75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024.
Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

IAC–24–D1.8.4.x84612

Vision-based Positioning via Linked Neural Networks in Lunar Environment

Alessio Derobertis[a*], **Toshinori Kuwahara**[b], **Paolo Lunghi**[c]

[a] *Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa, 34 20156 Milano, Italy, alessio.derobertis@mail.polimi.it*

[b] *Department of Aerospace Engineering, Tohoku University, 6-6-11 Aramaki Aza Aoba Aoba-ku, Sendai, Miyagi 980-8579, Japan, toshinori.kuwahara.b3@tohoku.ac.jp*

[c] *Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa, 34, 20156 Milano, Italy, paolo.lunghi@polimi.it*

[*] *Corresponding author*

Abstract

The Moon has emerged as a central focus in contemporary space exploration, evidenced by the growing deployment of satellites for both orbital and landing missions. To achieve autonomous navigation research has explored various methods, from traditional algorithms to machine learning and deep learning. Despite advancements, a comprehensive absolute navigation system remains elusive, as existing solutions are often restricted by limitations in terms of altitude and availability of a-priori information. This paper introduces an innovative end-to-end AI-based solution for positioning around the Moon, capable of estimating absolute position without the need for any a-priori information while covering altitudes from tens to thousands of kilometers. The algorithm is built on a Linked Neural Networks architecture for localization, taking grayscale images of the Moon's surface and outputting a probability map of the satellite's position through a multi-stage feature extraction process. Initially, the network computes a rough estimate of the position from a Wide Angle Camera picture, to delimit the observed site. Subsequently, a Narrow Angle Camera picture is aligned with a comprehensive representation of the site using a Siamese Neural Networks architecture. The resulting output is transformed into position information to generate a refined estimate. Trained in a fully simulated environment with photorealistic images using NASA's LRO mission dataset, this approach offers high flexibility, enabling localization from any location around the Moon, without the need for any a-priori information. This innovative scheme is then compared to the current state-of-the-art in optical navigation, showcasing potential applications for miniaturized Moon missions.

Keywords: Artificial Intelligence, Deep Neural Network, Absolute Navigation, Visual-based navigation, Moon Environment, Digital Twin.

Acronyms

AI Artificial Intelligence
API Application Programming Interface
CNN Convolutional Neural Networks
DCM Direction Cosine Matrix
DEM Digital Elevation Model
DNN Deep Neural Networks
EKF Extended Kalman Filter
FOV Field Of View
GNSS Global Navigation Satellite Systems
IoU Intersection over Union
LOS Line Of Sight
LRO Lunar Reconnaissance Orbiter
LROC Lunar Reconnaissance Orbiter Camera
LSTM Long-Short Term Memory
LVLH Local Vertical Local Horizontal
ME Mean Earth/Polar Axis
NAC Narrow Angle Camera
SSD Single Shot Detection
SSIM Structural Similarity
VBN Vision-Based Navigation
WAC Wide Angle Camera

1. Introduction

Human exploration of the Moon has evolved from early missions, such as Apollo, to recent global efforts by space agencies and private companies. Key missions include NASA's *Artemis program*, which aims for a manned lunar flyby by 2025, and ISRO's *Chandrayaan-2*, contributing to the study of the lunar south pole. Japan's *SLIM mission* achieved high-precision landing, while private ventures such as SpaceX's *Starship*, Ispace's *HAKUTO-R*, and Intuitive Machine's *Odysseus* underscore growing commercial interest in lunar exploration. These missions address challenges in deep space, especially in navigation systems, which are critical in the Moon's unique environ-

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

ment, where automatic positioning systems for Earth, such as the Global Navigation Satellite Systems (GNSS), are not available, and similar solutions are still being developed for lunar applications [1].
Due to the aforementioned challenges, autonomous navigation has become crucial for mission success. Optical navigation methods are emerging as a standard in space engineering, aligning with trends like spacecraft miniaturization [2, 3] and advanced real-time processing capabilities [4]. These methods are particularly suited for deep-space missions, as they support lightweight and agile spacecraft with essential instruments. For instance, [5] presents an autonomous lunar orbit determination method using star timings, while [6] introduces a framework for crater recognition and matching to provide absolute pseudo-measurements for filtering routines. Concurrently, deep learning offers powerful tools for navigation, with Deep Neural Networks (DNN) proving effective for both absolute and relative navigation tasks. Their adaptability and performance in image processing, especially in vision-based geo-localization, have been demonstrated to surpass traditional methods [7, 8]. Thus, a rapid, adaptable navigation scheme based on camera data and DNN processing holds great promise for future missions.

1.1 State of the Art

The application of DNN in navigation is a growing field, with several approaches being explored across aerospace engineering. Starting with lunar navigation, [9] employed a crater detection strategy using the *DeepMoon* network [10], a U-Net-like segmentation architecture trained on imagery from the Lunar Reconnaissance Orbiter Camera (LROC). This network detects craters, which are then tracked using an Extended Kalman Filter (EKF). While the results were promising, the technique has significant limitations, such as altitude constraints and sensitivity to lighting changes, as the input images were cropped from the WAC Global Morphology Mosaic. Moreover, the approach requires an initial estimate of the spacecraft's location to reduce the search space for possible craters.
[11] further advanced this concept by employing a Single Shot Detection (SSD) network combined with MobileNetV2 to detect craters in lunar imagery. Once detected, traditional algorithms are used to match the craters to a database, allowing for absolute navigation with an EKF. This approach achieved a remarkable accuracy of 200 meters along the spacecraft's trajectory, but it still underutilizes the full potential of Artificial Intelligence (AI), limiting the role of the DNN to feature extraction. Additionally, the method is restricted to landing scenarios, operating only within an altitude range of 100 km to 3 km. As in previous methods, the EKF narrows the search space by estimating the camera's projection onto the lunar surface.
An improvement on this method was demonstrated by [12], who implemented a dual-mode VBN, switching between absolute and relative navigation modes depending on the spacecraft's altitude. In the absolute mode, the DNN is used to detect craters, while in relative mode, it tracks surface features like rocks and fissures across subsequent frames. Although this approach showed promising results, it shares similar limitations with [11], using AI primarily for feature extraction in absolute navigation mode.
On the other hand, [13] presented an end-to-end AI system for image-based powered descent landing guidance. Their network, composed of a Convolutional Neural Networks (CNN) and a Long-Short Term Memory (LSTM), was trained using sequences of images from a virtual scenario. The network maps these image sequences directly into actuator commands, showcasing how an end-to-end implementation can mimic traditional guidance methods. Although this is a significant achievement, the scheme is limited to local, relative navigation in a descent scenario and does not explicitly perform navigation tasks.
In the field of UAV navigation, [14] developed an end-to-end AI-based absolute navigation scheme using a Multi-Stage, Multi-Task Neural Network. This network first performs segmentation to generate a scene mask, which is then used in two parallel tasks: regression for a rough location estimate and localization to map the input mask onto a larger geographical map. The same encoder is employed in different stages, with specialized decoders for segmentation, regression, and localization. While not directly related to lunar navigation, this method achieved GPS-level accuracy in aerial imagery, highlighting the potential of multi-stage AI architectures for absolute navigation.
These examples illustrate both the progress and the limitations in the field of DNN-based navigation, particularly in lunar exploration. Current methods either focus on feature extraction or have limited applicability to specific tasks, such as landing or crater detection, leaving room for broader applications of AI in absolute navigation.

1.2 Aim of the Work

Building upon the aforementioned methods, this work aims to develop an innovative end-to-end AI-based solution for absolute navigation specifically tailored for the lunar environment. Unlike previous research, this work explores a more general navigation solution that applies to orbiting missions. The proposed algorithm performs absolute navigation from any location around the Moon, covering altitudes ranging from a few dozen kilometers to thousands of kilometers.

