

EXECUTIVE SUMMARY REPORT

AI-AIDED-XR

Prepared by: AI-Aided-XR Project Team

Approved by: Steven Kay (GMV)

Authorized by: Steven Kay (GMV)

Code: GMV-AIAXR-ESR

Version: 2.0

Date: 20/03/2025

Internal code: GMV 22601/24 V2/25

DOCUMENT STATUS SHEET

Version	Date	Pages	Changes
2.0	20/03/2025	24	Initial release
2.0	20/03/2025	28	Updated for final delivery

TABLE OF CONTENTS

1. INTRODUCTION	5
1.1. PURPOSE AND SCOPE	5
2. DEFINITIONS AND ACRONYMS	6
2.1. DEFINITIONS	6
2.2. ACRONYMS	6
3. REFERENCES	7
3.1. APPLICABLE DOCUMENTS	7
3.2. REFERENCE DOCUMENTS	7
4. PROJECT OVERVIEW	8
5. USE CASES	9
5.1. XR4AI-UC-1.6 PLANETARY TERRAIN CLASSIFICATION	9
5.2. AI4XR-UC-1.2 GENERATION OF TERRAINS BASED ON FRACTALS AND GENERATIVE ADVERSARIAL NETWORKS	10
6. AI-AIDED-XR ARCHITECTURE OVERVIEW	11
6.1. COMBINED ARCHITECTURE	11
7. RESULTS	12
7.1. XR4AI-UC-1.6 PLANETARY TERRAIN CLASSIFICATION RESULTS	12
7.1.1. Metrics	12
7.1.2. Inference Results	13
7.1.3. AINabler Visualization	15
7.2. AI4XR-UC-1.2 GENERATION OF TERRAINS BASED ON FRACTALS AND GENERATIVE ADVERSARIAL NETWORKS ARCHITECTURE RESULTS	17
7.2.1. Navigation dataset generation	17
7.2.2. Configurable Parameters	17
7.2.3. 360 image for XR visualization	19
7.2.4. Stereo image for any visualization device	20
7.2.5. Terrain Editor	21
7.2.6. Terrain Learner	22
8. NEXT STEPS	26
9. SUMMARY	27

LIST OF TABLES AND FIGURES

Table 2-1 Definitions	6
Table 2-2 Acronyms	6
Table 3-1 Applicable Documents	7
Table 3-2 Reference Documents	7
Table 3: Models and their metrics	12
Figure 4-1: The AI-Aided-XR Loop – Synthetic images, generated in an XR environment, fed into a terrain classifier, and returned to train the XR terrain learner.	8
Figure 5-1: Virtuous, Synergistic relationship of AI and XR technologies.	9
Figure 5-2: GMV POL LHDAC Synthetic Moon Dataset.....	10
Figure 6-1: AI-Aided-XR Architecture	11
Figure 7-1: Results from the terrain classifier	12
Figure 7-2: Inference Results on LHDAC Dataset	13
Figure 7-3: Results of Inference on Base Model	14
Figure 7-4: Results of Fine-tuning	15
Figure 7-5: The AIAXR Pipeline	16
Figure 6-6: Visualized Predictions	16
Figure 7-3: Example height map of a path	17
Figure 7-4: RGB image, label image and distance image from a single traverse section	17
Figure 7-5: Example Images from Configurable Parameters	19
Figure 7-6: 360 Image Visualization.....	20
Figure 7-7: Stereo Imagery for VR Devices	20
Figure 7-8: Terrain Editor GUI.....	21
Figure 7-9: Different types of images and saving them.....	22
Figure 7-10– Training Loss function of the Moon Heightmap Classifier using CNN.....	23
Figure 7-11 - Confidence matrix of the Moon Heightmap Classifier using CNN	23

1. INTRODUCTION

1.1. PURPOSE AND SCOPE

The objective of this document is to concisely summarise the findings of the AI-Aided-XR activity in an organized fashion.

The structure of the document is as follows:

- Chapter 1: The introduction chapters.
- Chapter 2: Definitions and Acronyms
- Chapter 3: References & Applicable Documents
- Chapter 4: Project Overview
- Chapter 5: Use Cases
- Chapter 6: Architecture Overview
- Chapter 7: Results
- Chapter 8: Next Steps
- Chapter 9: Summary

2. DEFINITIONS AND ACRONYMS

2.1. DEFINITIONS

Concepts and terms used in this document and needing a definition are included in the following table:

Table 2-1 Definitions

Concept / Term	Definition

2.2. ACRONYMS

Acronyms used in this document and needing a definition are included in the following table:

Table 2-2 Acronyms

Acronym	Definition
AI	Artificial Intelligence
AR	Augmented Reality
BDRF	Bidirectional reflectance distribution function
BERT	Bidirectional Encoder Representations from Transformers
BSSRDF	Bidirectional Subsurface Scattering Distribution Function
BTF	Bidirectional Texture Function
CAD	Computer-Aided Design
CCD	Charge Coupled Device
CG	Computer Graphics
cGAN	Conditional General Adversarial Network
CLIP	Contrastive Language-Image Pre-Training
CNN	Convolutional Neural Network
CV	Computer Vision
DEM	Digital Elevation Models
DL	Deep Learning
DM	Diffusion Model
DDM	Denosing Diffusion Model
DDPM	Denosing diffusion probabilistic models
DMTet	Differentiable marching tetrahedrons
DSN	Deep Space Network
ES-DDPM	Early Stopping DDPM
GAN	General Adversarial Network
GPL	GNU General Public License
GPU	Graphical Processing Unit
IGM	Implicit Generative Model
INS	Inertial Navigation System
INS	Input-Output operations
LDM	Latent Diffusion Models
LIDAR	Light Detection and Ranging
LoD	Level of Detail
ML	Machine Learning
MPL	Mozilla Public Licence
MTM	Multiple-Try Metropolis
NCSN	Noise conditioned score networks / Noise-Contrastive Signal-to-Noise Ratio
NeRF	Neural Radiance Fields
NURBS	Non-Uniform Rational B-Splines
PBR	Physically-based rendering
RL	Reinforcement Learning
ROCC	Rover Operations Control Centre
ROI	Region of Interest
SGM	Score-based Generative Models
SDE	Stochastic Differential Equations
SVBRDF	Spatially Variant BRDF
TDPM	Truncated Denosing Process Model
TRL	Technology Readiness Level
VAE	Variational AutoEncoders
VFX	Visual Effects
VR	Virtual Reality
XR	Extended Reality

3. REFERENCES

3.1. APPLICABLE DOCUMENTS

The following documents, of the exact issue shown, form part of this document to the extent specified herein. Applicable documents are those referenced in the Contract or approved by the Approval Authority. They are referenced in this document in the form [AD.x]:

Table 3-1 Applicable Documents

Ref.	Title	Code	Version	Date
[AD. 1]	AI-Aided-XR Final Report	GMV-AIAXR-FR	1.0	26/04/2024
[AD. 2]	AI-Aided-XR Full Test Harness	GMV-AIAXR-TR-0001	1.0	26/04/2024
[AD. 3]	AI-Aided-XR Lessons Learned	GMV-AIAXR-TN-0005	1.0	26/04/2024

3.2. REFERENCE DOCUMENTS

The following documents, although not part of this document, amplify or clarify its contents. Reference documents are those not applicable and referenced within this document. They are referenced in this document in the form [RD.x]:

Table 3-2 Reference Documents

Ref.	Title	Code	Version	Date
[RD. 1]	AINabler documentation https://gitlab.space-codev.org/ainabler/ainabler-documentation			

4. PROJECT OVERVIEW

At its core, the AI-Aided-XR activity is conceptualized to address the symbiotic relationship between Artificial Intelligence (AI) and eXtended Reality (XR) technologies through the selection of applicable space domain use cases, followed by the implementation and validation of software prototypes which address the requirements of the concerned use cases. This project embodies a novel approach to investigate exactly how AI and XR can mutually enrich and empower each other.

An emphasis of the AI-Aided-XR project is the harmonious, cyclical loop created by AI and XR feeding each other. Use cases which demonstrate this are investigated in the context of the space domain, but with the lineage of GMV’s experience in previous projects which interfaced directly with ESA data stores to automatically and routinely run computer vision tools to aid mission planning. This, coupled with the consortium’s expertise in XR technology, grants a unique opportunity to create a state-of-the-art coupling of synthetically generated terrain environments and expanded semantic segmentation for terrain classification.

AI in the space domain is an incredibly valuable resource. From computer vision to natural language processing, there are boundless opportunities for AI to assist the space industry in the pursuit of science and exploration. However, amidst this rapidly evolving field lies a formidable challenge in the scarcity and quality of data to train these AI models. Training robust machine learning models crucially requires high-quality labelled data and a vast amount of it – this is multiplied by the challenging environments in space, making gathering these datasets incredibly difficult. Yet, amongst these challenges there lies an opportunity for innovation and discovery with novel research.

With the above knowledge, it makes sense that the selected use cases, discussed in greater detail in the next chapters, focus on synthetic data generation to get the most out of these data-hungry AI applications. By creating a truly bespoke XR environment using fractal generation techniques, any number of images can be captured with tens of parameters to define element of the terrain, essentially creating a high-fidelity, highly realistic simulation of a lunar environment. The AI half of the harmonious loop, as discussed above, will inherit, and evolve ViBEKO’s terrain classification technology. The most compelling part of this project is that *high-quality, high-quantity datasets for the moon simply do not exist*. We are directly filling a gap which is in need of attention, especially as more attention is shifting to lunar activities.

Alongside the overview of the project above, it is the intention to design, implement, validate then deliver the SW prototypes up to a TRL of 4, and include an assessment for the next steps to see the further advancement of them towards operational tools.

