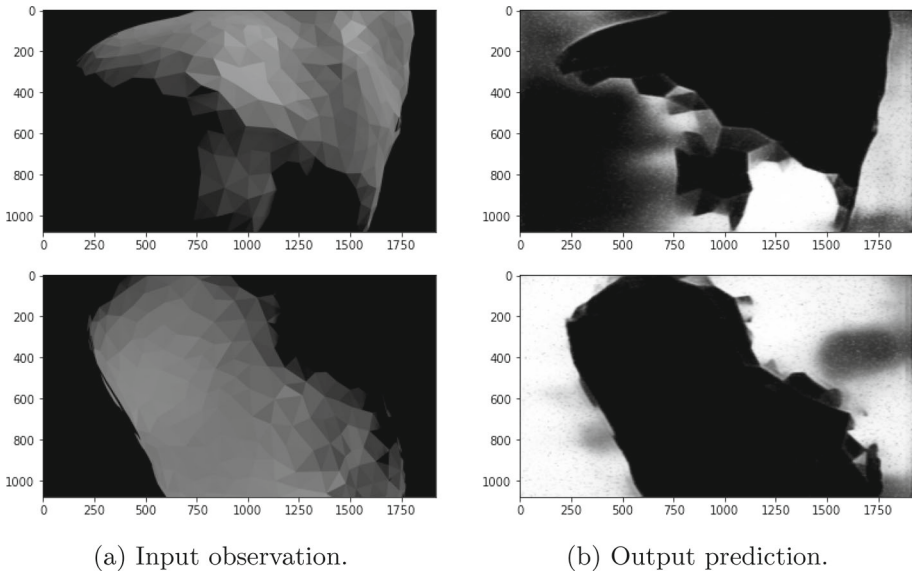


Fig. 9 Early shadow detection examples based on the MTMT method by Chen et al. (2020). Input is an image rendered using *astroRender*. Output is a binary map with ones (white pixels) where shadows are detected

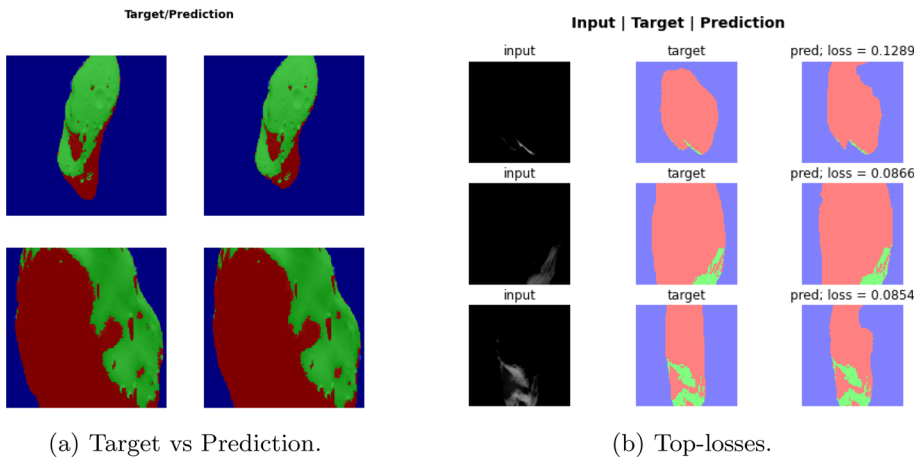


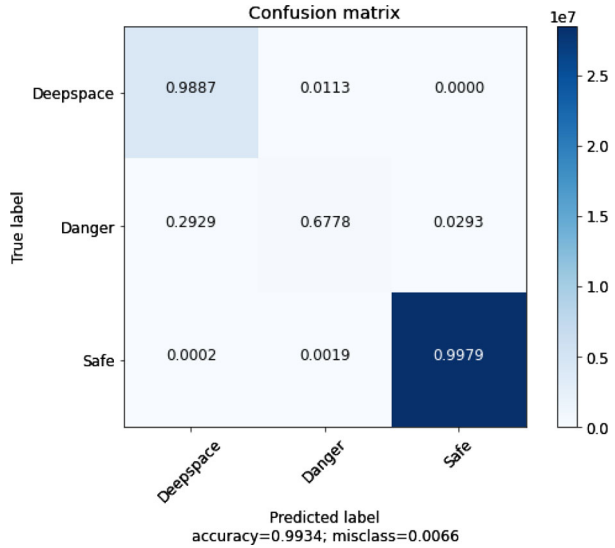
Fig. 10 Shadow detection example. Inputs and masks are overlaid. Labels map danger areas in red, safe areas in green, and deep space in blue

loss value these predictions yield comes from the inaccurate detection of the limit between shadowed area and deep space.

A more complete overview of the results obtained for the whole data set is provided by the confusion matrix shown in Fig. 11. There, the predicted labels for each of the images in the data set are compared to the true labels, as described by the masks. Four categories can be identified, namely true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These values are computed for each different category. For the shadow detection case, deep space and safe labels are very accurately predicted, and shadow detection

Fig. 11 Confusion matrix for the shadow detection layer.

Performances are normalised with respect to the total number of true labels for each category, indicated by the colorbar on the right-hand side of the figure



achieves 67.78% rate for TPs. The rest of the labels predicted as shadowed areas correspond mostly to deep-space predictions (the 29.29%), and only 2.93% of the cases were labelled as safe areas.

After close inspection of the generated masks for the training, it was observed that, when detecting the limb of the body in the original images (using thresholding and pixel subtraction techniques), some of the pixels around the limbs of the body are labelled as danger. This can be better observed in Fig. 14, where an arbitrary mask is displayed, together with a more in-detail view of the labels used for each pixel. It can be observed how a number of danger pixels close to the limbs of the body are actually wrongly labelled by the pre-processing and mask generation algorithm.

Nevertheless, the predictions obtained using this layer are capable of correcting this labelling mistakes and actually label these mistaken TNs correctly as deep space. On top of that, both deep space and shadowed labels correspond to hazards in the definitions used for the algorithms developed in this work, so the final set of hazards can be composed from the combination of these two sets. According to Luo et al. (2020), traditional methods for shadow estimation present performances of around 80%, while similar CNN-based approaches could go up to 91.79% (in particular, the network proposed there). In the results presented here, deep space and danger predictions (all the non-safe conditions) together, both represent 97.07% of the classified pixels, which is a very high success rate for shadow detection.