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

The architecture of the proposed solution is inspired by the Multi-Stage Multi-Task Neural Network approach from [14], modified and adapted for the lunar surface. A key innovation of this thesis is the use of a fully convolutional *Siamese Neural Network* [15], designed to handle the vast expanse of the Moon's surface. The network is trained in a high-fidelity virtual environment, utilizing data from NASA's LRO mission and 3D terrain reconstructions created using Blender and DEM data.
The novelty of this approach lies in its adaptability, flexibility and the capability to perform absolute navigation without any a-priori information. Previous AI-based navigation methods have concentrated on specific tasks such as landing [11, 13], crater detection [9], or hazard detection [16, 17]. In contrast, the method proposed in this thesis aims to serve as a universal navigation solution, capable of performing absolute navigation from various altitudes and locations, even in the absence of more sophisticated alternatives like GPS.
The ultimate goal is to develop a navigation system that can operate autonomously in the challenging lunar environment, providing a reliable alternative to traditional methods and enabling more versatile mission profiles for future lunar exploration.

2. Virtual Environment

The cornerstone of the proposed framework is the creation of a highly realistic virtual lunar environment, also referred to as a "digital twin". This digital twin focuses on reproducing the optical features of the Moon, accounting for lighting and shadowing caused by its surface morphology.
Various commercial solutions, such as Lunar Quickmap[·], PANGU[†], and SurRender[‡], offer advanced capabilities for lunar surface simulation, including shadow casting and texture generation. However, their limitations—particularly the need for licenses or lack of automation for large datasets—make them unsuitable for this research, which emphasizes open-source accessibility.
Blender, an open-source tool for 3D modeling and rendering, was selected for its flexibility in creating custom models, importing Digital Elevation Model (DEM) data, and simulating lighting conditions. Blender's Python Application Programming Interface (API) allows full automation of the modeling process, enabling large-scale image's datasets generation. This approach allows the creation of highly detailed, photorealistic lunar surface pictures, necessary for the development of an DNN-based lunar navigation system.

2.1 Lunar Reconnaissance Orbiter (LRO) Dataset

To ensure real world-like fidelity of the virtual environment, high-resolution data from NASA's Lunar Reconnaissance Orbiter (LRO) mission is utilized [18]. The LROC archive contains high-resolution global maps with resolutions as fine as 100 m/px. These maps not only depict surface color features but also provide DEM that capture the Moon's topography.
To determine the minimum usable resolution for the maps, we first set the maximum resolution and minimum FOV of the simulated camera, then find the minimum altitude where one pixel of the texture corresponds to one pixel in the camera frame. Equation 1 describes the relationship between the image width w, the camera FOV, altitude h, and the map resolution ρ. This equation assumes the map is applied to a flat surface.

$$w = 2h \tan\left(\frac{FOV}{2}\right)\frac{1}{\rho} \qquad [1]$$

Figure 1 illustrates how the width of the image in pixels varies with the FOV and altitude, given a map resolution of $\rho = 100$ m/px. As shown, a narrower FOV results in a less resolved image at a fixed altitude. The figure highlights y-values corresponding to 128, 256, and 512 pixels, standard image formats. Section 3.2.1 will provide a detailed analysis of the FOV, altitude, and image resolution relationship for designing the DNN architecture.

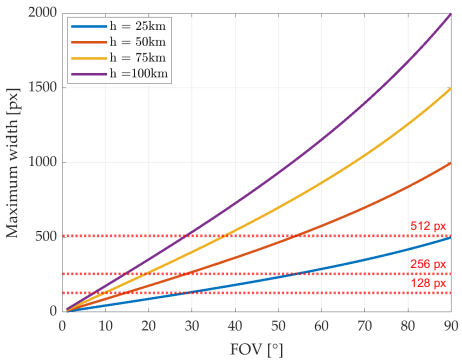


Fig. 1. Maximum width of an image depending on camera FOV and altitude. Map resolution set to 100 m/px.

Digital Elevation Model The Global Lunar DTM 100 meter topographic model (GLD100) [19] provides elevation data for the lunar surface at a resolution of 100 meters per pixel. This data is crucial for reproducing the Moon's topography in Blender. The DEM is divided into 10 tiles,

[·] https://quickmap.lroc.asu.edu/
[†] https://pangu.software/
[‡] https://www.airbus.com/en/products-services/space/customer-services/surrendersoftware

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

covering latitudes from -60 to 60 degrees in equirectangular projection and the poles in stereographic projection, as shwon in Figure 2. For compatibility with Blender, the DEM data is converted to 32-bit floating-point *.TIFF* files using QGIS, an open-source geographic information system tool.

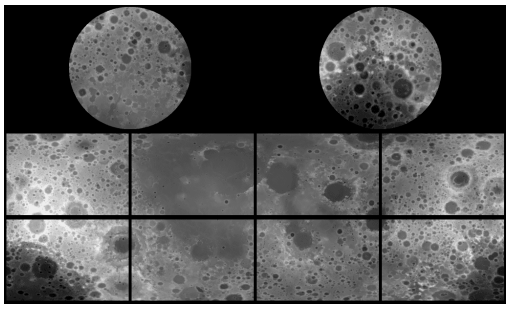


Fig. 2. GLD100 digital elevation model from the LROC data archive.

Color Maps The Empirically Normalized Reflectance Map [20], constructed from more than 137,400 images captured by the LROC's wide-angle camera (WAC), is used as the base color map for the lunar surface. This map has been photometrically normalized to remove shadows, providing a shadow-less texture onto which dynamic shadows can be cast in Blender, based on the Sun's ephemeris. It is stored in the same fashion as the GLD100, as shown in figure 3

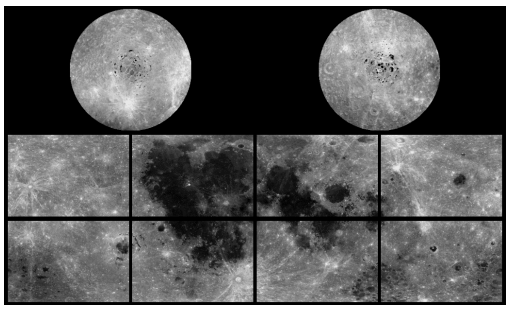


Fig. 3. Empirically Normalized Reflectance map from the LROC data archive.

2.2 Reference Frame

To ensure consistency between the optical digital twin and real lunar data, Blender's reference system is aligned with the Mean Earth/Polar Axis (ME) reference frame used by LROC products [19, 21]. The ME frame is a lunar body-fixed system, where the z-axis represents the mean rotational pole and the Prime Meridian aligns with the mean Earth direction.

2.3 Environment Modeling

To accurately align the Blender environment with LROC maps, it is crucial to create a mesh that matches the data format and apply the appropriate displacement settings. A UV Sphere is used for the mesh, with subdivisions of 180 rings and 360 segments to maintain a 1° resolution. The sphere's radius is scaled to 1737.4 m for usability while ensuring it corresponds to the Moon's actual radius. The mesh is divided based on the LROC data's stereographic and equirectangular projections to efficiently apply DEM data and address computational limits, with each subdivision representing a distinct geographical region. Pole pinching is corrected using Blender's mesh editing tools, which involve remeshing the poles to achieve smoother geometry. LROC images are mapped onto the mesh using UV mapping techniques, and materials are assigned to each object, integrating DEM data with displacement mapping for accurate surface representation. The surface's diffuse reflection is simulated using a *Diffuse BSDF* node with the Oren-Nayar model, where the roughness parameter σ_r is set to 0.364 rad. Lighting is modeled using a *Sun Light* object, with parameters based on actual solar irradiance.

2.4 Rendering

The *Cycles* engine[§] is used for rendering, employing the *Experimental Feature Set* to enable adaptive subdivisions. Figure 4 compares a real Moon image captured by the Orion optical navigation camera[¶] [22] with a simulated image rendered under similar conditions. Despite the absence of precise position, attitude, and camera model data for the real image, this qualitative comparison verifies the simulator's accuracy. The model excludes lens distortion and noise, though these could be added later through image augmentation, as common in visual dataset generation pipelines [23, 24]. Lastly, the *Blender Python API*[¨] facilitates automation of the rendering process by importing a file with scene settings, which is then processed in a loop to configure the camera and sun position and orientation and render the view. This approach supports multiple cameras within a single scene and maintains consistent Sun settings, while Blender handles mesh adjustments through adaptive subdivisions.