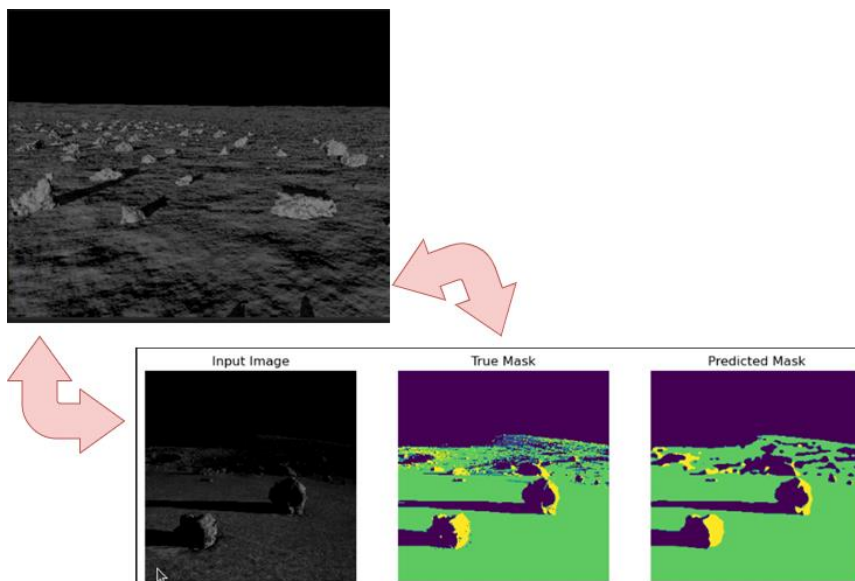


Figure 4-1: The AI-Aided-XR Loop – Synthetic images, generated in an XR environment, fed into a terrain classifier, and returned to train the XR terrain learner.

5. USE CASES

Two use cases are to be implemented as software (SW) prototypes in the AI-Aided-XR activity. Both use cases are tackling one half of the synergistic relationship between AI and XR technologies, as graphically shown in Figure 5-1 below. The objective is to implement prototypes of these use cases in parallel, to highlight the synergies and symbiotic relationship, emphasising and demonstrating the combined benefits that these technologies provide to each other. The use cases selected are described in the sub-sections below.

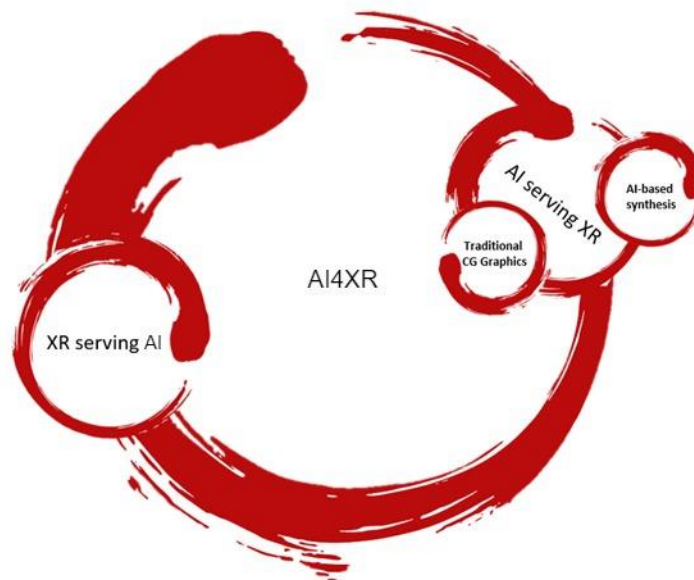


Figure 5-1: Virtuous, Synergistic relationship of AI and XR technologies.

5.1. XR4AI-UC-1.6 PLANETARY TERRAIN CLASSIFICATION

Understanding the type of terrain, and therefore the characteristics, that lie ahead of a planetary rover is an essential input for evaluating the traversability cost in driving through that specific area. With knowledge of upcoming terrain types, an enhanced situational awareness is obtained, providing an invaluable perspective for mission planners performing iterative activity drive plans, or from the on-board software perspective, an important input when planning paths autonomously. While historically the classification of planetary terrain has been a largely manual, time-consuming process only performed by trained geologists/planetary scientists from downlinked image products, recent advancements in Computer Vision, and Deep Learning in particular, have enabled the automation of this task, both in the ground segment as an operational tool, but also from the flight segment as an on-board process. Due to communication lag between ground station and a rover on Mars, a rover is effectively expected to operate autonomously when given a series of instructions as real-time human operation is impossible - terrain classification, coupled with other conventional applications such as hazard detection and path planning, can greatly increase the safety margins of a rover which operates (for the most part) autonomously in a hazardous environment.

A bespoke dataset has been created by GMV for LHDAC project. It contains images taken from simulated lander from below 2 km altitude. The dataset was created using custom setup in Blender graphical modelling and rendering software paired with automation scripts written in Python. Since it is made to look accurate from around 50m altitude it could be adapted for rover point of view simulation. This was done by mainly increasing base ground mesh resolution at a cost of decreased terrain size. In addition, adaptive subdivision of terrain mesh was enabled. It is an algorithm that splits terrain mesh into smaller polygons near the camera, separately for each image. This results in dramatic increase of image fidelity at a cost of significant (+75%) render time increase. Randomized camera position parameters have also been modified to reflect camera position and orientation typical to rover.

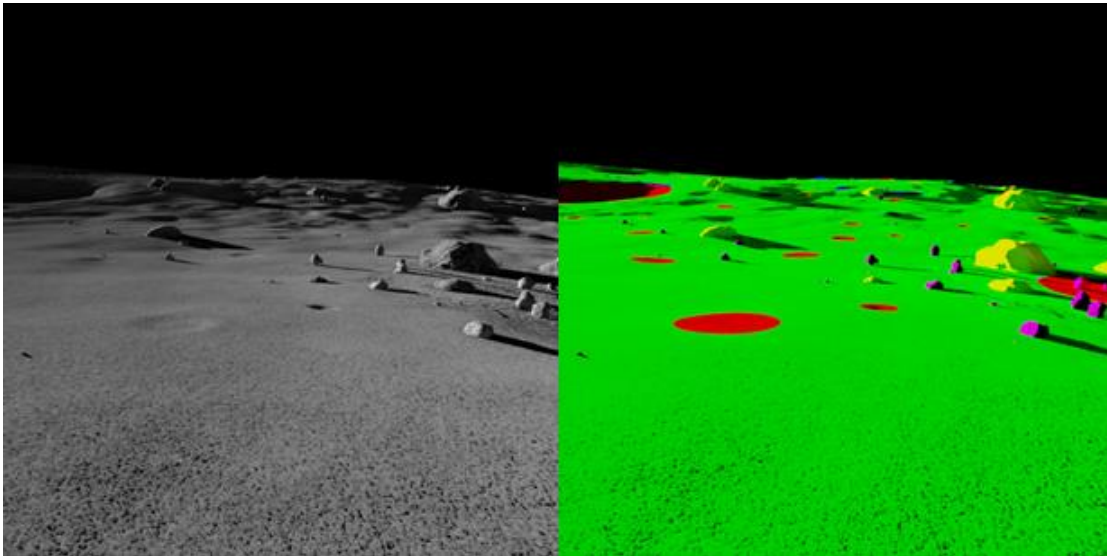


Figure 5-2: GMV POL LHDAC Synthetic Moon Dataset

5.2. AI4XR-UC-1.2 GENERATION OF TERRAINS BASED ON FRACTALS AND GENERATIVE ADVERSARIAL NETWORKS

Our XR terrain representation will support the use case of planetary terrain classification, with a specific focus on rover navigation and the associated AI methods. Considering the rover's limited range of motion, hierarchical structures are not required. Instead, we require a flexible approach to generate highly detailed terrain configurations that enable the creation of multiple training sets for effective AI navigation algorithm training.

The algorithms proposed for this use case must generate terrains surfaces that closely resemble realistic terrain patterns derived from actual datasets. Those datasets should be composed of example high maps (also called DEM) and their corresponding pixel-level labels, which, ideally, should identify the geological elements in the provided terrain. A parameter estimation subsystem will analyse the examples and it will reduce them into a set of well-designed terrain parameters, so another functional block will use them to procedurally recreate new terrains that adhere to these parameters. Furthermore, the software should allow for fine-tuning, enabling the generation of interesting variety of synthetic datasets for training navigation algorithms.

In this use case, two main software modules need to be considered.

The modules are for the synthetic terrain generator are:

- The first module, known as the Terrain Learning module, will focus on learning and evaluating the specific parameters that define the navigation scenario. Its primary purpose is to understand the characteristics of the desired terrain based on the images from the dataset that show a similar initial landscape. For example, a specific area on the surface of the Moon with distinctive topography, where impact craters and a certain distribution of boulders, rocks, lunar dust, as well as cracks on the ground can be observed and analysed to be reduced to a set of parameters that will be the seed to replicate the terrain with a myriad of variations in the next module.
- The second module is responsible for recreating the new virtual landscape and rendering the views that the rover's sensors will capture during navigation in synthetically generated terrains. These rendered views will serve as inputs to the navigation training algorithms designed for planetary terrain classification.
- The learning module will utilize the initial DEM information from input datasets to estimate the fractal dimensions of different areas within the provided sampled terrain, and other needed parameters to reproduce the virtual terrain. To facilitate analysis, the DEM will be partitioned into smaller tiles or patches of manageable size. The specific patch size may vary based on terrain complexity and desired level of detail.

6. AI-AIDED-XR ARCHITECTURE OVERVIEW

6.1. COMBINED ARCHITECTURE

The combined architecture for the entire system is depicted in Figure 6-1. The combined architecture is the highest level, offering an insight into how the interfaces communicate but black boxing the advanced internal components of each use case.

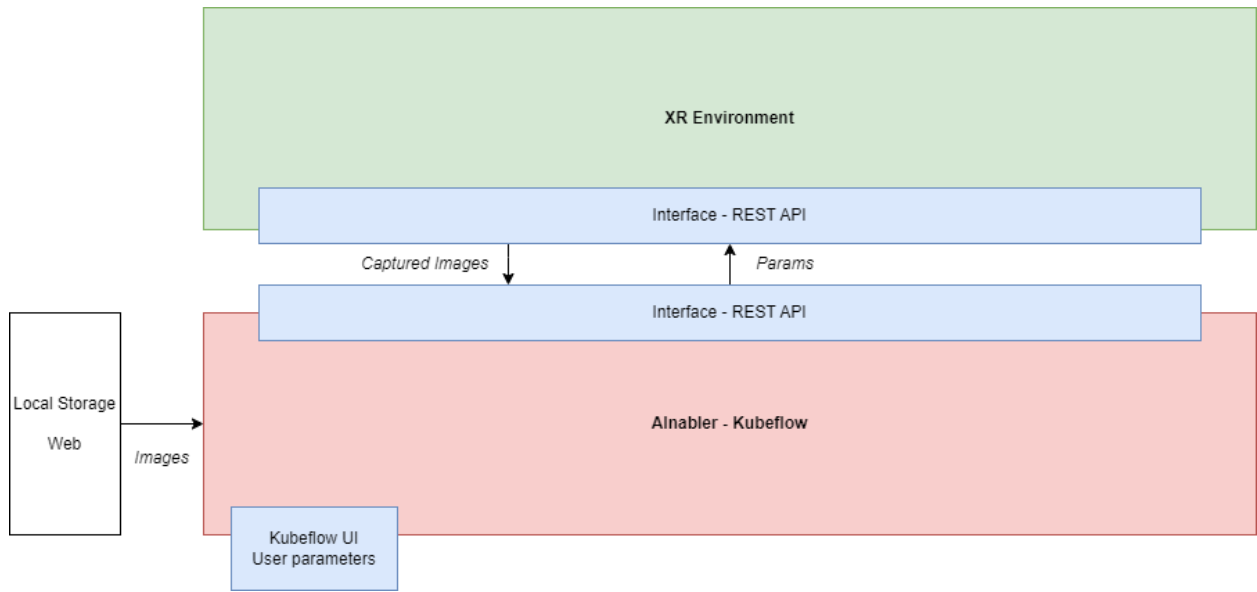


Figure 6-1: AI-Aided-XR Architecture

This architectural diagram encapsulates the loop that these two components form. Directional arrows can be observed coming to and going from both components to form a circle.