4.2 Feature detection

Following the results obtained for the shadow detection layers, and given the similarity of the study case, the same network architecture was chosen for the feature detection layer. In this case, safe areas correspond to pixels in absence of any boulder, crater, or feature that could represent a problem for the landing; hazards correspond to the opposite, and deep space and undetected keep the same meaning as for the shadow detection layer.

Figure 12 shows the results obtained after the training of the network. The left-hand side picture shows the results on a couple of target images. There, it can be observed how shadowed areas do not trigger any feature detection, since they are not visible and the target was not set

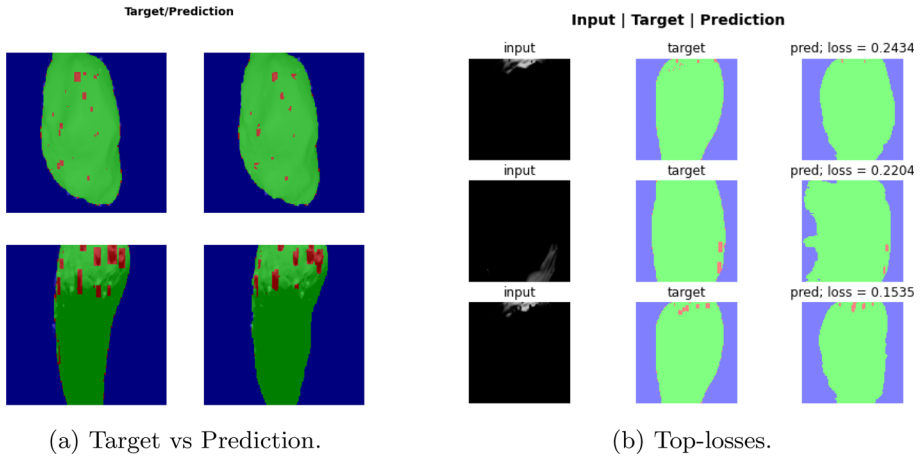
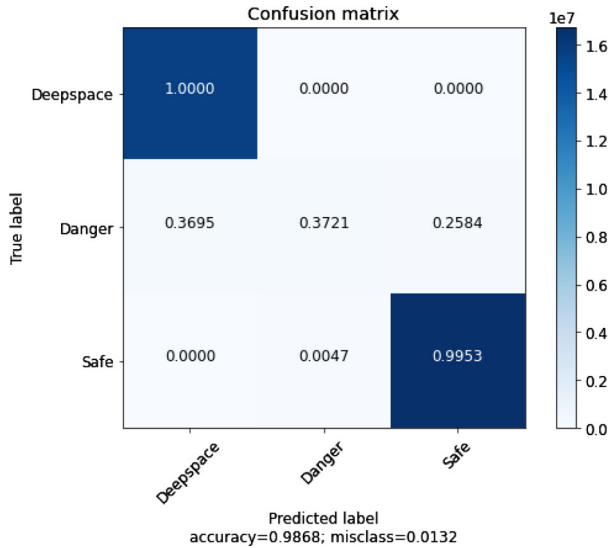


Fig. 12 Feature detection example. Inputs and masks are overlaid. Labels map danger areas in red, safe areas in green, and deep space in blue

Fig. 13 Confusion matrix for the feature detection layer. Performances are normalised with respect to the total number of true labels for each category, indicated by the colorbar on the right-hand side of the figure



to train the network to detect those features in those areas. The detection of the illuminated sections of the asteroid is very accurate and even achieves more natural shapes for the features detected than the original mask, which has enclosing rectangles instead.

The right-hand image shows the top-losses for the network, i.e. the input images that have the worst performances w.r.t. the target mask. All of these cases are images with very poor illumination conditions where not even the limb of the body can be accurately inferred. These results are in line with what was expected and would align well with the shadow detection layer that would discard those areas nevertheless.

In Fig. 13, statistic results for the training data set are shown. There it can be observed how, similarly to the shadow detection layer, deep space and safe labels are very accurately estimated. The TPs for the danger label look very low, with an overall accuracy of 37.21%



(a) Arbitrary ground truth mask.

(b) Zoomed-in sector (top edge).

Fig. 14 Ground truth mask used in training for feature detection. Right-hand side shows the inaccuracies in the pre-processing algorithm: features are labelled around the limbs of the body, and rectangles are used to flag features on the surface

for the data set. When inspecting the results visually (as shown in Fig. 12, for instance), it is clear that the accuracies shown in the confusion matrix do not look realistic with respect to the quality of the prediction. Two main reasons have been identified, both related to the way the data set images are pre-processed.

A 36.95% of all of the true danger labels are labelled as deep space, which is almost as high as the accuracy rate for the TPs. As for the shadow detection, this is not really a practical concern, since both labels are considered equally hazardous, and the problems stems again from the inaccuracies found in the ground truth mask generation algorithm.

On the other hand, 25.84% of the danger labels in the ground truth masks are, according to the confusion matrix, wrongly labelled as safe pixels. This is due to the fact that, because of simplicity reasons, the features detected in the pre-processing of the masks were framed in rectangles. These rectangles, however, also cover areas where no features can be observed in the original image, mostly next to their corners. This can be seen in Fig. 14b, where rectangles enclosed featured areas but do not exactly fit the feature's morphology. Nevertheless, the network learns how to adopt more natural shapes for the features and to distinguish very accurately the limbs of those features.

4.3 Slope estimation

Figure 15 shows the same structure as discussed for the previous two layers, applied to the slope estimation network. Again, the left-hand image shows how fully shadowed areas are problematic for the estimation of the surface slopes (as seen in the upper row example), but how the illuminated parts yield better performances. The same behaviour shown before, where the network, learns to naturalise the shapes given in the target is present in these results too. The predictions estimate the main hazard areas while paying less attention to very punctual spots, resulting in a more natural prediction of the overall slope of the body.

In the top-losses plot, a particular case appears as the worst prediction. Differently from the rest of the worst cases, where illumination conditions play a fundamental role in the accuracy of the estimation, this input image shows a well-lit celestial body. To the human eye, the prediction does not appear to be that far off from reality, e.g. the main green corridor in the middle of the asteroid is roughly correct, some of the smaller spots are covered too, and even the limb detection appears to be accurate. If anything, it appears that the network has given a conservative prediction, probably influenced by the features that are present on the right-hand side of the asteroid, which could have raised the loss value.