3. Linked Neural Network Architecture

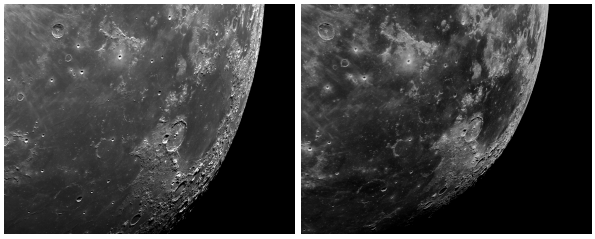
While most research on Lunar Vision-Based Navigation (VBN) with DNN has limited the network's role to

[§] https://docs.blender.org/manual/en/latest/render/cycles/index.html

[¶] https://www.flickr.com/photos/nasa2explore/52554839804/in/album-72177720303788800

[¨] https://docs.blender.org/api/current/index.html

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.



(a) Orion picture (b) Generated picture

Fig. 4. Comparison with real data.

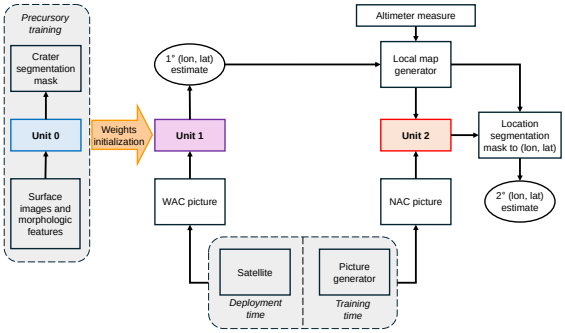


Fig. 5. Linked Neural Networks architecture.

feature extraction [10–12], this work proposes an end-to-end AI-based positioning scheme utilizing Linked Neural Networks for refined estimation accuracy. This method extends the operational range from dozens to thousands of kilometers, bypassing the need for initial conditions or any a-priori information.
The proposed architecture is divided into three units:

- Crater Segmentation Network (Unit 0);
- Location Regression Network (Unit 1);
- Location segmentation Network (Unit 2).

Unit 0 is a U-net-based segmentation network that identifies craters in lunar images, providing a dense representation of geological features. While it does not directly contribute to localization, Unit 0 serves as the foundation for Unit 1, a regression network that estimates the latitude and longitude of Wide Angle Camera (WAC) images using an encoder pre-trained on Unit 0.
To further refine the estimate, Unit 2 employs a fully convolutional Siamese network for change detection [15]. It takes a Narrow Angle Camera image and a local map generated from Unit 1's estimate, outputting a binary segmentation image. Activated pixels correspond to the Narrow Angle Camera (NAC) image's projection onto the local map, allowing for a refined position estimate by computing the centroid of the activated zone.
Figure 5 illustrates the full network architecture, detailing data flow during both training and deployment.

3.1 Crater Segmentation Network - Unit 0

Unit 0 is built on a fully-convolutional auto-encoder architecture, consisting of a contractive section (the encoder) and an expansive section (the decoder). The encoder generates a feature map, which serves as a latent representation of the features extracted from the input image. This feature map is then passed to the decoder, which produces a segmentation map of the same size as the input. The output contains multiple channels, with each channel corresponding to the segmentation map of a specific class present in the dataset. The network structure is illustrated in Figure 6.

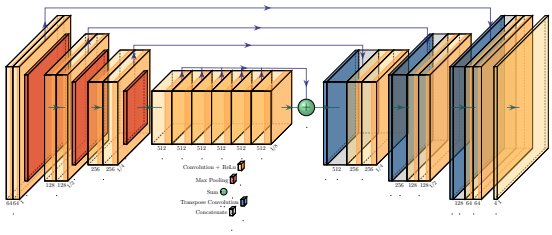


Fig. 6. Unit 0's architecture.

Encoder The encoder comprises three blocks, each with two 2D Convolution layers and a 2x2 Max Pooling layer, followed by six chained 2D Convolution layers with exponentially increasing dilation rates. ReLU activations follow each convolution, while dropout layers are inserted between the convolutions.

Decoder The decoder generates segmentation maps by upscaling the feature maps. After summing the output of each convolution layer in the chained convolution channel-wise, the input is scaled up using 2x2 Transpose Convolution layer, concatenated with the correspondent encoder's layer using skip connection and then passed through two 3x3 2D Convolution layers. This block of operations is repeated for three rimes. ReLU activations follow each convolution, and dropout layers are applied between the two convolutional layers.

3.1.1 Dataset

The dataset used to train Unit 0 is the one developed in [25], consisting of 768 images obtained by cropping the WAC Global Mosaic and manually labeled in COCO†† format. The dataset is divided into a training set (538 images)

††https://cocodataset.org

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

and a validation set (230 images). The images are stored as *.jpg* files at 512x512 px resolution. In this research, the labels are divided into three groups, depending on the area of the craters in an image, leading to four classes to segment (including the background). The subdivision of the craters is as follows:

- Craters with an area greater than 96 px^2;
- Craters with an area between 96 px^2 and 32 px^2;
- Craters with an area smaller than 32 px^2.

Figure 11 shows an image with the corresponding mask and the network's prediction.

3.1.2 Training

The model is trained using TensorFlow API with weighted categorical cross-entropy loss. To reduce overfitting and improve the generalization of the network, image augmentation is employed, making use of random flips, translations, rotations and zooms. To compensate for the dataset's unbalance, being the big craters more represented than the smaller ones, class weights were employed during training. After some experimentation, satisfactory results were achieved by setting the weights as reported in table 1. The workflow of the experiments, along with the results, is reported in section 4.1.

Class	Weight
Background	0.5
Small craters	10
Medium craters	7
Large craters	5

Table 1. Unit 0's class weights.

3.2 Location Regression Network - Unit 1

Unit 1 is fundamental in the process of localizing the spacecraft. Its objective is to estimate the position in terms of longitude and latitude in the Moon's ME reference frame. It is composed of an encoder, shared with Unit 0, and a fully connected regression top.
Unit 1's encoder is copied from Unit 0 after training, then re-trained together with the regression top. This allows the network to generate a meaningful feature map to be fed to the regression top, leading to better localization. The network architecture is shown in Figure 7.

Encoder The structure of Unit 1's encoder is the same as Unit 0's one. The weights are initialized by copying those of Unit 0's encoder and re-trained together with the rest of the network. Since Unit 0 is trained to segment craters, its encoder learns to extract the key morphological features, compressing them into a feature map. The same features are employed in Unit 1 for localization purposes.

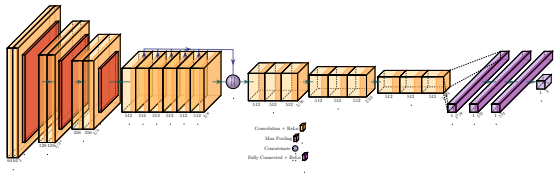


Fig. 7. Unit 1's architecture.

Regression Top While Unit 0's decoder generates a segmentation map from the feature map, Unit 1 uses it to estimate location coordinates (longitude and latitude). Instead of segmentation, Unit 1 performs regression with a fully connected network on top of the encoder, linked via three convolutional blocks. Unlike Unit 0's decoder, the outputs of convolution layers in Unit 1 are concatenated channel-wise before passing through three convolutional blocks, each with three 3x3 2D convolution layers, where the first layer has a stride of 2 to reduce input dimensions. A flattening layer connects the convolutional part to a fully connected one, formed by two dense layers with 512 neurons each, and an output layer with 2 neurons. ReLU activation functions are applied after each convolutional or dense layer, except the last one.