7. RESULTS

7.1. XR4AI-UC-1.6 PLANETARY TERRAIN CLASSIFICATION RESULTS

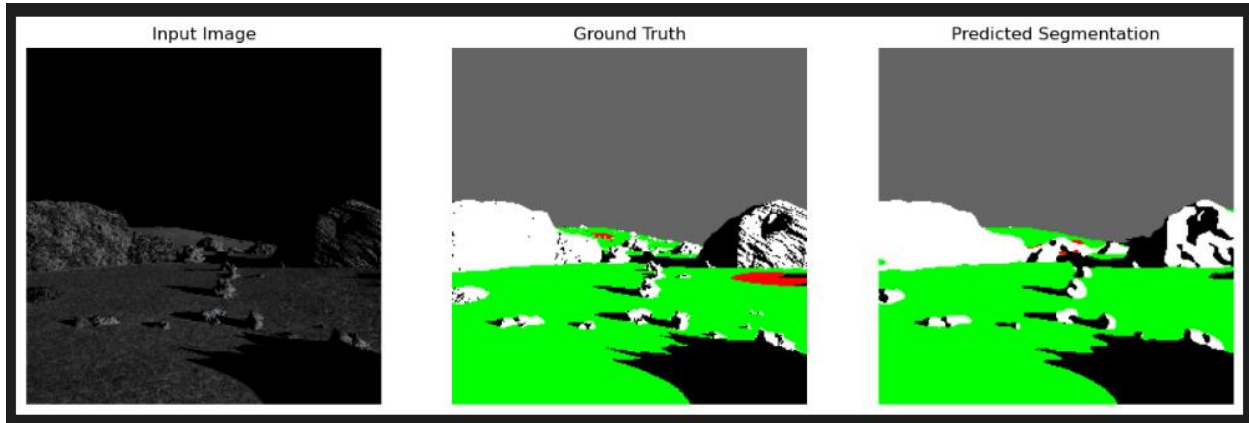


Figure 7-1: Results from the terrain classifier

Figure 7-1 illustrates the results obtained by the terrain classification component. These masks are received by the XR synthetic generator and fed through a DeepLabV3+ semantic segmentation model. The input images and predicted masks are all stored for later interrogation and to be fed back into the XR terrain learner, discussed in section 7.2.6 of this document.

7.1.1. METRICS

Table 3: Models and their metrics

Model Name	Loss	Accuracy	Mean IoU
model_epochs200_patience12_lr0.0001	0.308376	0.900723	0.800165
model_epochs150_patience10_lr0.001	0.340152	0.899886	0.800898
model_epochs250_patience15_lr0.0001	0.321353	0.905472	0.811696
model_epochs200_patience15_lr0.0005	0.345179	0.908623	0.816666
model_epochs150_patience12_lr0.0005	0.314730	0.910249	0.821123

The following metrics were defined to score the models against the test dataset.

- Loss
 - The loss metric determines the difference between predicted and actual values of the predicted images. The lower the accumulated loss, the better the performance.
- Accuracy
 - The accuracy metric measured the classification's performance as a percentage. The higher the value, the better the performance.
- Mean IoU
 - The mean IoU (Intersection over Union) evaluates the accuracy of a predicted mask by overlapping it with the ground truth mask. The IoU is the raw pixel difference between the pixels.

7.1.2. INFERENCE RESULTS

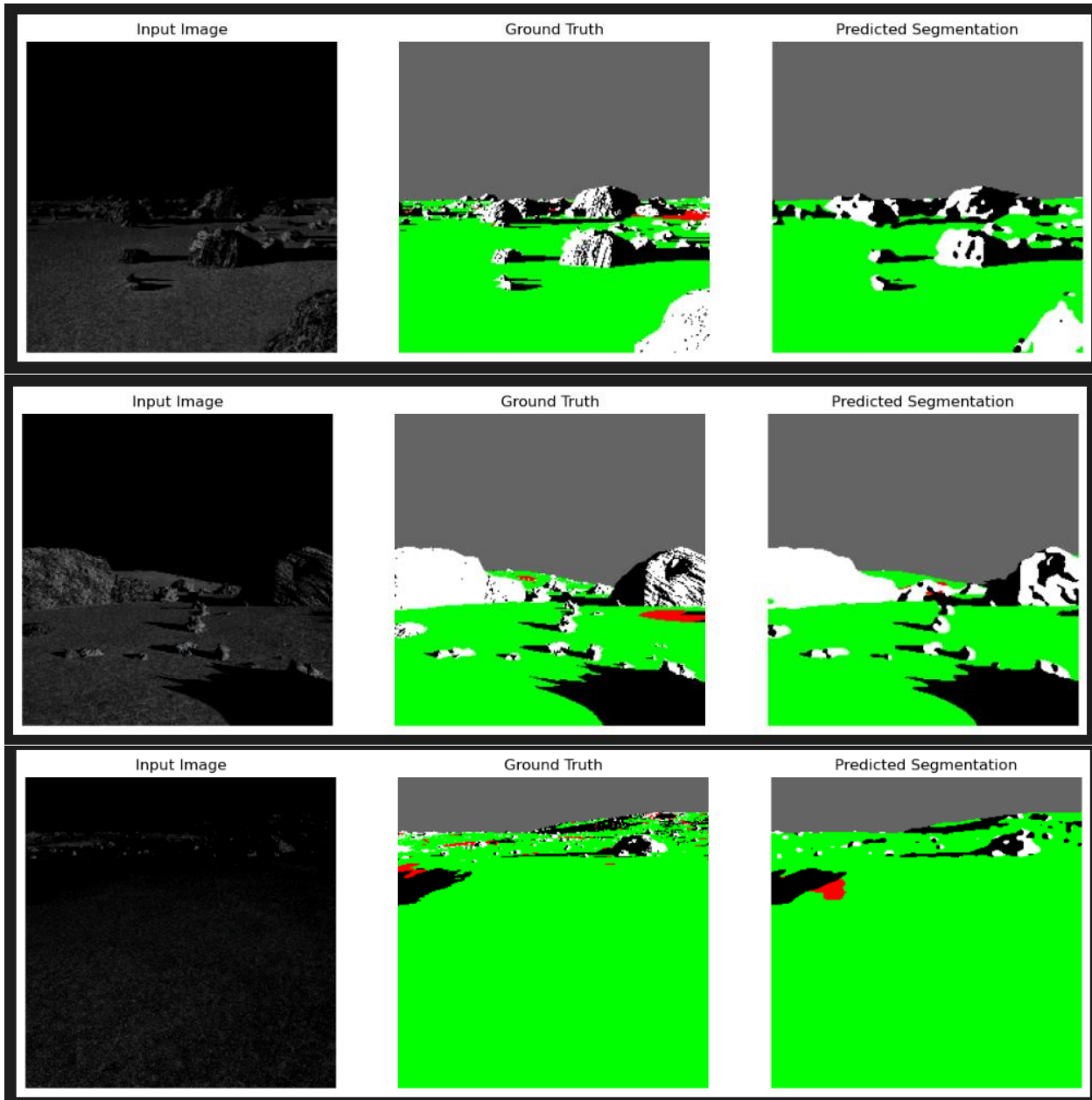


Figure 7-2: Inference Results on LHDAC Dataset

Figure 7-2 illustrates the results obtained by the terrain classification component. These masks are received by the XR synthetic generator and fed through a DeepLabV3+ semantic segmentation model. The input images and predicted masks are all stored for later interrogation and to be fed back into the XR terrain learner.

To complete the harmonic loop of AI and XR, the activity needed to demonstrate that it could use images obtained from the XR simulator to enhance the terrain classification capabilities. To demonstrate the success of this loop, we do the following experiment. Note that all these images were obtained while deployed on ESA's AINabler, using the hardware provided by that server.