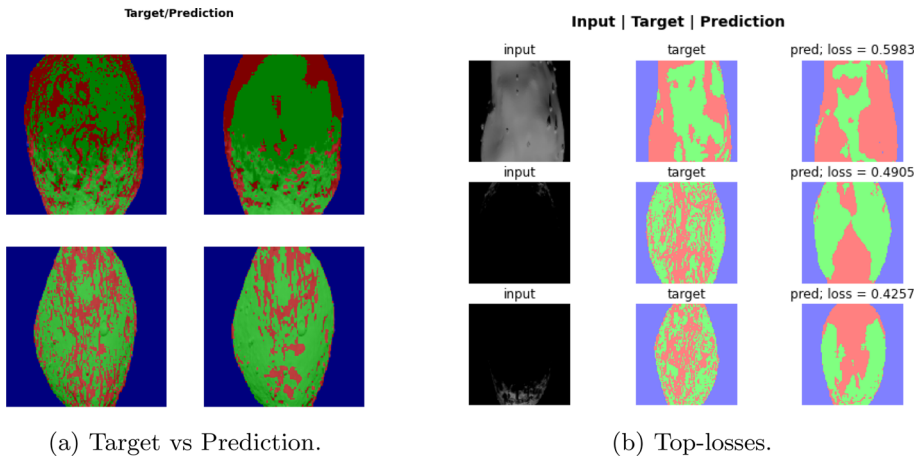
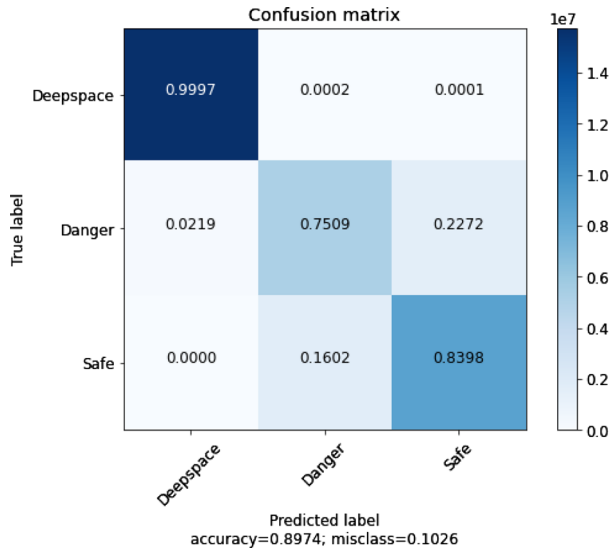


Fig. 15 Slope estimation example. Inputs and masks are overlaid. Labels map danger areas in red, safe areas in green, and deep space in blue

Fig. 16 Confusion matrix for the slope estimation layer. Performances are normalised with respect to the total number of true labels for each category, indicated by the colorbar on the right-hand side of the figure



Differently from how the masks are generated for the shadow and feature detection, slope thresholds are rendered directly from *astroRender*, so no pixel subtraction takes place here. This results in ground truth masks that do not have any mislabelling due to pre-processing. Thus, the results seen in the confusion matrix actually represent a better statistical interpretation of the quality of the predictions for the slopes. Almost no actually dangerous pixels are labelled as deep space (around 2.19%), and those detected as danger mostly lie near the limbs of the body. For the rest of the pixels, which belong to the observable surface of the body, a 75.09% success rate is achieved for the slope estimation. As described in Fig. 15a, most of this error comes from the network learning to estimate smoother-sloped areas and paying less attention to isolated high-slope surface patches.

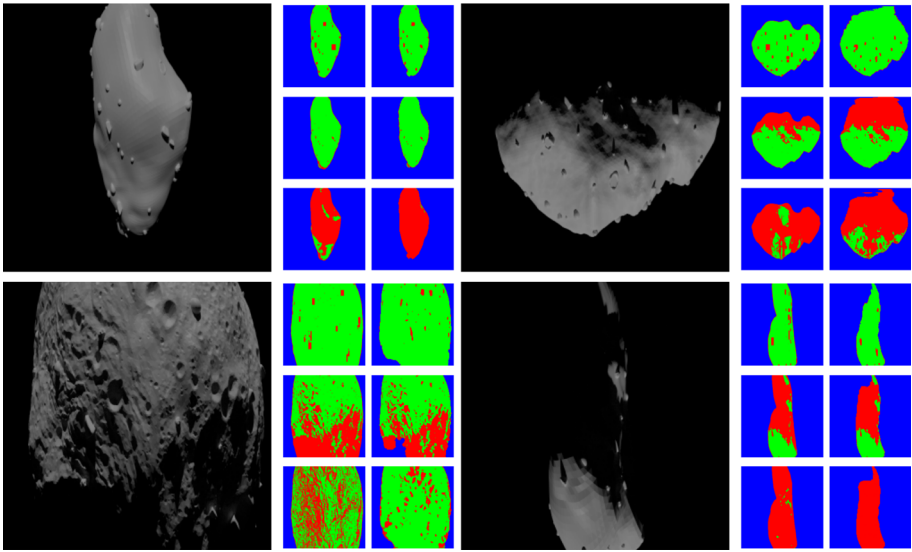


Fig. 17 Test data set results on four different rendered images. On each of the panels, the left-hand side corresponds to the rendered image given as input to the networks, the first column is the target, and the second column is the prediction. The first row corresponds to feature prediction, the second to shadow detection, and the third to slope estimation

4.4 Test set

The second round of results is generated using the test data set, which is completely external to the training of the networks, to assess how robust the predictions are when facing conditions or environments that are relatively new. Figure 17 shows four characteristic examples on which *astroHda* was tested. Different bodies are rendered and the corresponding images are then given to the CNNs, which obtain predictions for each layer. Separately, the target masks are also generated with the same pre-processing functionalities developed for the data set pipeline and used to create the training data set.

The results suggest that the feature and shadow detection capabilities of the networks are robust and adapt well to scenarios they are not familiar with. Remarkably, it can be observed how the algorithm outperforms the labelling technique used for the pre-processing of the data set. For instance, in the lower-left panel of Fig. 17, visual inspection reveals that some of the features detected by the networks had been overlooked by these pre-processing functions and are, nevertheless, detected by that layer. Shadow prediction is accurate and identifies the illuminated areas well, even under very constrained Sun-face angles.