3.2.1 Dataset

The dataset used to train Unit 1 is composed of artificially generated WAC pictures of the Moon's surface coupled with labels containing the location of the camera. This dataset is generated using the scheme developed in Chapter 2. First, a vector of epochs is generated, which is then used to define the position of the Sun. For every point in the time vector, a random position around the Moon is generated, and the corresponding attitude is computed. All these data are then stored in a *.csv* file and input into Blender through the Python API for rendering.
A total of 9,954 pictures are generated and divided into training (70%) and validation/test (30%) sets.

Random Sampling Around a Sphere To guarantee a uniform distribution of points around the Moon, at least in terms of longitude and latitude, a random unit vector is first generated using three independent normal random numbers (*X, Y, Z*) with mean 0 and variance 1. This means that:

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

$$\mathbf{u} = \frac{1}{\sqrt{X^2 + Y^2 + Z^2}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad [2]$$

is uniformly distributed on the surface of a unit sphere [26]. Then, for each point, a random number h from a continuous uniform distribution with the lower endpoint being the minimum altitude and the upper endpoint being the highest altitude is generated. The position at a specific point is then given by:

$$\mathbf{r} = h\mathbf{u} \quad [3]$$

Although this choice results in more points clustered at lower altitudes than at higher ones, this behavior is actually desired, as the network has been found to be less accurate when predicting locations closer to the surface compared to those further away.

Altitude Choosing the altitude range is an important task, as it will limit the network's capability to certain altitudes. In this case, the lower limit is dictated by the resolution of the image generator (see section 2.1). For Unit 1's dataset generation, the FOV is set to $90°$, which allows not only for low altitudes to be reached guaranteeing a wide portion of the Moon's surface in the picture, but also for pictures taken at higher altitudes to capture enough details. However, the strongest requirement comes from the FOV of Unit 2's dataset, which is generated using the same positions as Unit 1, but employing a NAC with a $30°$ FOV, as will be explained in section 3.3.1. This constrains the minimum altitude to 100 km. Regarding the maximum altitude, there is no such constraint, but it is important to consider the relative size of the Moon in a picture taken from a very high altitude. For this reason, the maximum altitude is set to 1,500 km.
After preliminary experiments, it was observed that the pictures taken between 100 km and 500 km resulted in the lowest accuracy in position estimation. To address this issue, an unbalanced dataset was generated, increasing the number of pictures taken at lower altitudes while reducing those taken at higher altitudes. Table 2 explains the distribution in detail.

Altitude Range	Percentage of Pictures
[100 km, 500 km]	50%
[500 km, 1,000 km]	30%
[1,000 km, 1,500 km]	20%

Table 2. Unit 1's dataset altitude distribution.

Lastly, it should be noted that positions generated using this method are already expressed in the Mean Earth/Polar Axis reference frame, since it is the one used by the picture generator.

Illumination The Sun's position is determined using ephemeris models from SPICE[‡‡], which computes the position in the Mean Earth/Polar Axis reference frame from an epochs' vector. This method ensures lunar surface features are observed under realistic illumination conditions, reducing the variability in the dataset and aiding the network's training process. Figure 8 illustrates the distribution of the Sun's azimuth and elevation as seen from the Moon's surface. It is noticeable how, near the equator, the Sun is visible at all elevations, with azimuth concentrated within the $80° - 100°$ and the $260° - 280°$ regions. At the poles, the Sun is visible at all azimuths, but maximum elevation is about $5°$.
To maintain a minimum portion of the Moon's surface in each image, the Sun's elevation angle is constrained to $[0°, 90°]$, while the azimuth angle is unconstrained. This constraint is applied by generating more samples than needed and filtering to retain those meeting the constraint. The epoch vector spans 20,000 hours (about 833 days) starting from *2024-06-25 11:41:42.819936 UTC*, ensuring comprehensive sampling of all possible solar positions relative to the Moon.

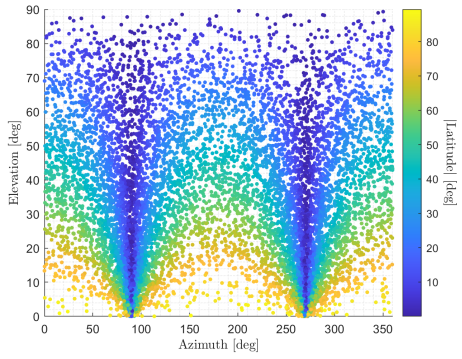


Fig. 8. Distribution of azimuth and elevation of the Sun.

Attitude Constraints are also imposed on the satellite's attitude. Figure 9 shows the configuration of the body frame and the camera frame. The satellite is assumed to be constantly pointing in the radial direction while keeping its attitude aligned with a frame similar to the Local Vertical Local Horizontal (LVLH) frame, but with the y-axis always parallel to the equatorial plane. This frame is referred to as the Equatorial LVLH and is defined as fol-

[‡‡] https://naif.jpl.nasa.gov/naif/toolkit.html

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

lows. The x-axis (named r) is parallel to the radial direction, pointing towards space. The y-axis (named t) is defined as the cross product between the z-axis of the ME reference frame and the x-axis of the Equatorial LVLH frame. Finally, the z-axis (named h) results from the cross product between the x and y axes. Equation 4 shows the computation of the reference frame starting from the position of the satellite in the Moon Centered ME frame $\boldsymbol{y_{MC,ME}}$.

$$\begin{aligned} \boldsymbol{r} &= \frac{\boldsymbol{y_{MC,ME}}}{||\boldsymbol{y_{MC,ME}}||} \\ \boldsymbol{t} &= \frac{\begin{bmatrix}0\\0\\1\end{bmatrix} \times \boldsymbol{r}}{||\boldsymbol{r}||} \\ \boldsymbol{h} &= \boldsymbol{r} \times \boldsymbol{t} \end{aligned} \quad [4]$$

Using the unit vectors of the Equatorial LVLH frame, the Direction Cosine Matrix (DCM) that transforms vectors from the ME frame to the Equatorial LVLH frame can be computed as:

$$\boldsymbol{A}_{ME,elvlh} = \begin{bmatrix}\boldsymbol{r}\\\boldsymbol{t}\\\boldsymbol{h}\end{bmatrix} \quad [5]$$

Since the body frame is assumed to coincide with the Equatorial LVLH frame, the matrix $\boldsymbol{A}_{ME,elvlh}$ can be used to transform vectors directly from the ME frame to the body frame. The same unit vectors can be used to define the DCM that transforms a vector from the ME frame to the camera frame, since, as shown in figure 9, the axes of the camera frame are parallel to those of the body frame but point in different directions and have a different order.

$$\boldsymbol{A}_{ME,Cam} = \begin{bmatrix}\boldsymbol{n}\\\boldsymbol{t}\\-\boldsymbol{r}\end{bmatrix} \quad [6]$$

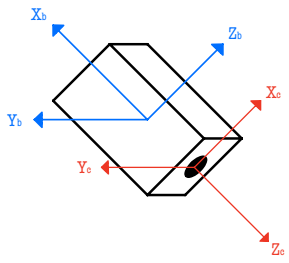


Fig. 9. Satellite body and camera frames.

The attitude constraint ensures that all images are taken with the camera consistently pointing north, which reduces dataset variability and simplifies training. In practice, this can be achieved using a horizon sensor for radial alignment and a star sensor for inertial attitude estimation [27]. By rotating images based on known ephemerides and inertial attitudes, the process remains autonomous and does not require prior information, allowing free rotation of the satellite around the local vertical.

3.2.2 Training

Unit 1 was trained using the TensorFlow API. The loss function used for regression is mean squared error.The network is trained using the *Adam* optimizer with a learning rate of 1×10^{-4}. To reduce overfitting and help the network achieve better generalization, image augmentation is applied. Unlike Unit 0's augmentation, Unit 1's training dataset is augmented using random zoom, contrast and brightness adjustment. This augmentation technique also aims to reproduce the behavior of a real camera, where pictures are taken under different exposure conditions. Results from Unit 1's training are shown in section 4.2

3.3 Location Segmentation Network - Unit 2

Unit 2 refines the position estimated by Unit 1 by taking two inputs: a NAC image of the Moon and a local map generated based on Unit 1's estimate. The output is a segmentation map where activated pixels indicate the overlay between the two images. By averaging these activated pixels, a refined location estimate is obtained. Unit 2's architecture is inspired by the fully convolutional networks described in [15] but uses blocks from Unit 0. The architecture of Unit 2 is shown in Figure 10.