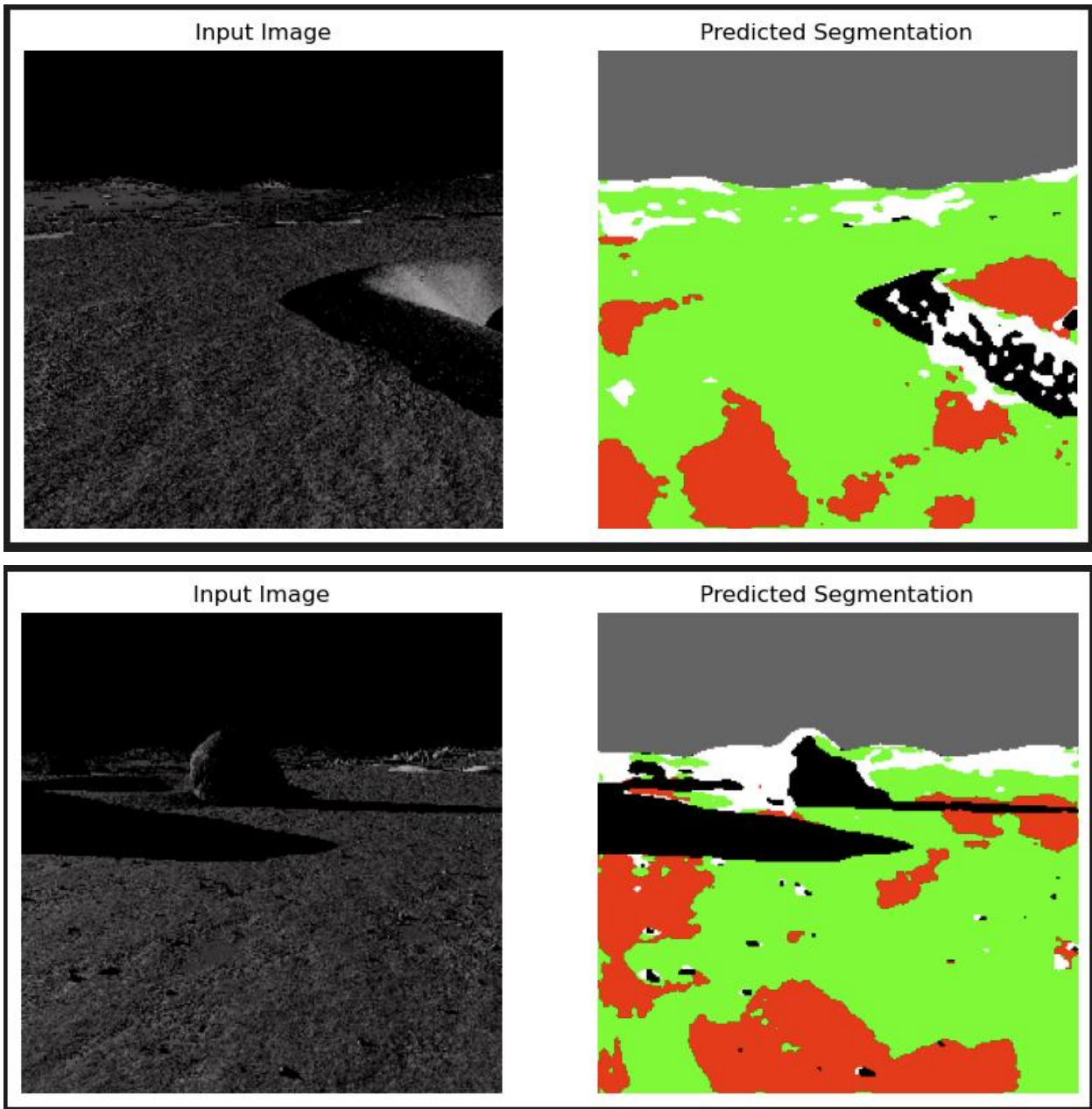


Figure 7-3: Results of Inference on Base Model

Figure 7-3 is the first step in the pipeline. A request is sent to the XR simulator using a REST API call – several images are specified to run segmentation on using the base model trained exclusively on the LHDAC dataset. The model makes a good attempt at running inference because the images are close to the baseline provided by the LHDAC corpus but overall has a poor union score.

To improve the performance of the base model and complete the symbolic loop between AI and XR, we implement a concept known as fine-tuning. This allows the model to train on new images without retraining the whole model.

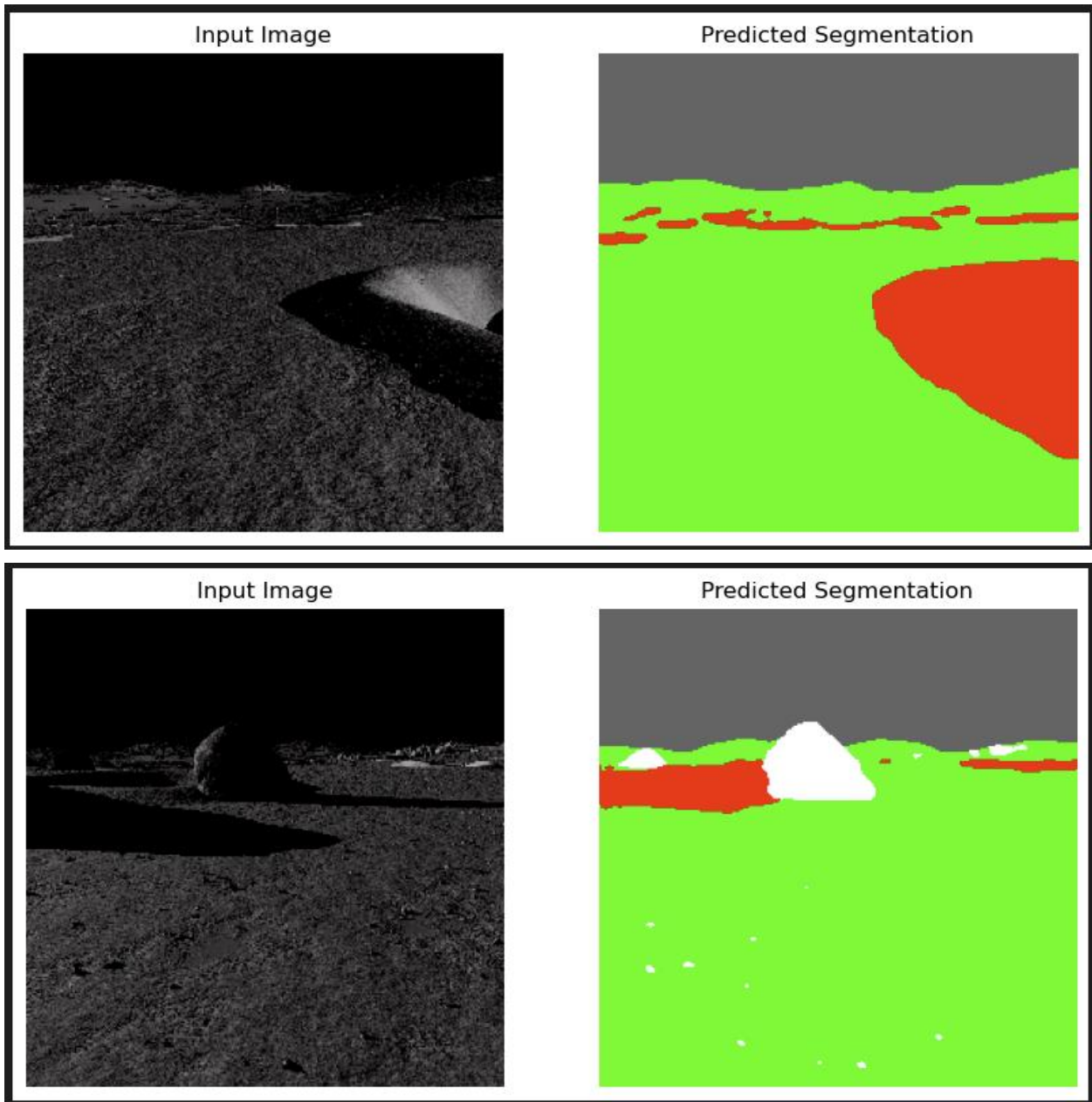


Figure 7-4: Results of Fine-tuning

Figure 7-4 illustrates the increased performance of the model after being fine-tuned on a sample of 200 images from the simulator.

7.1.3. AINABLER VISUALIZATION

When the operator accesses AINabler, they're presented with a UI to run the Kubeflow pipeline. Pipelines are automated workflows that orchestrate a sequence of tasks.

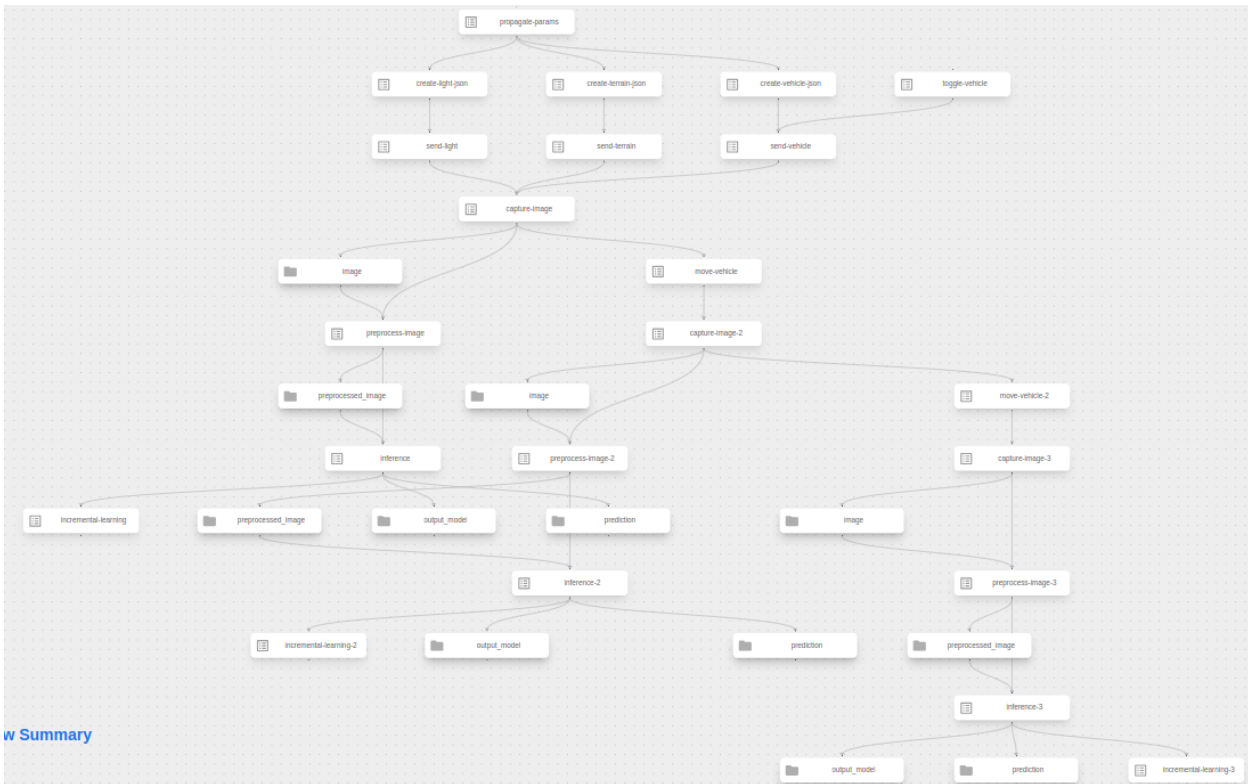


Figure 7-5: The AIAXR Pipeline

Figure 7-5 is the AIAXR pipeline. Each step in the DAG is its own containerized code which uses dependencies provided by the previous steps. Images and predictions can then be viewed in their corresponding steps by clicking on the folder icons, as demonstrated in **Figure 7-6**.