On the other hand, functionalities related to the intrinsic geometry of the body are not that well predicted for the test data set images. Slope and body limb estimation struggle to yield satisfactory predictions when compared to the training data set. Limb estimation, which was already one of the main problems in the results shown in the top-losses plots for all layers, continues to be a challenge under poor illumination conditions and it seems that it becomes particularly difficult when the geometry of the body is not included in the training of the network. The same holds true for the slope estimation, which seems to be far from the targets for most images.

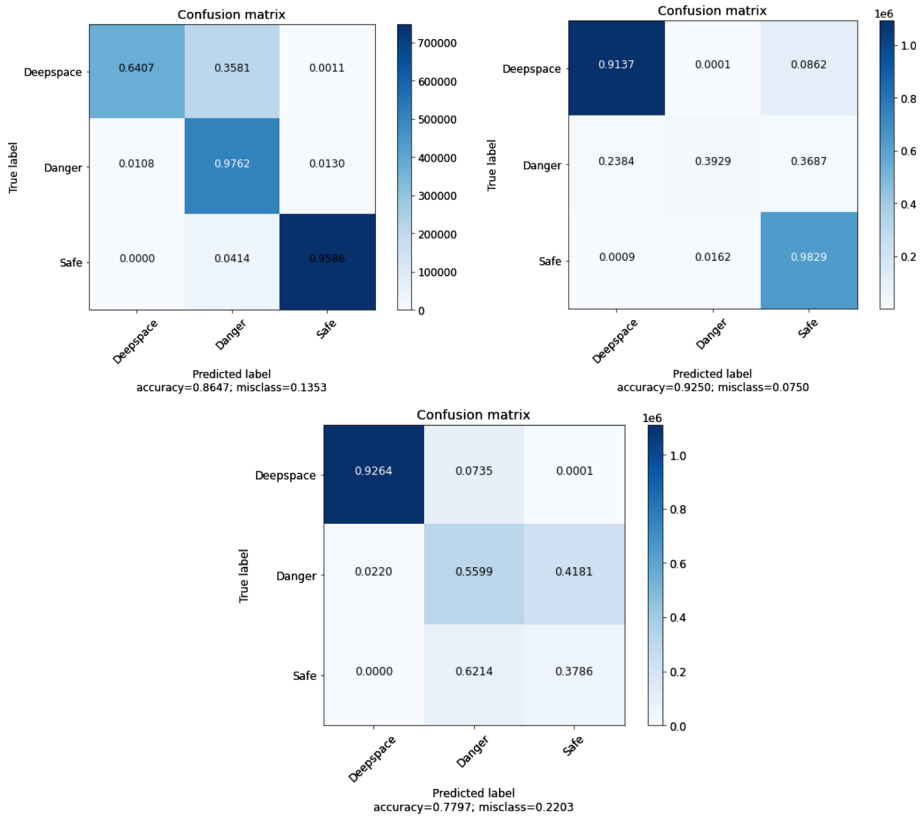


Fig. 18 Statistics obtained from the test data set for shadow detection, feature detection, and slope estimation

Similarly to what was done in previous sections, Fig. 18 shows statistics obtained from the test data set predictions and Fig. 19 the top-losses plots for that set. As expected, performances are in general worse than the ones from the training and validation data set, mainly due to the fact that the shape models and feature population distributions and geometries used to conform the test data set were not employed during training.

Nevertheless, the networks are still capable of predicting shadows and features with accuracies of 86.5% and 92.5%, respectively. A stronger impact can be observed in the performances for the slope estimation layer, which goes from an overall accuracy of 89.7% to 78.0%. As has been discussed previously, this layer is the most dependent on the knowledge of the geometry of the bodies at the training phase, thus suffering the most when completely new geometries are used as input for the prediction.

Looking at the top-losses plots also helps understand the conditions that make it more challenging for the networks to perform accurate predictions. Poor illumination conditions are again present in most of the worst cases for the networks. However, it can be seen how the limb estimation is much less accurate now that the networks have no *a priori* knowledge about the geometry of the body.

These results help confirm that including different geometries in the training phase of the algorithm greatly affects the performances of the posterior predictions for hazard detection, especially for slope estimation.

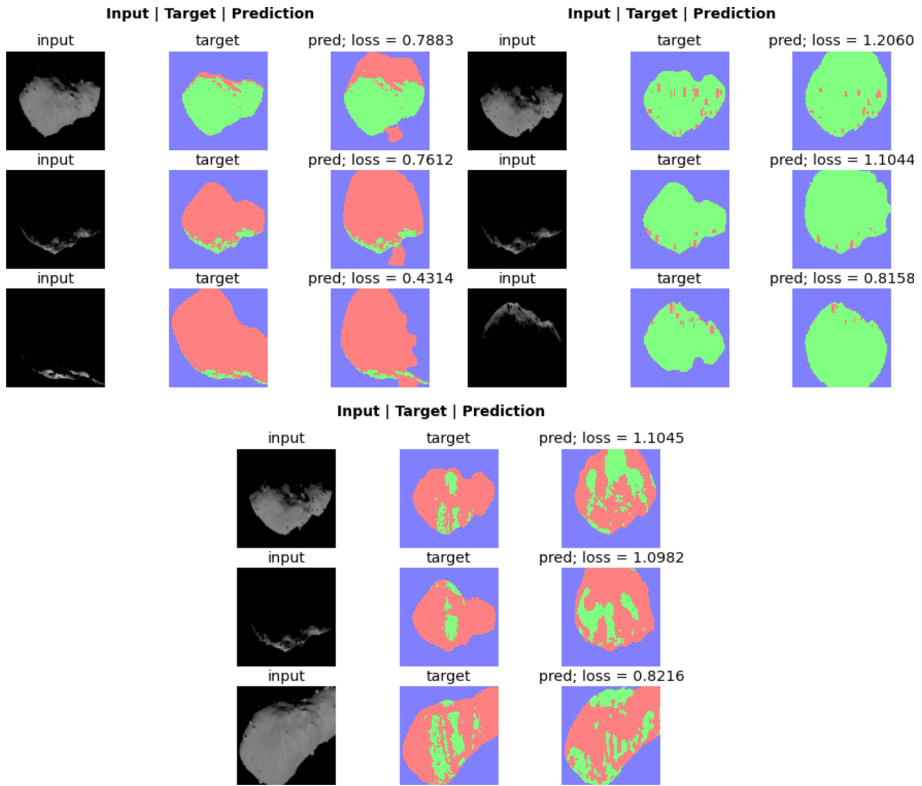


Fig. 19 Top losses for shadow detection, feature detection, and slope estimation

4.5 Layer composition

The results obtained from each layer have now been shown and discussed; however, they are not the final result that is expected from an HDA system. Instead, the different layers involved in hazard detection must be integrated to generate what is typically called a safety map.