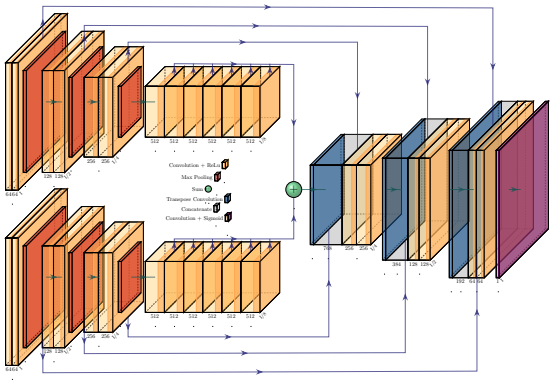


Fig. 10. Unit 2's architecture.

Encoder The network includes two parallel encoders, each with the same structure as Unit 0's encoder. Unlike

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

Unit 1, these encoders are independently initialized rather than sharing weights with Unit 0's encoder.

Decoder The decoder follows Unit 0's design, with the addition of concatenating three layers at each level of the decoder, instead of two. Skip connections link corresponding layers from both encoders to the decoder: one encoder processes the Moon's surface NAC image, and the other processes the local map. This setup enhances segmentation accuracy by integrating spatial information from early encoder layers.

3.3.1 Dataset

Unit 2's dataset consists of image pairs with corresponding segmentation masks. Each pair includes a NAC picture of the Moon and a local map. The segmentation mask represents the overlapping region between the NAC image and the local map. The dataset used to train Unit 2 is identical to the one used for testing Unit 1 (section 4.2) and includes 9931 images from random points around the Moon, with altitudes ranging from 100 km to 1500 km. Unlike Unit 1's training data, this dataset has a uniformly distributed altitude range. It is divided into training (70%), validation/test (30%).

NAC Pictures The NAC pictures are produced together with the WAC ones, using same position, attitude and illumination as explained in section 3.2.1. The only difference is the Field Of View (FOV) of the camera, which, in this case, is set to $30°$. This value was chosen considering the constraints explained in section 2.1, as it is the lowest FOV that allows for pictures with a resolution of 512x512 pixels.

Local Maps The local map is an orthographic projection of a portion of the Moon, centered on the position estimated by Unit 1. Unlike perspective cameras, orthographic cameras allow close positioning to the surface while representing a large region by adjusting the sensor size, making them ideal for map generation without FOV limitations. The same model from section 2 was used, with modifications to the camera type and lighting. Sunlight was set to illuminate the entire area from the spacecraft's position. The orthographic camera sensor's dimension d_{ortho} is computed taking into account Unit 1's estimation error, together with an altitude estimate coming from a laser altimeter. Since Unit 1's localization accuracy varies with altitude (Table 4), different map dimensions are needed to ensure the NAC's projected FOV is located within the map. Equation 7 outlines the map length calculation, starting with estimated altitude $\hat{h}$ and Moon radius R_M, computing the angle α that accounts for the NAC's projected FOV's width. The angle γ is determined by adding the maximum between the mean absolute error for longitude and latitude to three times its standard deviation, ensuring a 99.73% probability that the NAC's projected FOV is within the map, barring outliers.

$$\begin{aligned} \alpha &= \pi - \frac{NAC}{2} - \arcsin\left(\sin\left(\frac{NAC}{2}\right)\frac{\hat{h}}{R_M}\right) \\ \sigma &= \mu_{mae} + 3\sigma_{mae} \\ d_{orhto} &= 2R_M \sin\left(\sigma + \alpha\right) \end{aligned} \quad [7]$$

To prevent the map from capturing space beyond the Moon's surface, the maximum map length is limited by:

$$d_{ortho,max} = 2R_M \sin\left(\frac{\pi}{4}\right) \quad [8]$$

The map dimensions are set to 512x512 pixels. For spacecraft deployment, generating maps on-board from a full equirectangular projection of the Moon (e.g., WAC Empirically Normalized Reflectance map [20]) can be efficiently achieved using geospatial manipulation libraries like GDAL[§§], which is computationally cheaper than Blender and requires only the map characteristics and Moon surface data.

Masks The ground truth data for training Unit 2 consists of binary masks that represent the projection of the NAC's FOV onto the local orthographic map. To generate these masks, the FOV of the NAC camera is first projected onto the lunar surface. The Line Of Sight (LOS) for each pixel in the camera frame is computed following Equation 9. Using the DCM from section 3.2.1, the LOS is transformed into the ME reference frame. The intersection points $\boldsymbol{p}_{MC,ME}$ on the Moon's surface are computed as the intersection between the LOS and a sphere centered at the Moon's origin, solved using Equation 11. Each pixel's position is assumed to match the satellite's position $\boldsymbol{y}_{MC,ME}$, used to generate both WAC and NAC images.

$$\boldsymbol{l}_{CC,Cam} = \frac{\begin{bmatrix} v - v_c \\ -u + u_c \\ f \end{bmatrix}}{\sqrt{(v - v_c)^2 + (-u + u_c)^2 + f^2}} \quad [9]$$

$$\boldsymbol{l}_{CC,ME} = \boldsymbol{A}^T_{ME,Cam}\boldsymbol{l}_{CC,Cam} \quad [10]$$

$$\begin{cases} ||\boldsymbol{p}_{MC,ME}||^2 = R_M^2 \\ \boldsymbol{p}_{MC,ME} - \boldsymbol{y}_{MC,ME} = d\boldsymbol{l}_{CC,ME} \end{cases} \quad [11]$$

In Equation 9, u and v are the pixel coordinates, with u_c and v_c being the center of the frame. After computing d for each pixel, the intersection points $\boldsymbol{p}_{MC,ME}$ are

[§§] https://gdal.org/en/latest/

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

mapped to the orthographic frame using Equation 12. A mask is created by activating pixels in a black image according to the projected coordinates $\boldsymbol{p}_{MC,ME}$. The transformation to the Camera Centered Camera frame is done as follows:

$$\begin{bmatrix} X_{CC,Cam} \\ Y_{CC,Cam} \\ Z_{CC,Cam} \end{bmatrix} = \boldsymbol{A}_{ME,Cam}(\boldsymbol{p}_{MC,ME} - \hat{\boldsymbol{y}}_{MC,ME})$$
$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_c + f\frac{Y_{CC,Cam}}{d_{ortho}} \\ v_c - f\frac{X_{CC,Cam}}{d_{ortho}} \end{bmatrix} \quad [12]$$

Here, $X_{CC,Cam}$, $Y_{CC,Cam}$, and $Z_{CC,Cam}$ denote the position in the camera frame, and $\hat{\boldsymbol{y}}_{MC,ME}$ is the Unit 1 estimate. The focal length f is calculated as:

$$f = \frac{s}{2\tan\left(\frac{FOV}{2}\right)} \quad [13]$$

The process of creating maps and masks is outlined in Algorithm 1 while figure 12 shows some samples of NAC pictures and local maps, with the correspondent masks and the network's predictions.

Algorithm 1 Local maps and masks creation.