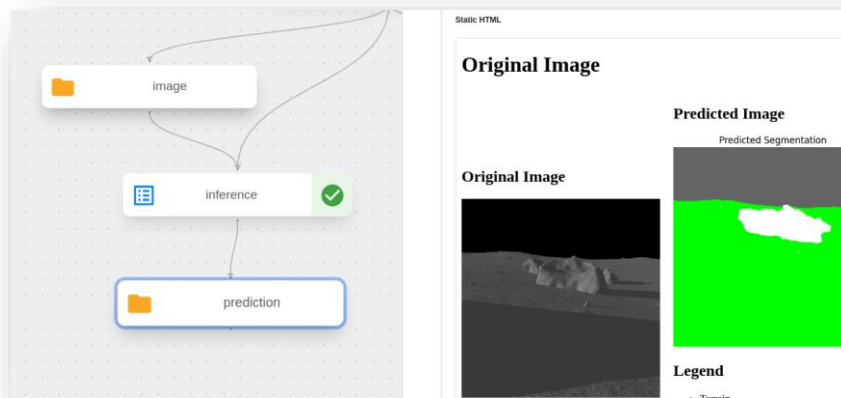


Figure 7-6: Visualized Predictions

7.2. AI4XR-UC-1.2 GENERATION OF TERRAINS BASED ON FRACTALS AND GENERATIVE ADVERSARIAL NETWORKS ARCHITECTURE RESULTS

7.2.1. NAVIGATION DATASET GENERATION

The first type of results to highlight would be obtaining sets of images along with their associated labels and the relative distance to the vehicle (at the pixel level). From these images, datasets can be created depicting the lunar vehicle's journey across multiple positions. The following images display an example of a sequence taking 4 positions in which the vehicle has advanced 5 meter in a straight line at each step. For each position, RGB images, associated labels, and the distance map (with maximum distance of 40m.) have been generated. Moreover, the heightmap of the terrain has been also rendered including the associated labels. In this case the heightmap shows a detail of 25 m. around the vehicle.

The example on the next page illustrates one traverse. More details and examples can be found in Error! Reference source not found..

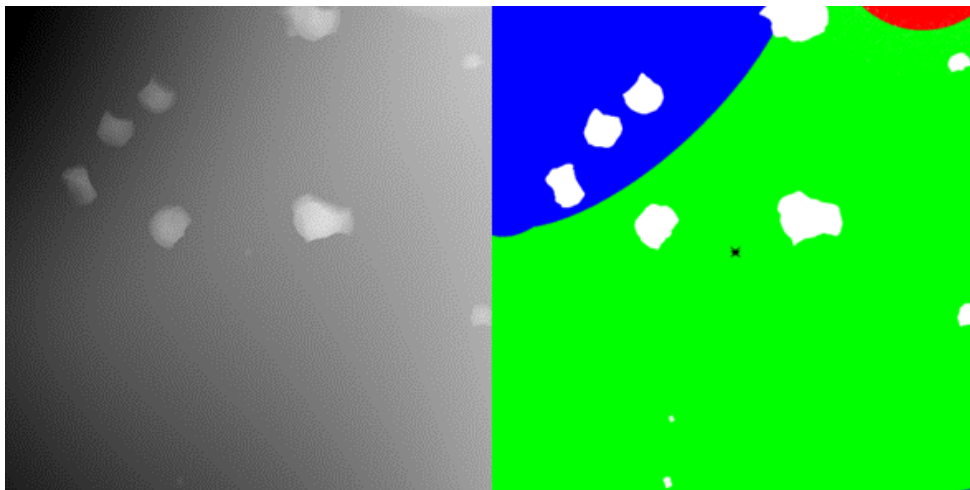


Figure 7-7: Example height map of a path

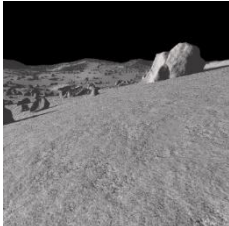
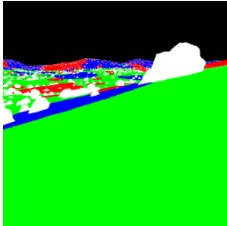
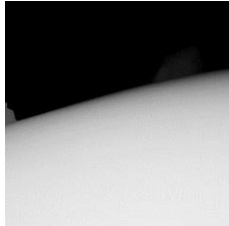
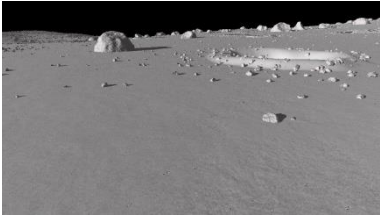
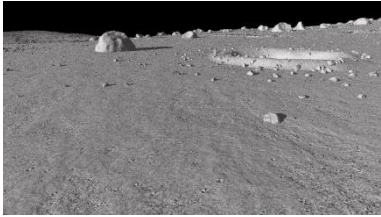
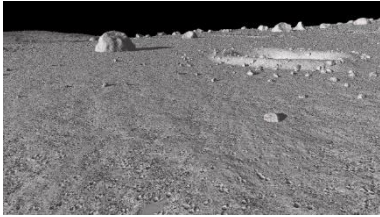
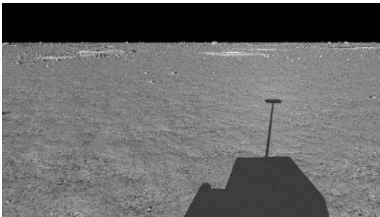
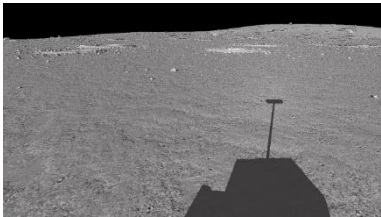

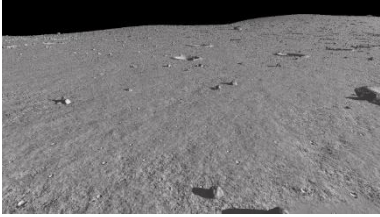
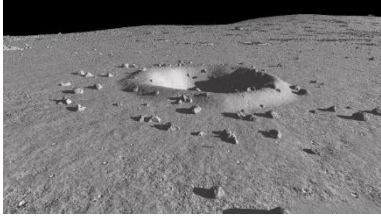
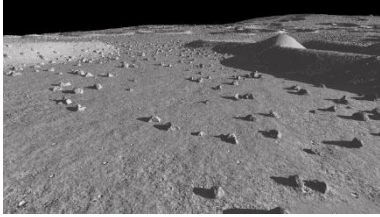
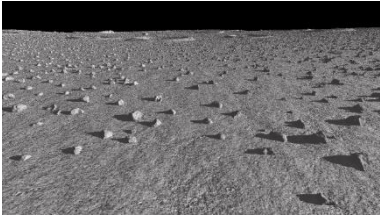
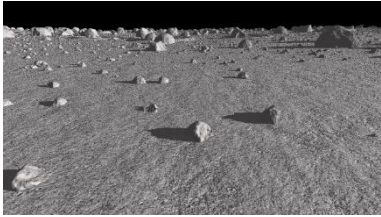
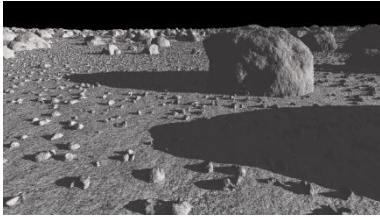
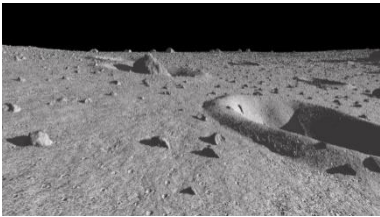
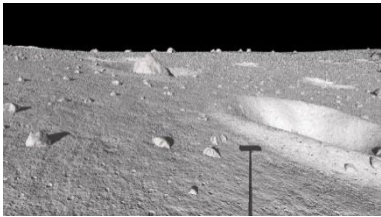
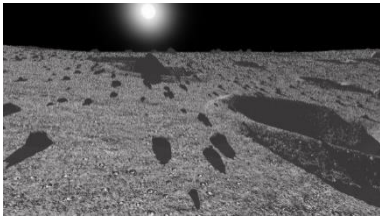
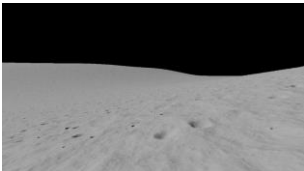
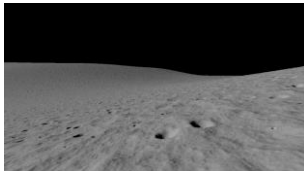
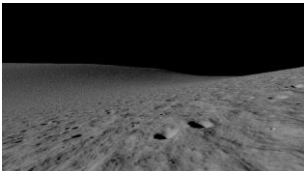
RGB image	Label image	Distance image
		

Figure 7-8: RGB image, label image and distance image from a single traverse section

7.2.2. CONFIGURABLE PARAMETERS

Figure 7-9 holds examples of the configurable parameters.