For the creation of the safety maps in this work, a priority-based logic is chosen, where pixels flagged as deep space are prevalent over the ones flagged as a hazard, which in turn prevail over the safe pixels. This means that, if in each of the three layers described above a pixel is labelled differently, e.g. safe for the first layer, hazard for the second, and deep space for the third; it will be considered as deep space. Figure 20 shows the result of the composition of the three layers included in this work for an arbitrary image of the data set, and Table 4 shows some examples of how the logic works for a better understanding.

Each of the layers used in *astroHda* includes a softmax activation function that determines the label for the pixel, which is used in the generation of the safety maps. However, it is important to remark that this labelling comes from a probability assigned by the softmax function of a layer only concerned with its own purpose (i.e. feature detection, shadow detection, or slope estimation), but that is not aware of other layers. To further explain this concept, Fig. 21 includes the probability maps for each of the layers in addition to the target and prediction masks (already shown in Fig. 17). In case a pixel is simultaneously identified as

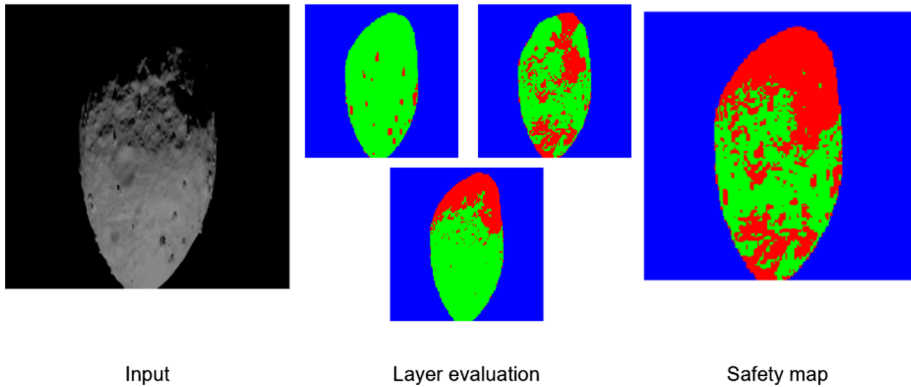


Fig. 20 Example of a rendered image, the evaluation of the different hazard detection layers, and the final safety map composition. Left is the input rendered image, centre shows (from left to right) feature detection, shadow detection, and slope estimation, and right displays the final safety map

Table 4 Example of the logic followed for layer composition

Layer 1	Layer 2	Layer 3	Final prediction
Deep space	Deep space	Safe	Deep space
Safe	Safe	Safe	Safe
Hazard	Safe	Safe	Hazard
Hazard	Deep space	Safe	Deep space

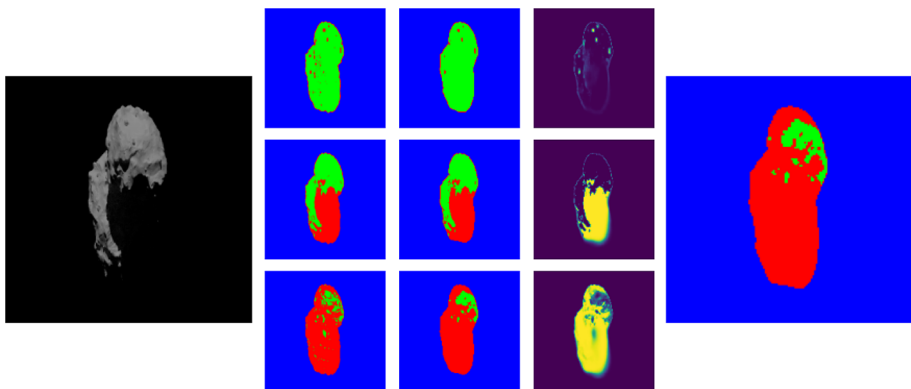


Fig. 21 Example of the probability maps used to conform a safety map on an image of comet Churyumov-Gerasimenko/67P

shadowed, featured, and sloped, one would need to trace these labels back to their respective levels of certainty in their corresponding layers. Based on that certainty (or probability), an overall best fit can be obtained and labelled. For the particular case of a safety map, where the importance lies on whether a certain pixel represents a hazard or not, this is no longer needed, since it is not ultimately relevant which type of hazard the spacecraft is looking at, but the fact that it is indeed a hazard.



Fig. 22 Set-up used at DFKI Bremen for the image acquisition. On the left, a close-up look on the mount used to hold together the camera and the processing unit. On the right, an overview of the Space Hall before completely darkening the room

4.6 HIL

One of the main weaknesses of CNN-based methods for space navigation is the little amount of experience and testing they gather. Exposing technology to real-world scenarios where the developed algorithms need to cope with inputs that have not been computer-generated help point out the weaknesses of these technologies and understand whether they are ready-to-use. By performing HIL tests, the Technology Readiness Level (TRL) of the product is increased, as a measure of how close to a final state a certain technology is.

With that in mind, and to further test the trained networks, real images, obtained at the Space Hall facility from Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) Bremen, were taken imitating the lighting conditions that a lander probe or spacecraft could be facing in the last approach phase of a descent trajectory. Different feature configurations and illumination conditions are set up, to test the algorithm in plausible space scenarios.

The hardware used for the testing consisted of a Raspberry Pi 4 model B¹ and a Raspberry Pi Camera Module v2². The networks developed were loaded into the processing unit of the Raspberry Pi and the image acquisition and safety map generation were executed “on-board”. Although the whole test happens online in the Raspberry Pi, without any connection to a computer, the results are not considered to be processor-in-the-loop (PIL) because the algorithms developed lack any kind of processor-oriented optimisation. The computational times measured from the image acquisition instant until the safety map generation were in the order of seconds, which can be greatly improved by using dedicated hardware (as it was done in Pester and Schrittmesser (2019)), optimising the input formatting for the networks, or even with more structural changes to the algorithms such as CNN-merging, for instance. A support structure for the Raspberry Pi 4 and the camera module was designed and 3D-printed and a rack was used to suspend the camera above the features. Figure 22 shows an overview of the set-up and the mount for the components utilised for image acquisition and processing.