1: **for** Every pair of WAC and NAC images and the corresponding location $\boldsymbol{y}_{MC,ME}$ in the dataset **do**
2: Predict the WAC image location using Unit 1;
3: Simulate altitude measurement;
4: Convert the prediction to Cartesian coordinates to obtain $\hat{\boldsymbol{y}}_{MC,ME}$;
5: Compute d_{ortho} using Equation 7;
6: Generate a local map using Blender, the predicted position $\hat{\boldsymbol{y}}_{MC,ME}$, and d_{ortho};
7: Project the NAC's FOV onto the Lunar surface using Equation 11 and $\boldsymbol{y}_{MC,ME}$;
8: Generate a black image representing the orthographic camera frame;
9: Activate the pixels corresponding to the projected NAC following Equation 12.
10: **end for**

3.3.2 Training

Unit 2 was trained using the TensorFlow API, employing a hybrid loss function to enhance segmentation performance, as defined in [28]. It combines binary cross-entropy, Intersection over Union (IoU) and Structural Similarity (SSIM). The network was trained using the *Adam* optimizer with a learning rate of 1×10^{-4} and $\epsilon = 1 \times 10^{-8}$. Image augmentation is also included, applying brightness and contrast adjustments only to NAC images. Results are outlined in section 4.3

4. Experiments

4.1 Unit 0

Unit 0, the Crater Segmentation Network, is designed to generate segmentation maps of craters from lunar surface images. It was trained for 2500 epochs on the dataset detailed in section 3.1.1.

The main metric for evaluating Unit 0 is the IoU [29], defined as the ratio of overlap between predicted and ground truth segmentation maps to their union. Another key metric is Recall, which measures the ability of a model to correctly identify all relevant instances in a dataset. It is defined as the ratio of true positive instances to the sum of true positive and false negative instances. Since the network may segment additional craters not present in the labels, recall is useful in evaluating the network's ability to correctly identify craters without penalization for unannotated ones. Table 3 summarizes the performance on the test dataset, with IoU for all classes and recall for the crater classes. Figure 11 shows examples of segmented images.

Class	IoU	Recall
Background	0.934	
Small craters	0.319	0.687
Medium craterts	0.494	0.749
Big craters	0.468	0.601

Table 3. Unit 0's results.

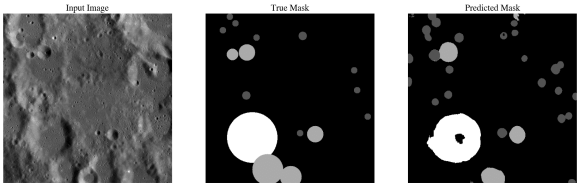


Fig. 11. Examples of crater segmentation maps predicted using Unit 0.

While Unit 0's accuracy is lower than in [25], its performance is sufficient for its main role as an encoder for Unit 1. It effectively generates a dense spatial representation of the lunar surface, enabling Unit 1 to estimate the position of images taken from various locations on the Moon.

4.2 Unit 1

The Location Regression Network (Unit 1) estimates the position of a satellite orbiting the Moon based on a WAC image of the surface. Unit 1 was trained for 500

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

epochs using the dataset described in section 3.2.1, with the validation mean squared error tracked throughout training.
For evaluating regression performance, the bootstrap method is employed, allowing computation of statistical quantities such as mean error and variance by generating permutations of the network's predictions on the test dataset [30]. Unit 1's performance is evaluated using mean squared error and variance, with the number of permutations set to $N = 20,000$. To perform the bootstrap analysis, a new test dataset was generated with a uniform altitude distribution, resulting in 9,931 images. Predictions were computed and categorized into altitude ranges to evaluate performance within specific ranges (Table 2). Table 4 shows the mean absolute error and standard deviation for both longitude and latitude.
To better assess Unit 1's performance, predictions were transformed into Cartesian coordinates using simulated laser altimeter measurements [11], assuming a standard deviation of 1% of the true altitude. Then the bootstrap analysis was performed. This transformation allows a statistical evaluation of the error dispersion in Cartesian space. Results are presented in Table 5 for the Body Centered Equatorial Local Vertical Local Horizontal frame.

h [km]	μ_{lon}	σ_{lon}	μ_{lat}	σ_{lat}
[100,1500]	15.1°	0.448°	4.67°	0.0801°
[100,500]	31.3°	1.10°	8.75°	0.212°
[500,1000]	7.99°	0.532°	2.95°	0.0757°
[1000,1500]	7.40°	0.572°	2.66°	0.0676°

Table 4. Unit 1's results from Bootstrap analysis (in geographic coordinates referred to the Moon Centered Mean Earth/Polar Axis reference frame).

Results from Unit 1's regression task clearly show that the network effectively learned to represent the Moon's surface for localization purposes. Despite not reaching the level of accuracy required for real-world applications, it is noteworthy that, starting from a single picture taken from any location around the Moon, the network can localize it with only a few degrees of error. When combined with a laser altimeter, the network produces estimates with errors within hundreds of kilometers on all axes, with accuracy varying by altitude range.

4.3 Unit 2

The Location Segmentation Network (Unit 2) refines the estimates from Unit 1 by mapping a NAC image onto local map of the Moon, generated using the position estimated by Unit 1. The network then outputs a probability map representing the NAC's FOV projection onto the local map, enabling a refined location estimate. Training of Unit 2 spans 285 epochs using the dataset in section 3.3.1, with validation mean IoU monitored ate each epoch.
Unit 2's segmentation performance is first evaluated through IoU on the test dataset, followed by location data analysis. Two datasets are used: the test dataset described in section 3.3.1 for IoU computation and a newly generated 5032-image dataset for statistical analysis using the bootstrap method.
Table 6 shows Unit 2's IoU results. Figure 12 displays example segmentation masks alongside ground truth data. The results reveal Unit 2 struggled with segmentation, with some masks entirely black (figure 12a) or showing activations where none were expected (figure 12b). While some masks are reasonably accurate (figure 12c), others are significantly off (figure 12d).

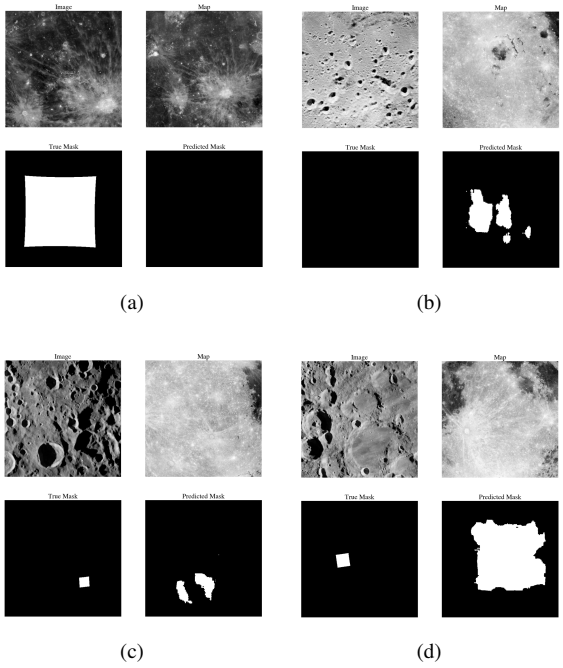


Fig. 12. Examples of segmentation results: false black masks, over-activated masks, and correct masks.

Segmentation results indicate that Unit 2's accuracy is insufficient for high-precision localization, particularly for NAC images close to the Moon's surface, where masks display few activated pixels.
To evaluate the localization performance of Unit 2, the output mask is converted into position using Equation 17. Since Unit 2 outputs a probability map, where each pixel value ranges from 0 to 1 representing the probability of ac-

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

h [km]	μ_x	σ_x	μ_y	σ_y	μ_z	σ_z
[100, 1500]	109.43 km	9.6066 km	208.04 km	6.6959 km	195.58 km	6.3880 km
[100, 500]	259.08 km	24.547 km	332.44 km	15.966 km	318.03 km	15.090 km
[500, 1000]	31.101 km	7.8235 km	145.07 km	7.6488 km	125.21 km	5.8973 km
[1000, 1500]	34.164 km	10.204 km	143.83 km	6.7949 km	141.32 km	8.3132 km

Table 5. Unit 1's results from Bootstrap analysis (in Cartesian coordinates referred to the Body Centered Equatorial Local Vertical Local Horizontal reference frame).