			
Flat and regular ground	Slightly irregular ground with a few small craters	Irregular ground with many small craters	
			
Planar terrain	Small hills terrain	Irregular terrain with mountains	
			
A few craters with small size	Crater with bigger size and small rocks around it	Many big craters with small rocks around them	
			
High number of small rocks	Medium size rocks	Combination of high number of small rocks and a high number of big rocks	
			
Light azimuth = 270° Light elevation = 20°	Light azimuth = 5° Light elevation = 20°	Light azimuth = 180° Light elevation = 10°	
	Light elevation = 45°	Light elevation = 20°	Light elevation = 10°
Diffuse			

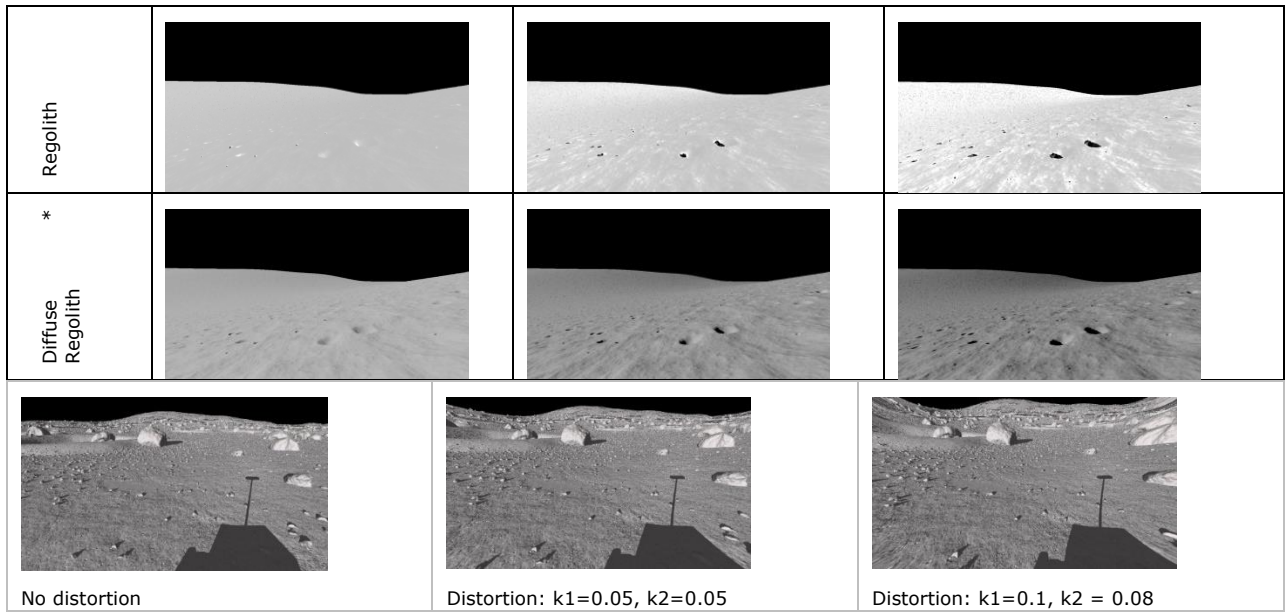


Figure 7-9: Example Images from Configurable Parameters

7.2.3. 360 IMAGE FOR XR VISUALIZATION

A potential use case for the image rendering engine would be the creation of 360-degree panoramic images. This type of image covers the entire field of vision of the observer, allowing the user to look in any direction Figure 7-10 illustrates this.

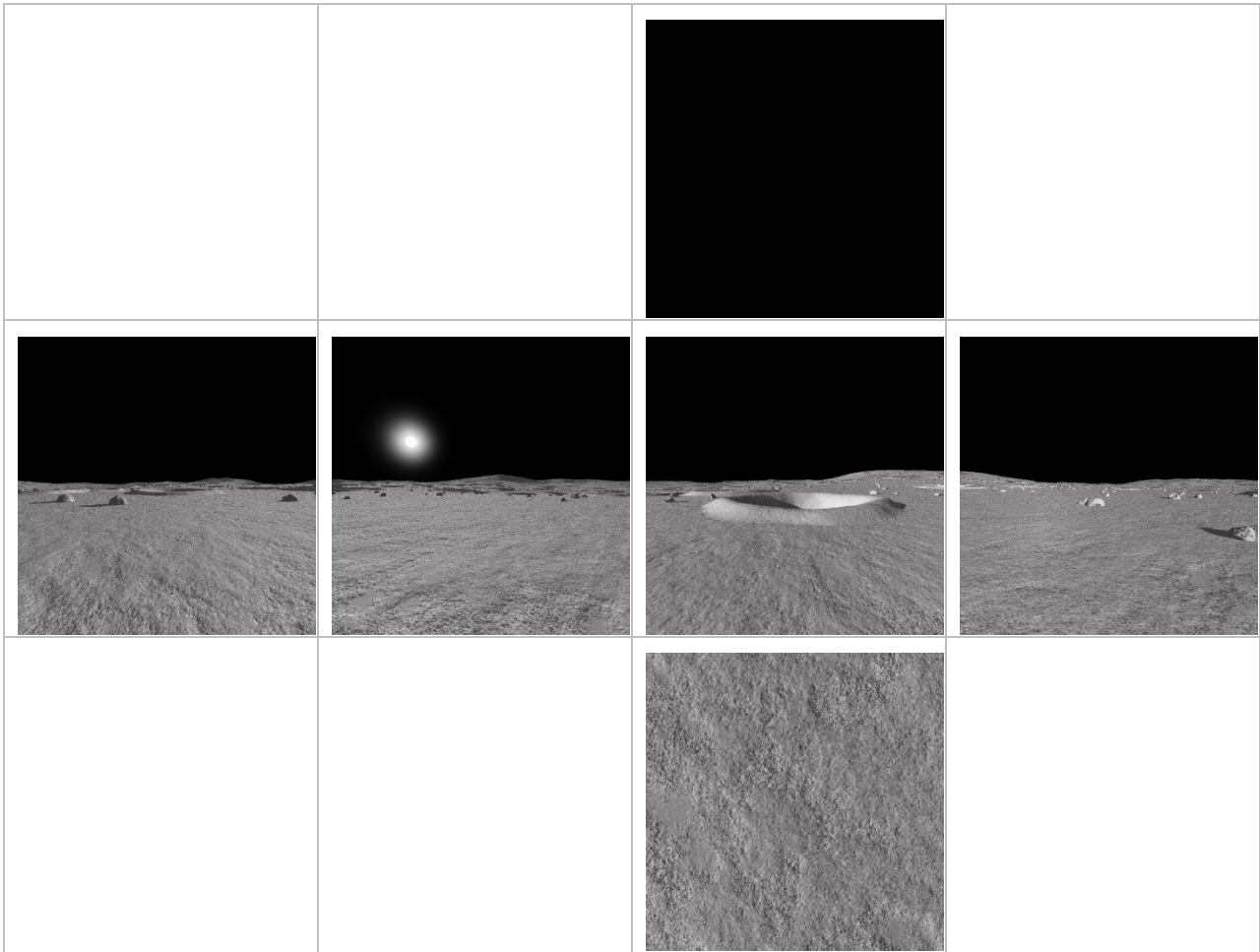


Figure 7-10: 360 Image Visualization

7.2.4. STEREO IMAGE FOR ANY VISUALIZATION DEVICE

Another potential use of the renderer is the creation of stereoscopic images for any type of display device. Figure 7-11 demonstrates this.

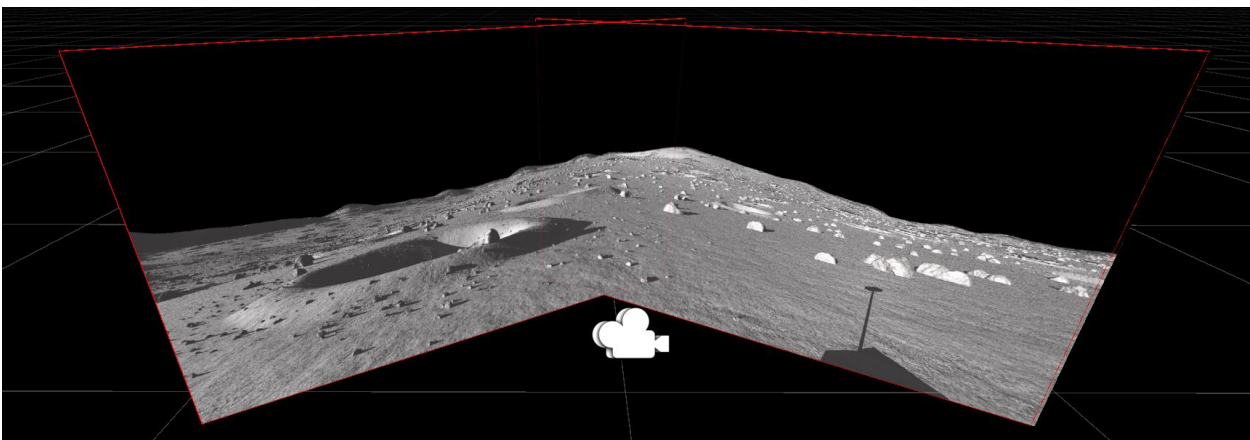


Figure 7-11: Stereo Imagery for VR Devices

7.2.5. TERRAIN EDITOR

Figure 7-12 illustrates the terrain editor GUI. As can be seen in the image, the editor has three main parts: the rendered image, the parameter configuration area, and a progress bar.

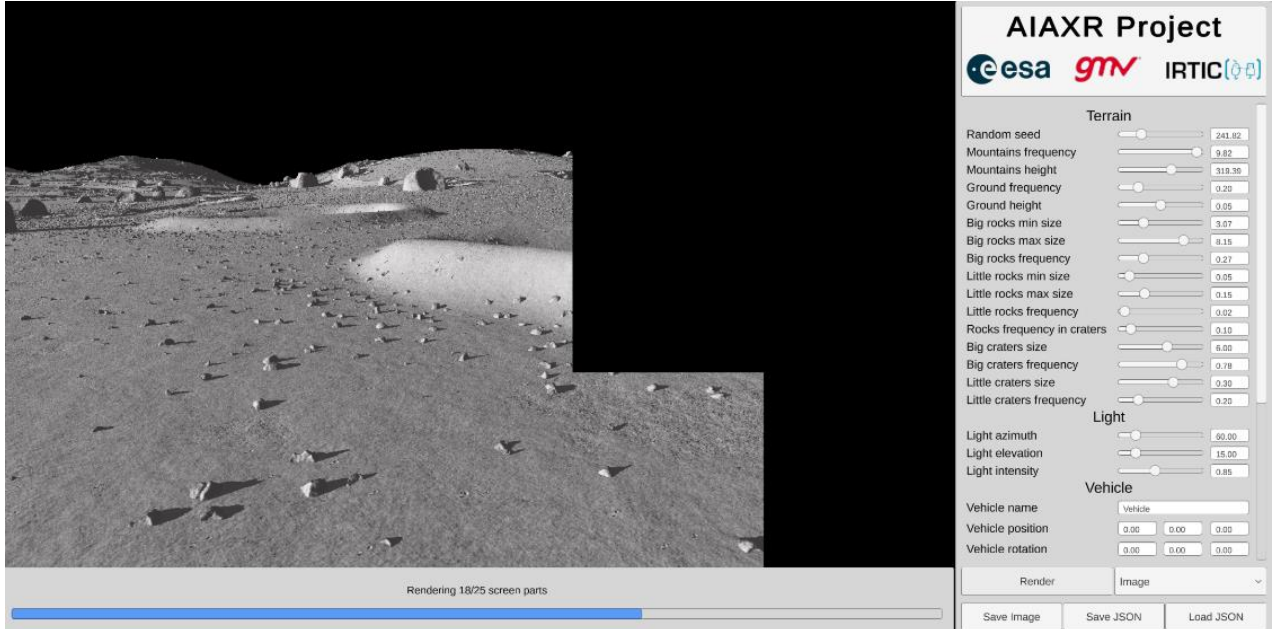


Figure 7-12: Terrain Editor GUI

Additionally, there is a combo box to change the type of image and an option to save them.