¹ Specifications for the Raspberry Pi 4 Model B can be found here.

² Specifications for the Raspberry Pi Camera Module v2 can be found here.

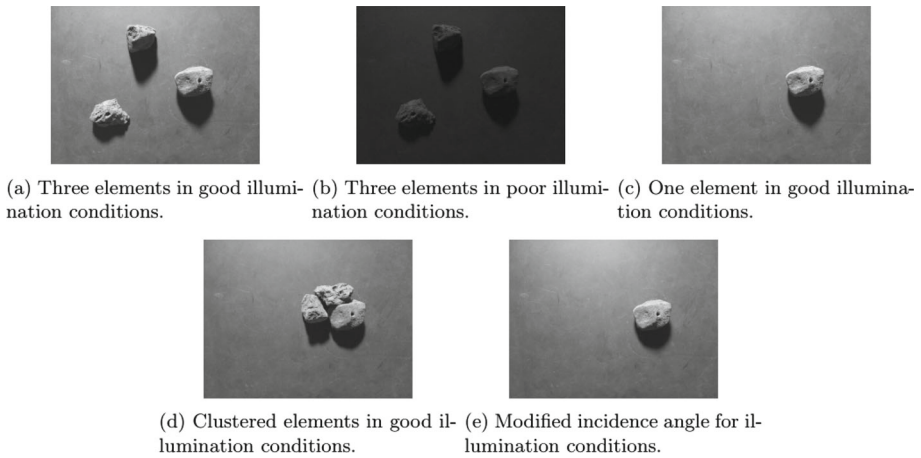


Fig. 23 Images taken at DFKI, with different feature configurations and illumination conditions

Using the described set-up, several images with different feature configurations and lighting conditions were taken. Some of these images are shown in Fig. 23. Notice how, differently from the images used in the data set, the obtained images are saturated by what is meant to be the surface of the small body, i.e. no deep-space areas are present. These images as input made showed that the inclusion of zooming and noise as augmentation techniques was paramount to obtaining accurate predictions. Figure 24 shows the probability maps obtained for image (a) in Fig. 23 before and after these two particular augmentations were applied. The main effect in the probability masks predicted by the networks is the foggy structure of the initial maps (before noise inclusion), especially in the slope estimation layer. This is due to the noise introduced by the optical sensor and it is greatly improved by the inclusion of Gaussian noise in the images in the generated data set. The features and shadows are also improved by these augmentations and the edges are now better defined.

By substituting the softmax activation functions of the three layers by a high-pass filter triggering when the probability predicted by the network was higher than 40% (empirically), some of the remaining noise is removed. The obtained predictions are shown in Fig. 25, where it can be seen how the trained networks yield accurate results for the detection of hazards even for images taken with real hardware. Nevertheless, there is still room for improvement and it can be observed how poor illumination conditions make it more difficult for the networks to understand the limbs of the features, as was anticipated during training and testing. Also, the use of entirely different geometric conditions with respect to the data set poses a challenge for the network (even after including zooming in the augmentations), which struggles to fully identify the features and leaves some of the interior areas of them as safe spots. For these tests, the focus was put on the feature and shadow detection layers, since the floor underneath the features used was flat, so more testing would be required to check how well the networks can predict slopes when in very close approach conditions.

5 Conclusions

This paper shows how the detection block of a passive HDA system is developed. CNNs utilised of to perform semantic segmentation on images, which are rendered using a dedicated

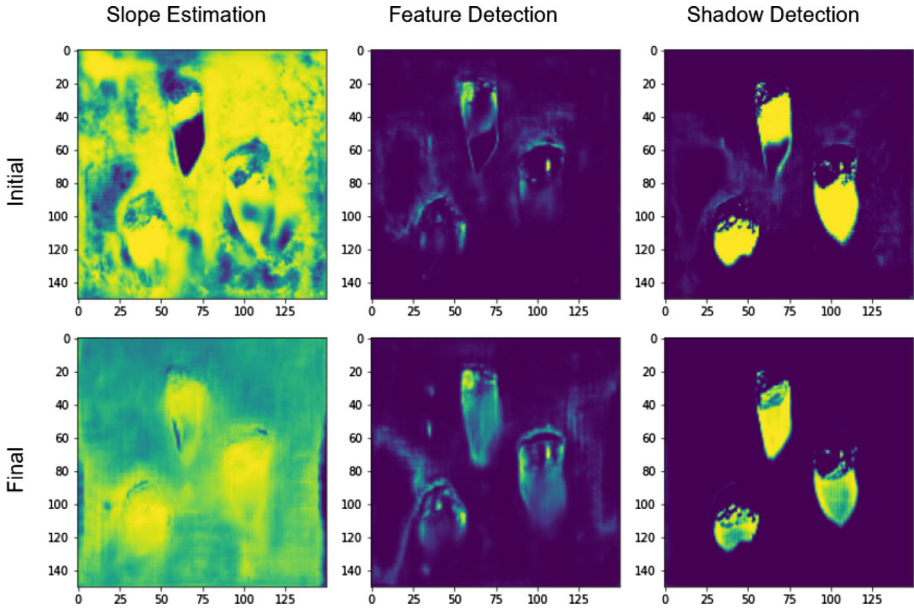


Fig. 24 Probability maps for the images obtained at DFKI facilities at Bremen before (upper row) and after (lower row) including noise and zoom augmentations for the training data set