Class	IoU
Background	0.386
Mask	0.265

Table 6. Unit 2's IoU.

tivation, these values are converted into a binary mask by applying a threshold of 0.5. Then the coordinates of the pixel corresponding to the centroid of the activated region in the mask are computed and converted to position information, obtaining a refined location estimate in terms of longitude and latitude. For black masks, the central pixel of the image was assumed to be the centroid.
The position of the centroid is converted into location information by finding the intersection of the Line Of Sight of the centroid's pixel with the Moon, as shown in Equation 17. Equation 14 shows how to compute the LOS of a pixel in the ortographic camera frame while equation 16 defines the position of the pixel in the Moon Centered ME frame ($\boldsymbol{x}_{MC,ME}$). $\hat{\boldsymbol{y}}_{MC,ME}$ denotes the position estimated by Unit 1 and converted to Cartesian coordinates using the altimeter's measure. When computing the centroid's projection, u and v in equation 16 has to coincide with its coordinates in the camera frame, while u_c and v_c are the coordinates of the center of the frame.

$$\boldsymbol{l}_{CC,Cam} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{14}$$

$$\boldsymbol{l}_{CC,ME} = \boldsymbol{A}^T_{ME,Cam} \boldsymbol{l}_{CC,Cam} \tag{15}$$

$$\boldsymbol{x}_{MC,ME} = \boldsymbol{A}^T_{ME,Cam} \frac{f}{d_{ortho}} \begin{bmatrix} -v + v_c \\ u - u_c \\ 0 \end{bmatrix} - \hat{\boldsymbol{y}}_{MC,ME} \tag{16}$$

$$\begin{cases} ||\boldsymbol{p}_{MC,ME}||^2 = R_M^2 \\ \boldsymbol{p}_{MC,ME} + \boldsymbol{x}_{MC,ME} = d\boldsymbol{l}_{CC,ME} \end{cases} \tag{17}$$

After the conversion of each mask into latitude and longitude, the same altimeter measure used in section 4.2 is attached and the bootstrap method is applied to evaluate the accuracy of the refined location estimates. The ground truth position used to calculate the error is the one used to generate the WAC image input to Unit 1 for the initial estimate, and the NAC image input to Unit 2 along with the local map.
Table 7 presents the results in geographical coordinates, while Tables 8 reports the Cartesian conversion using the altimeter measurement, in Equatorial Local Vertical Local Horizontal frame.

h [km]	μ_{lon}	σ_{lon}	μ_{lat}	σ_{lat}
[100,1500]	16.3°	0.684°	5.12°	0.136°
[100,500]	31.5°	1.52°	10.2°	0.36°
[500,1000]	9.60°	0.978°	2.73	0.0698°
[1000,1500]	8.35°	0.907°	2.58°	0.123°

Table 7. Unit 2's results from Bootstrap analysis (in geographic coordinates referred to the ME reference frame).

The localization results in Tables 7 and 8 highlight the limitations of Unit 2's segmentation performance. Compared to Unit 1's results (Tables 4 and 5), Unit 2 not only fails to improve accuracy but actually performs worse. This degradation is linked to the local map generation mechanism. Each map represents a segment of the Moon's surface centered on the position estimated by Unit 1, covering a span which is related Unit 1's estimation accuracy. More specifically, Unit 1's mean error can be computed as the distance between the central pixel of the map and an edge pixel, which is equal to half of the width of the NAC's FOV projection, plus Unit 1's bootstrap mean absolute error, plus three times the bootstrap variance, as described in section 3.3.1. In contrast, the worst-case error for Unit 2 corresponds to the maximum distance between 2 pixels in the local map, when the estimated and ground truth centroids are maximally distant. This explains why Unit 2's error can be greater than Unit 1's one. Further-

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

h [km]	μ_x	σ_x	μ_y	σ_y	μ_z	σ_z
[100,1500]	118.48 km	6.1967 km	221.92 km	4.5017 km	201.80 km	3.8936 km
[100,500]	290.11 km	16.197 km	384.44 km	10.582 km	353.37 km	9.7407 km
[500,1000]	28.607 km	4.6611 km	142.27 km	4.8095 km	121.93 km	3.3758 km
[1000,1500]	39.577 km	7.0052 km	141.69 km	5.2421 km	132.68 km	4.3981 km

Table 8. Unit 2's results from Bootstrap analysis (in Cartesian coordinates referred to the Equatorial Local Vertical Local Horizontal reference frame).

more, pixel dimensions, which vary from 100 meters to hundreds of kilometers depending on altitude, introduce an intrinsic error, setting a lower bound for achievable accuracy.

To assess the potential of Unit 2's estimation process, a comparative analysis using ground truth masks is performed, assuming perfect segmentation. The results, shown in Tables 9 and 10, demonstrate that with ideal segmentation, the estimation error is significantly reduced, achieving errors within tens of kilometers. However, even with perfect segmentation, the intrinsic map-to-position conversion process introduces error due to the map resolution and pixel dimensions.

h [km]	μ_{lon}	σ_{lon}	μ_{lat}	σ_{lat}
[100,1500]	7.33°	0.538°	1.51°	0.121°
[100,500]	16.4°	1.28°	4.02°	0.348°
[500,1000]	2.73°	0.650°	0.183°	0.0332°
[1000,1500]	2.99°	0.676°	0.379°	0.107°

Table 9. Unit 2's results from Bootstrap analysis using ground truth masks (in geographic coordinates referred to the ME reference frame).

The poor segmentation performance of Unit 2 can be attributed to several factors. Firstly, the characteristics of the local maps play a critical role. Differences in parameters like exposure, contrast, or texture (e.g., using the WAC Global Mosaic versus the WAC Empirically Normalized Reflectance). Additionally, the choice of a hybrid loss function [28] may not be optimal for this task. Exploring alternatives, such as a threshold loss [31], could improve performance. The model selection was also reasonable, however, experimenting with different Siamese architectures or models with fully connected tops [32, 33] may offer better results.

Despite these issues, the ground truth analysis validated the working principle of Unit 2's network. This suggests that with a more optimized architecture, significant accuracy improvements could be achieved.

5. Conclusion

The work expands on Deep Neural Networks-based lunar navigation schemes by enhancing altitude range and system autonomy through an end-to-end AI approach. A virtual environment, based on LRO mission data, was usd to generate realistic lunar surface images, allowing the development of a Linked Neural Networks architecture. The novelty of this work consists in the Linked Neural Network architecture implemented and its ability to perform localization completely autonomously. Even tough the performances of the network were slightly compromised by Unit 2's poor segmentation accuracy, overall, the system achieved position estimates within tens to hundreds of kilometers depending on altitude, demonstrating the potential of DNNs for autonomous lunar navigation. Indeed, the algorithm does not require any a-priori information, unlike many similar applications which require prior knowledge of position, allowing for navigation without any preliminary data. Furthermore, the sensor suite required to perform positioning is extremely limited, consisting only of two cameras with different FOVs (or a single camera with a variable FOV), an horizon sensor, a star tracker, and a laser altimeter. Ground contact is also not needed, keeping the system fully autonomous.

5.1 Future Developments

The future potential for development in this work is extensive. Starting with the DNN architecture, although basic networks based on the U-Net module were employed, modern DNN can offer far greater accuracy across different tasks. With further research into advanced networks, there is a strong potential for improvement in position estimation accuracy. Moreover, because of the linked architecture of the DNN, enhancements in one unit, such as better feature extraction in Unit 0, would propagate throughout the system, improving localization in Unit 1 and reducing the local maps for Unit 2, leading to better accuracy overall.

The dataset could be greatly expanded using the optical digital twin of the Moon. This expansion could include images under various illumination conditions and slightly

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

h [km]	μ_x	σ_x	μ_y	σ_y	μ_z	σ_z
[100,1500]	89.860 km	6.2221 km	61.640 km	3.9417 km	49.994 km	3.3824 km
[100,500]	220.87 km	16.400 km	147.14 km	10.360 km	123.33 km	9.0254 km
[500,1000]	19.147 km	4.5934 km	19.709 km	3.7704 km	10.852 km	2.0444 km
[1000,1500]	31.927 km	7.1969 km	19.651 km	4.0361 km	17.189 km	3.3781 km

Table 10. Unit 2's results from Bootstrap analysis using ground truth masks [km] (in Cartesian coordinates referred to the Equatorial Local Vertical Local Horizontal reference frame).

relaxed attitude constraints to simulate real satellite behavior, further increasing the network's robustness and generalization.