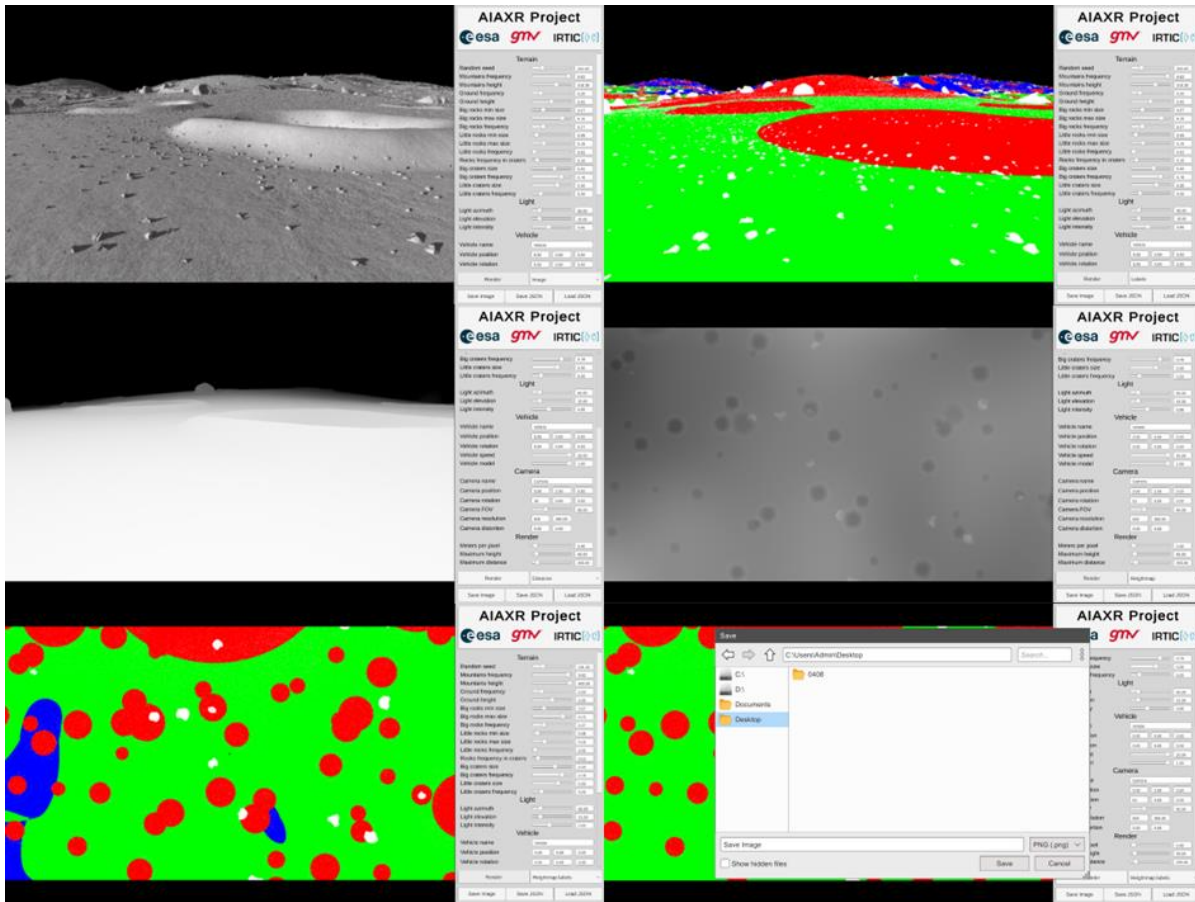


Figure 7-13: Different types of images and saving them

7.2.6. TERRAIN LEARNER

The terrain learner has been trained using 2700 height maps synthetically generated with different distributions of mountains, craters, and rocks. These images have a resolution of 512x512, which is the maximum resolution supported by the GPU training the system.

Two learning procedures have been developed: a parameter estimator and a classifier, both based on CNNs. The classifier aims to obtain a model that, once trained with synthetic images generated based on 3 parameters (M, C, R), can predict these parameters on new synthetic and real height maps of the lunar surface.

The class associated with each image is defined in the filename based on the parameters M, C, R with which the image was generated (Mountains, Craters, Rocks). The possible values of M, C, R are discrete and can also be 3, High/Med/None, resulting in a total of $3^3 = 27$ classes to classify. For example, the image "M_High_C_High_R_High_0.png" will have the class $3^2 * 2 + 3^1 * 2 + 3^0 * 2 = 18 + 6 + 2 = 26$.

After training, the loss function values obtained for each epoch can be seen in Figure 1. It can be observed how the system learns and reduces the error in each iteration.

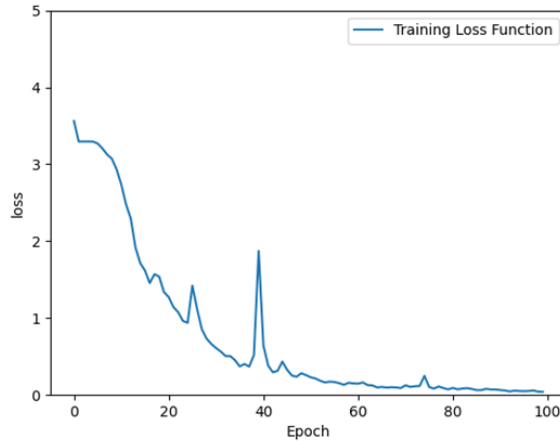


Figure 7-14– Training Loss function of the Moon Heightmap Classifier using CNN

We can observe how the values of the confidence matrix are around the diagonal. This indicates that the model is making accurate predictions in the majority of classes.

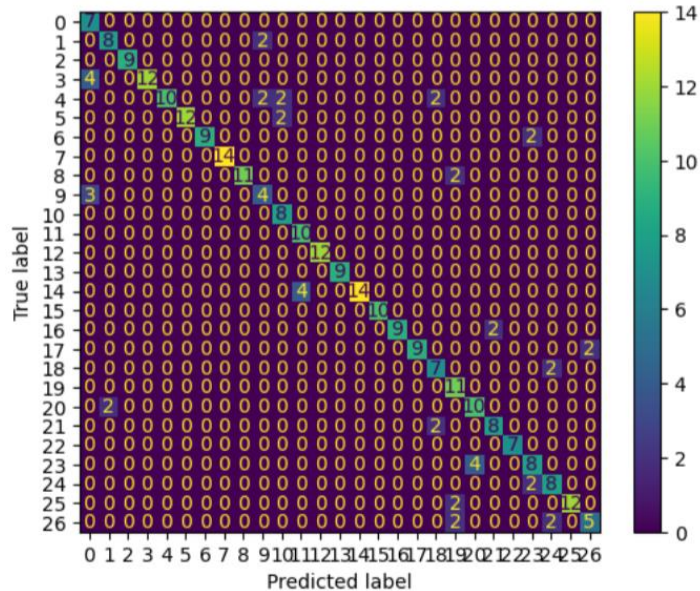


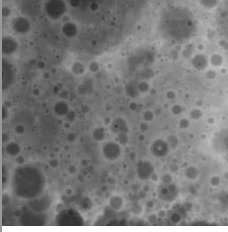
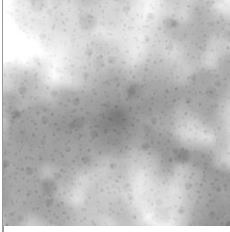
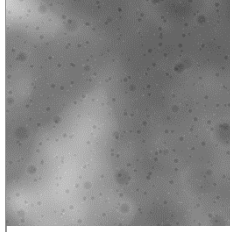
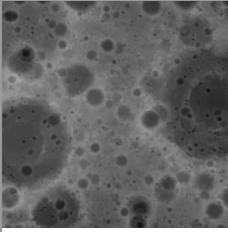
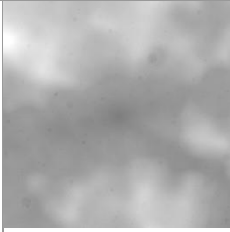
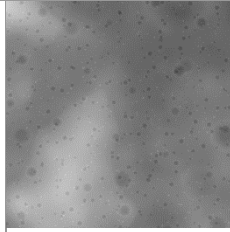
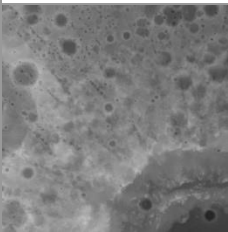
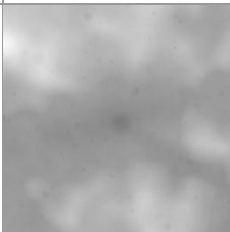
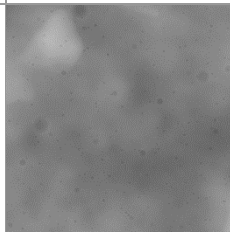
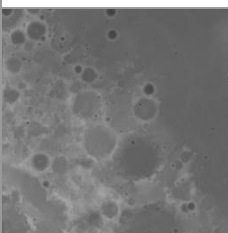
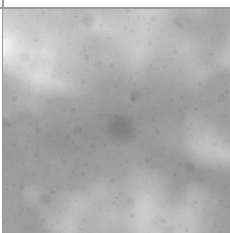
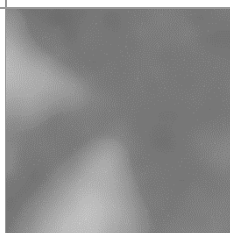
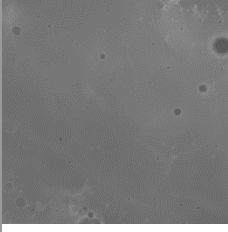
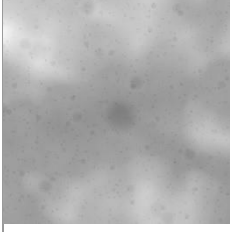
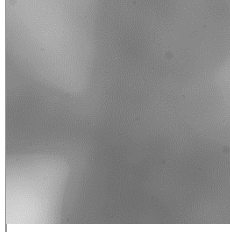
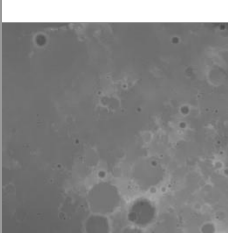
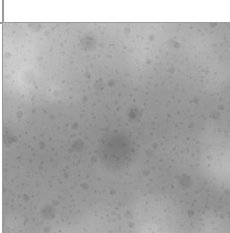
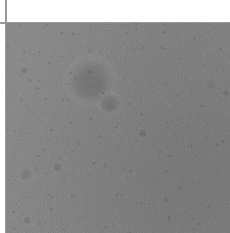
Figure 7-15 - Confidence matrix of the Moon Heightmap Classifier using CNN