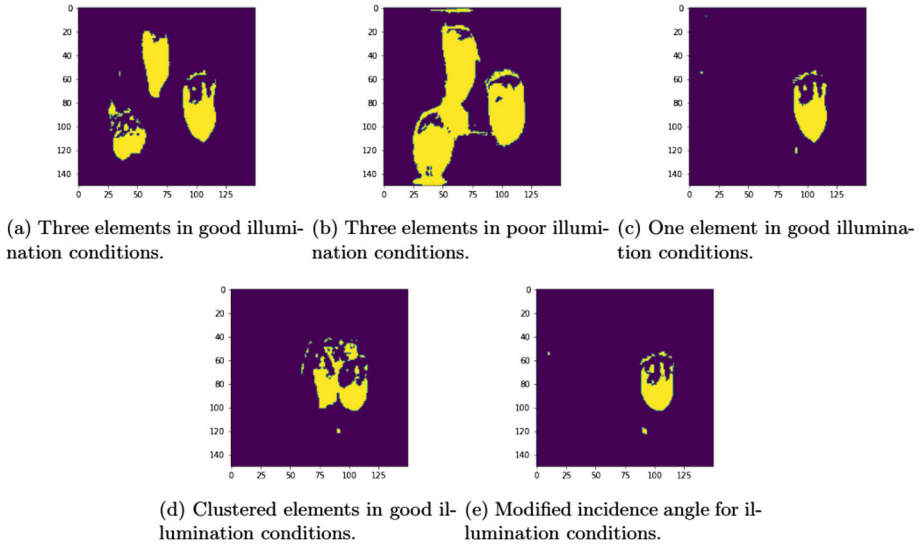


Fig. 25 Predictions obtained for the different images taken at DFKI. Hazards are highlighted in yellow

pipeline for space exploration missions around small bodies. This pipeline is capable of working with different shape models, modifying their illumination conditions (based on Sun position, if needed), and even distributing stochastically populations of features over the surface of the selected body mesh.

With the data set generated, three networks are trained using a transfer learning approach, taking ResNet-34 and ImageNet as the backbone and initial database, respectively. After a few epochs of training, and including some augmentation techniques for the generated database, the networks are capable of producing satisfactory predictions for feature detection, shadow detection, and slope estimation, with the additional sub-product of limb detection.

When the networks are tested on a different data set, external to the training phase, the feature and shadow detection capabilities still yield accurate results, but the performances for slope estimation and limb detection are considerably lowered, suggesting that the inclusion of the target bodies (or at least their rough geometries) could be key to ensure the quality of the predictions. The predicted masks of the three layers trained are merged into what are called safety maps, which are a binary representation of the areas with and without hazards on the surface of the body and that are meant to be fed to the guidance subsystem of the spacecraft to modify its trajectory if needed. Whether using layer-specific pixel predictions or probability maps is a better alternative to conform the safety maps remains a very interesting research point and leaves a door open for future work opportunities in the topic.

Additionally, HIL tests were also performed, where the networks are tested against images taken by real sensors. The inclusions of Gaussian noise and cropping in the augmentations for the data set proves to be key for the evaluation of these images, and the outcome of these tests, albeit preliminary, is satisfactory.

Acknowledgements The authors of this paper 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. Also, the authors thank DFKI Bremen for the use of their facilities and expertise to achieve some of the results shown in this paper. The authors would like to acknowledge too the funding received for APC from Universidad Carlos III de Madrid under the Read & Publish Agreement CRUE-CSIC 2023.



Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

Akçay, S., Kundegorski, M.E., Devereux, M., Breckon, T.P.: Transfer learning using convolutional neural networks for object classification within X-ray baggage security imagery. In: 2016 IEEE International

- Conference on Image Processing (ICIP), pp. 1057–1061 (2016). <https://doi.org/10.1109/ICIP.2016.7532519>
- Al-Moosawi, N.M.A.-M.M., Khudeyer, R.S.: ResNet-34/DR: a residual convolutional neural network for the diagnosis of diabetic retinopathy. *Informatica* (2021). <https://doi.org/10.31449/inf.v45i7.3774>
- Boehnhardt, H., Bibring, J.-P., Apathy, I., Auster, H.U., Ercoli Finzi, A., Goesmann, F., Klingelhöfer, G., Knapmeyer, M., Kofman, W., Krüger, H., Mottola, S., Schmidt, W., Seidensticker, K., Spohn, T., Wright, I.: The Philae lander mission and science overview. *Philosoph Trans Royal Soc A Math Phys Eng Sci* **375**(2097), 20160248 (2017). <https://doi.org/10.1098/rsta.2016.0248>
- Chen, Z., Zhu, L., Wan, L., Wang, S., Feng, W., Heng, P.-A.: A Multi-Task Mean Teacher for Semi-Supervised Shadow Detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5610–5619. IEEE, Seattle, WA, USA (2020). <https://doi.org/10.1109/CVPR42600.2020.00565>. <https://ieeexplore.ieee.org/document/9157153/> Accessed 2021-05-12
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>. ISSN: 1063-6919
- Epp, C.D., Smith, T.B.: Autonomous precision landing and hazard detection and avoidance technology (ALHAT). In: 2007 IEEE Aerospace Conference, pp. 1–7 (2007). <https://doi.org/10.1109/AERO.2007.352724>. ISSN: 1095-323X
- Gao, M., Chen, J., Mu, H., Qi, D.: A transfer residual neural network based on ResNet-34 for detection of wood knot defects. *Forests* **12**, 212 (2021). <https://doi.org/10.3390/f12020212>
- Ghilardi, L., Scorsoglio, A., Furfaro, R.: ISS Monocular Depth Estimation Via Vision Transformer. (2022)
- Glassmeier, K.-H., Boehnhardt, H., Koschny, D., Kühr, E., Richter, I.: The Rosetta mission: flying towards the origin of the solar system. *Space Sci Rev* **128**(1), 1–21 (2007). <https://doi.org/10.1007/s11214-006-9140-8>
- He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. [arXiv:1512.03385](https://arxiv.org/abs/1512.03385), Accessed 2022-09-15 (2015)
- He, K., Zhang, X., Ren, S., Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. [arXiv:1502.01852](https://arxiv.org/abs/1502.01852), Accessed 2022-08-16 (2015)
- Howard, J., et al.: fastai. GitHub (2018)
- Huertas, A., Yang Cheng, Madison, R.: Passive imaging based multicue hazard detection for spacecraft safe landing. In: 2006 IEEE Aerospace Conference, pp. 1–14. IEEE, Big Sky, MT, USA (2006). <https://doi.org/10.1109/AERO.2006.1655794>. <http://ieeexplore.ieee.org/document/1655794/> Accessed 2021-05-11
- Johnson, A.E., Huertas, A., Werner, R.A., Montgomery, J.F.: Analysis of on-board hazard detection and avoidance for safe lunar landing. In: 2008 IEEE Aerospace Conference, pp. 1–9. IEEE, Big Sky, MT, USA (2008). <https://doi.org/10.1109/AERO.2008.4526301>. <http://ieeexplore.ieee.org/document/4526301/> Accessed 2020-07-14
- Kawaguchi, J., Fujiwara, A., Uesugi, T.: Hayabusa-Its technology and science accomplishment summary and Hayabusa-2. *Acta Astronautica* **62**(10–11), 639–647 (2008). <https://doi.org/10.1016/j.actaastro.2008.01.028>
- 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., Richter, K., Sandford, S.A.: OSIRIS-REx: sample return from asteroid (101955) Bennu. *Space Sci Rev* **212**(1), 925–984 (2017). <https://doi.org/10.1007/s11214-017-0405-1>
- Liu, B., Liu, Y., Zhou, K.: Image classification for dogs and cats. TechReport University of Alberta, Alberta (2014)
- Lunghi, P., Ciarambino, M., Lavagna, M.: Vision-based hazard detection with artificial neural networks for autonomous planetary landing. In: 13th ESA/estec Symposium on Advanced Space Technologies in Robotics and Automation, ASTRA 2015, pp. 1–8 (2015)
- Lunghi, P., Ciarambino, M., Lavagna, M.: A multilayer perceptron hazard detector for vision-based autonomous planetary landing. *Adv Space Res* **58**(1), 131–144 (2016). <https://doi.org/10.1016/j.asr.2016.04.012>
- Luo, S., Li, H., Shen, H.: Deeply supervised convolutional neural network for shadow detection based on a novel aerial shadow imagery dataset. *ISPRS J Photogramm Remote Sens* **167**, 443–457 (2020). <https://doi.org/10.1016/j.isprsjprs.2020.07.016>