Another area of exploration is the FOV of the satellite's cameras. If a variable focal length camera were available, it could be used to capture images at higher and lower altitudes, which could be easily incorporated into Unit 1 to significantly extend the altitude range. A similar approach for NAC images could yield more focused local maps for Unit 2, enhancing its precision.

Additionally, the whole system could be adapted to other celestial bodies if an optical twin and morphological features datasets for those environments were created. This would allow for the retraining of the entire system for navigation in different space environments.

The potential applications are numerous, and further optimization could significantly improve the current system, bringing the accuracy from within tens of kilometers to possibly tens of meters, paving the way for a fully autonomous and precise navigation solution.

References

[1] J. Kawaguchi *et al.*, "On the low-cost asynchronous one-way range measurement method and the device for micro to nano deep space probes," *Acta Astronautica*, vol. 214, pp. 457–465, 2024, ISSN: 0094-5765.

[2] T. Kuwahara *et al.*, "Impacts of space plug-and-play technology on micro- and nano-satellites," *IFAC Proceedings Volumes*, vol. 46, no. 19, pp. 289–294, 2013, 19th IFAC Symposium on Automatic Control in Aerospace, ISSN: 1474-6670.

[3] T. Kuwahara *et al.*, "International scientific microsatellite risesat based on space plug and play avionics," in *Proceedings of the 26th Annual AIAA/USU Conference on Small Satellites*, 2012.

[4] Y. Sato, T. Kuwahara, S. Fujita, Y. Sakamoto, and K. Yoshida, "Development and ground evaluation of fast tracking algorithm for star trackers," *Transactions Of The Japan Society For Aeronautical And Space Sciences, Aerospace Technology Japan*, vol. 16, no. 3, pp. 202–209, 2018.

[5] M. Psiaki and J. Hinks, "Autonomous lunar orbit determination using star occultation measurements," in *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aereonautics and Astronautics, August 2007. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2007-6657.

[6] J. A. Christian, H. Derksen, and R. Watkins, "Lunar crater identification in digital images," *The Journal of the Astronautical Sciences*, vol. 68, no. 4, pp. 1056–1144, Oct. 2021, ISSN: 2195-0571.

[7] P.-E. Sarlin *et al.*, *Orienternet: Visual localization in 2d public maps with neural matching*, 2023. arXiv: 2304.02009 [cs.CV].

[8] F. Pro, N. Dionelis, L. Maiano, B. L. Saux, and I. Amerini, *A semantic segmentation-guided approach for ground-to-aerial image matching*, 2024. arXiv: 2404.11302 [cs.CV].

[9] L. M. Downes, T. J. Steiner, and J. P. How, *Lunar terrain relative navigation using a convolutional neural network for visual crater detection*, 2020. arXiv: 2007.07702 [cs.CV].

[10] A. Silburt *et al.*, "Lunar crater identification via deep learning," *Icarus*, vol. 317, pp. 27–38, Jan. 2019, ISSN: 0019-1035.

[11] S. Silvestrini *et al.*, "Optical navigation for lunar landing based on convolutional neural network crater detector," *Aerospace Science and Technology*, vol. 123, p. 107 503, 2022, ISSN: 1270-9638.

[12] L. Ostrogovich, R. Del Prete, G. Tomasicchio, N. Longepe, and A. Renga, "A dual-mode approach for vision-based navigation in a lunar landing scenario," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2024, pp. 6799–6808.

Copyright © 2024 by the International Astronautical Federation (IAF). All rights reserved.

[13] A. Scorsoglio, A. D'Ambrosio, L. Ghilardi, B. Gaudet, F. Curti, and R. Furfaro, "Image-based deep reinforcement meta-learning for autonomous lunar landing," *Journal of Spacecraft and Rockets*, vol. 59, no. 1, pp. 153–165, 2022.

[14] A. Marcu, D. Costea, E. Slusanschi, and M. Leordeanu, *A multi-stage multi-task neural network for aerial scene interpretation and geolocalization*, 2018. arXiv: 1804.01322 [cs.CV].

[15] R. C. Daudt, B. L. Saux, and A. Boulch, *Fully convolutional siamese networks for change detection*, 2018. arXiv: 1810.08462 [cs.CV].

[16] P. Lunghi, M. Ciarambino, and M. Lavagna, "A multilayer perceptron hazard detector for vision-based autonomous planetary landing," *Advances in Space Research*, vol. 58, no. 1, pp. 131–144, 2016, ISSN: 0273-1177.

[17] L. Ghilardi and R. Furfaro, "Image-based lunar hazard detection in low illumination simulated conditions via vision transformers," *Sensors*, vol. 23, p. 7844, Sep. 2023.

[18] M. S. Robinson *et al.*, "Lunar reconnaissance orbiter camera (LROC) instrument overview," *Space Science Reviews*, vol. 150, no. 1, pp. 81–124, Jan. 2010.

[19] F. Scholten *et al.*, "Gld100: The near-global lunar 100 m raster dtm from lroc wac stereo image data," *Journal of Geophysical Research: Planets*, vol. 117, no. E12, 2012. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2011JE003926.

[20] A. K. Boyd, M. S. Robinson, and H. Sato, "Lunar Reconnaissance Orbiter Wide Angle Camera Photometry: An Empirical Solution," in *43rd Annual Lunar and Planetary Science Conference*, Mar. 2012, 2795, p. 2795.

[21] LRO Project and Lunar Geodesy and Cartography Working Group, "A standardized lunar coordinate system for the lunar reconnaissance orbiter and lunar datasets," National Aeronautics and Space Administration, Tech. Rep., 2008.

[22] C. N. D'Souza, K. Smith, and R. Inman, *Orion optical navigation testing and performance*, 2020. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2020-0229.

[23] M. Bechini, M. Lavagna, and P. Lunghi, "Dataset generation and validation for spacecraft pose estimation via monocular images processing," *Acta Astronautica*, vol. 204, pp. 358–369, 2023, ISSN: 0094-5765.

[24] M. Bechini, G. Gu, P. Lunghi, and M. Lavagna, "Robust spacecraft relative pose estimation via cnn-aided line segments detection in monocular images," *Acta Astronautica*, vol. 215, pp. 20–43, 2024, ISSN: 0094-5765.

[25] R. Del Prete, A. Saveriano, and A. Renga, "A deep learning-based crater detector for autonomous vision-based spacecraft navigation," in *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2022, pp. 231–236.

[26] M. E. Muller, "A note on a method for generating points uniformly on n-dimensional spheres," *Commun. ACM*, vol. 2, no. 4, pp. 19–20, Apr. 1959, ISSN: 0001-0782.

[27] T. Kuwahara, S. Battazzo, Y. Tomioka, K. Fukuda, Y. Sakamoto, and K. Yoshida, "System integration of a star sensor for the small earth observation satellite rising-2," *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan*, vol. 10, no. ists28, 2012.

[28] X. Qin *et al.*, *Boundary-aware segmentation network for mobile and web applications*, 2021. arXiv: 2101.04704 [cs.CV].

[29] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, *Generalized intersection over union: A metric and a loss for bounding box regression*, 2019. arXiv: 1902.09630 [cs.CV].

[30] U. Michelucci and F. Venturini, "Estimating neural network's performance with bootstrap: A tutorial," *Machine Learning and Knowledge Extraction*, vol. 3, no. 2, pp. 357–373, 2021, ISSN: 2504-4990.

[31] C. Bailer, K. Varanasi, and D. Stricker, *Cnn-based patch matching for optical flow with thresholded hinge embedding loss*, 2020. arXiv: 1607.08064 [cs.CV].

[32] S. Zagoruyko and N. Komodakis, *Learning to compare image patches via convolutional neural networks*, 2015. arXiv: 1504.03641 [cs.CV].

[33] J. Žbontar and Y. LeCun, *Stereo matching by training a convolutional neural network to compare image patches*, 2016. arXiv: 1510.05970 [cs.CV].