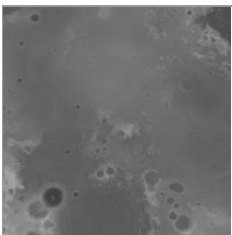
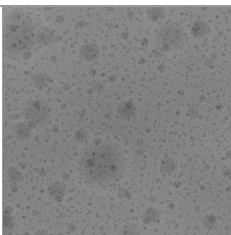
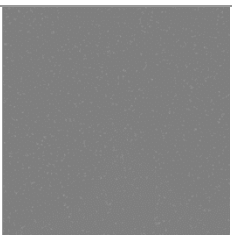
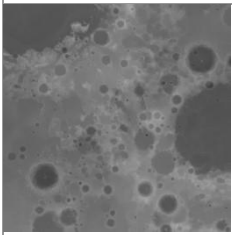
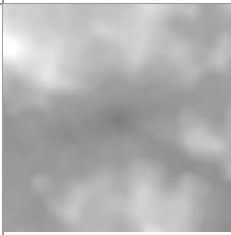
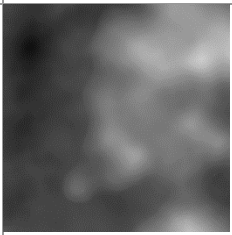
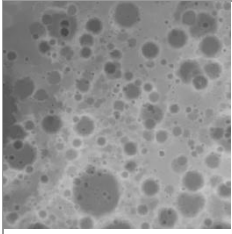
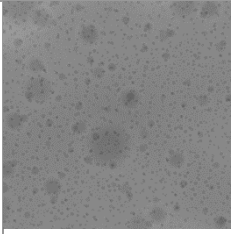
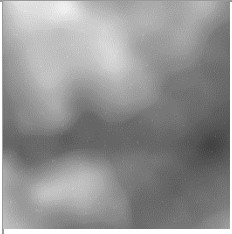
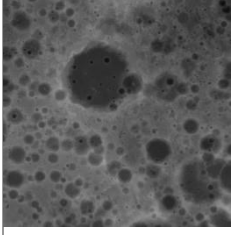
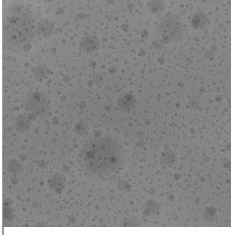
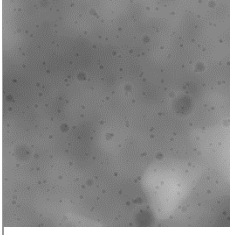
The results obtained in training the estimator are described below. In this model an attempt is made to estimate the parameters:

- MountainsFrequency (MF)
- MountainsHeight (MH)
- BigCratersFrequency (BCF)
- BigCratersSize (BCS)
- BigRocksFrequency (BRF)
- BigRocksMinSize" (BRmS)
- BigRocksMaxSize" (BRMS)

These are part of the values needed to define a terrain in the data generator.

After training both models, they have been tested on 10 real height map images of the lunar surface. The results obtained are shown in the table.

Input real DEM	Estimated parameters	Classification categories	Generated DEM from terrain learner	
			From estimated parameter	From classification categories
	"MF": -0.054, "MH": -7.80, "BCF": 1.07, "BCS": 12.03, "BRF": 0.62, "BRmS": 2.09, "BRMS": 12.03	M: high C: high R: high		
	"MF": 3.98, "MH": 41.62, "BCF": 1.00, "BCS": 10.19, "BRF": 0.38, "BRmS": 1.78, "BRMS": 7.15	M: high C: high R: high		
	"MF": -2.32, "MH": -36.65, "BCF": 0.80, "BCS": 9.60, "BRF": 0.71, "BRmS": 2.03, "BRMS": 13.34	M: high C: medium R: high		
	"MF": -0.10, "MH": -10.99, "BCF": 0.88, "BCS": 9.50, "BRF": 0.58, "BRmS": 1.85, "BRMS": 10.33,	M: medium C: none R: high		
	"MF": 1.76, "MH": 14.60, "BCF": 0.75, "BCS": 8.16, "BRF": 0.47, "BRmS": 1.73, "BRMS": 8.78	M: medium C: medium R: high		
	"MF": 0.92, "MH": 2.76, "BCF": 0.84, "BCS": 9.07, "BRF": 0.54, "BRmS": 1.87, "BRMS": 10.01	M: none C: medium R: high		

	"MF": 3.56, "MH": 36.15, "BCF": 0.90, "BCS": 8.94, "BRF": 0.35, "BRmS": 1.66, "BRMS": 6.44	M: none C: none R: high		
	"MF": 0.13, "MH": -9.04, "BCF": 1.05, "BCS": 11.21, "BRF": 0.56, "BRmS": 1.92, "BRMS": 10.38	M: high C: none R: medium		
	"MF": -10.72, "MH": -145.05, "BCF": 1.17, "BCS": 13.69, "BRF": 1.14, "BRmS": 2.22, "BRMS": 19.74	M: high C: none R: high		
	"MF": 2.92, "MH": 25.99, "BCF": 1.02, "BCS": 10.28, "BRF": 0.32, "BRmS": 1.71, "BRMS": 5.83	M: high C: high R: none		

8. NEXT STEPS

The AI-Aided-XR activity has unveiled numerous opportunities for further exploration. Our efforts have shed light on the potential for extensive synthetic data generation (an area rich for research in which computer vision applications in the space domain are in dire need) supporting ESA activities. Leveraging its integration with AINabler, the activity also explores MLOps using ESA infrastructure, opening avenues for enhanced operational efficiency and innovation. Using XR to generate these terrains on ESA hardware further amplifies the potential future use cases because essentially any conceptual environment can now be generated with a high enough fidelity that computer vision models are now proven to efficiently train on them.

The AI-Aided-XR project has successfully achieved TRL4. This is proven by the demonstration in the final review and demonstration, as well as the successful deployment to ESA hardware on both AINabler and a server for the XR environment to be hosted. It thus logically follows that the next major activity would expand upon this TRL from a range of 5 or 7, incorporating the technology in a demonstration in a real environment. Previous activities approached this possibility by interfacing directly with ESA's MOI to enhance operation support. AI-Aided-XR is more siloed, running solely on AINabler with no need to interact with internal ESA data.

To achieve TRL 7, knowledge from ViBEKO could be integrated into this system to re-use the interfaces and components which pull down from the FIM automatically and on a routine schedule to achieve a number of operational utilities – an interesting concept is an automated daily download to feed as data into the incremental learner component for AI-Aided-XR. The combination of both ViBEKO and AI-Aided-XR also demonstrates the commitment the consortium has to the lineage of ESA's AI Roadmap forming the next logical step in that roadmap.

AI-Aided-XR could also inspire future projects which serve as additional vision-based use cases (though it's important to note, a lot of the use cases could also easily be substituted here), most notably use cases which:

Increase situational awareness for both rovers and navigators.

Improve scientific return of missions by introducing computer vision into the loop, catching elements which may be missed by the human eye.

Improve telemanipulation feasibility (which, coupled with other projects which specialize in manipulation or control, could greatly expand the scope of the computer vision use).

Greatly improve planning tasks and save time spent on them.

Additionally, the synthetic generation techniques pioneered in this activity could benefit ESA in curating niche datasets for different missions. Rendering Mars scenery as opposed to the moon could be used retroactively to boost the already high accuracy of ViBEKO's terrain classifier. The synthetic generation techniques could:

Generate diverse datasets tailored to specific mission requirements.

Create realistic simulations of extraterrestrial environments for training and testing purposes.

Facilitate the development of advanced algorithms or bespoke training regiments for astronauts.

AI-Aided-XR owes a lot of its AI capabilities to the AI platform designed in the scope of the AI4Ops activity. Because of this inherent integration with ESA infrastructure, it makes it an ideal candidate for pushing development in the agenda outlined in the Artificial Intelligence for Automation (A²I) roadmap.

9. SUMMARY

AI-Aided-XR is an activity focused on investigating novel applications of the cyclicity between XR and AI. Through a process of gathering user stories and use cases and consulting experts in the space industry, two uses cases (one for AI, one for XR) were selected for implementation:

- XR4AI-UC-1.6 Planetary Terrain Classification
 - The AI, served by XR, focusing on planetary terrain classification by employing state-of-the-art semantic segmentation techniques. Google's DeepLabV3+ is employed as the base model, and a custom dataset synthesized by GMV POL, bespoke for lunar use cases, is used to train it. The use case is explored in greater detail in [AD. 1].
- AI4XR-UC-1.2 Generation of Terrains based on Fractals and Generative Adversarial Networks
 - The XR, served by AI, focusing on synthetically generating high-fidelity lunar terrain using fractals. The implementation includes tens of configurable parameters, illustrated in the architecture section of [AD. 1], and uses an AI terrain learner to learn environments based on images sent to it by XR4AI-UC-1.6. The images captured in this environment is sent to XR4AI-UC-1.6, effectively closing the AIXR loop.

Both uses cases were deployed on ESA hardware on separate machines, communicating with a REST API. XR4AI-UC-1.6 is deployed on ESA's AINabler platform, defined in the AI4Ops activity [RD. 1] to facilitate further development and inherit ESA infrastructure. AI4XR-UC-1.2 is deployed on a machine with sufficient graphics capability to fulfil the virtual environment rendering requirements.

Test cases for the implementation are available in [AD. 2], and lessons learned are discussed in length in [AD. 3]. For a more comprehensive breakdown of the project, readers are encouraged to read the final report [AD. 1].



VNIVERSITAT
D VALÈNCIA

Code:	GMV-AIAXR-ESR
Date:	20/03/2025
Version:	2.0
Page:	28 of 28

END OF DOCUMENT