- Neveu, D., Mercier, G., Hamel, J.-F., Simard Bilodeau, V., Woicke, S., Alger, M., Beaudette, D.: Passive versus active hazard detection and avoidance systems. *CEAS Space J* **7**(2), 159–185 (2015). <https://doi.org/10.1007/s12567-015-0074-4>
- Pasqualetto Cassinis, L., Fonod, R., Gill, E., Ahrens, I., Gil Fernandez, J.: CNN-based pose estimation system for close-proximity operations around uncooperative spacecraft. In: *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics, Orlando, FL (2020). <https://doi.org/10.2514/6.2020-1457>. Accessed 2020-08-31
- Peñarroya, P., Centuori, S., Hermosín, P.: AstroSim: A GNC simulation tool for small body environments. In: *AIAA SCITECH 2022 Forum*, (2022). <https://doi.org/10.2514/6.2022-2355>
- Pester, A., Schrittmesser, M.: Object detection with Raspberry Pi3 and Movidius Neural Network Stick. In: *2019 5th Experiment International Conference (exp.at'19)*, pp. 326–330. IEEE, Funchal (Madeira Island), Portugal (2019). <https://doi.org/10.1109/EXPAT.2019.8876583>. <https://ieeexplore.ieee.org/document/8876583/> Accessed 2022-09-29
- Pugliatti, M., Maestrini, M.: Small-body segmentation based on morphological features with a U-shaped network architecture. *J Spacecr Rocket* (2022). <https://doi.org/10.2514/1.A35447>
- Pugliatti, M., Franzese, V., Topputo, F.: Data-driven image processing for onboard optical navigation around a binary asteroid. *J Spacecr Rocket* **59**(3), 943–959 (2022)
- Ripley, B.D.: *Pattern Recognition and Neural Networks*, (2007)
- Schwehm, G.H., Schulz, R.: The international Rosetta mission. *Lab Astrophys Space Res* **236**, 537–546 (1999). https://doi.org/10.1007/978-94-011-4728-6_22
- Sharma, S., D'Amico, S.: Neural network-based pose estimation for noncooperative spacecraft rendezvous. *IEEE Transactions on Aerospace and Electronic Systems* **56**(6), 4638–4658 (2020). <https://doi.org/10.1109/TAES.2020.2999148>. Conference Name: IEEE Transactions on Aerospace and Electronic Systems
- Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., Tamayo, D., Menou, K.: Lunar crater identification via deep learning. *Icarus* **317**, 27–38 (2019). <https://doi.org/10.1016/j.icarus.2018.06.022>
- Silvestrini, S., Piccinin, M., Zanotti, G., Brandonisio, A., Bloise, I., Feruglio, L., Lunghi, P., Lavagna, M., Varile, M.: Optical navigation for lunar landing based on convolutional neural network crater detector. *Aerospace Sci Technol* **123**, 107503 (2022). <https://doi.org/10.1016/j.ast.2022.107503>
- Tomita, K., Skinner, K., Iiyama, K., Jagatia, B., Nakagawa, T., Ho, K.: Hazard detection algorithm for planetary landing using semantic segmentation. In: *ASCEND 2020*. American Institute of Aeronautics and Astronautics, Virtual Event (2020). <https://doi.org/10.2514/6.2020-4150>
- Vignesh, S.: The world through the eyes of CNN. (2020). <https://medium.com/analytics-vidhya/the-world-through-the-eyes-of-cnn-5a52c034dbeb> Accessed 2022-09-15
- Wang, R., Di, K., Wan, W., Liu, Z., Wang, Y., Liang, W., Chen, X., Zhi, S.: Topographic mapping and analysis based on 3d reconstruction model of simulated asteroid. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B3-2020*, 1165–1170 (2020). <https://doi.org/10.5194/isprs-archives-XLIII-B3-2020-1165-2020>
- Watanabe, S.-I., Tsuda, Y., Yoshikawa, M., Tanaka, S., Saiki, T., Nakazawa, S.: Hayabusa2 mission overview. *Space Sci Rev* **208**(1), 3–16 (2017). <https://doi.org/10.1007/s11214-017-0377-1>
- Werner, R.A.: The gravitational potential of a homogeneous polyhedron or don't cut corners. *Celestial Mech Dyn Astr* **59**(3), 253–278 (1994). <https://doi.org/10.1007/BF00692875>
- Wright, L., Demeure, N.: Ranger21: a synergistic deep learning optimizer. *arXiv*. [2106.13731](https://arxiv.org/abs/2106.13731) (2021). Accessed 2022-08-16
- Zeiler, M.D., Fergus, R.: Visualizing and Understanding Convolutional Networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *Computer Vision-ECCV 2014*. Lecture Notes in Computer Science, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53