AI Enabled Computer Vision Framework for Automated Knowledge Extraction in Planetary Rover Operations

Steven Kay (skay@gmv.com), Maciej Quoos, Robert Field, Maciej Józef Prokopczyk, Matteo De Benedetti, Fakher Mohammad, Maciej Szreter, Yang Gao, Evridiki V. Ntagiou

ASTRA 2023

Leiden, Netherlands

INNOVATING SOLUTIONS

UNIVERSITY OF SURREY

Contents

- Introduction & Concept
- AI-Enabled CV Framework for Rover Operations
- Software Prototype Overview & Results
- Conclusions

Introduction

ViBEKO: Vision Based Knowledge Extraction using Artificial Intelligence

Objectives of the study:

1. Creation of an extensible AI-CV Framework which includes MLOps capabilities
2. Integration of the AI-CV Framework alongside existing Mission Operations tools
3. Implementation of two AI-based software prototypes addressing relevant Use Cases to demonstrate the automation of Knowledge Extraction for Rover Operations
   - #1: Planetary Terrain Classification
   - #2: Global Localisation

Why is AI-based automation important?

- Increase the efficiency of short-term/tactical activity planning tasks
- Improve operator situational awareness
- Increase the throughput of rover telemetry processing
- In general, it may provide better generalisation vs. classical Computer Vision methods

AI-CV Framework Architecture

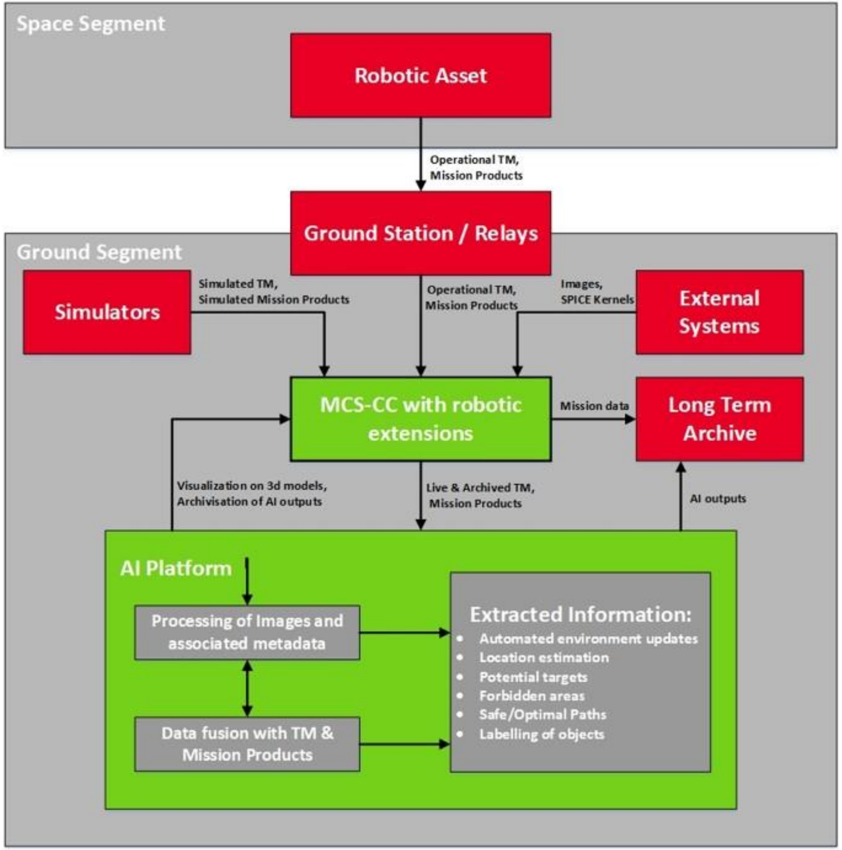
Architecture consists of two main components of interest in this activity:

- Mission Control System (MCS) → ONE-CC
- AI Platform → ESA AInabler platform

Workflow follows the classical approach in the processing of spacecraft telemetry

Selection of the MCS component allows for future integration of other commonly used tools

Software extensions created, using REST API, to allow the exchange of data products between the MCS and AI Platform



AI Platform based on ESA AInabler

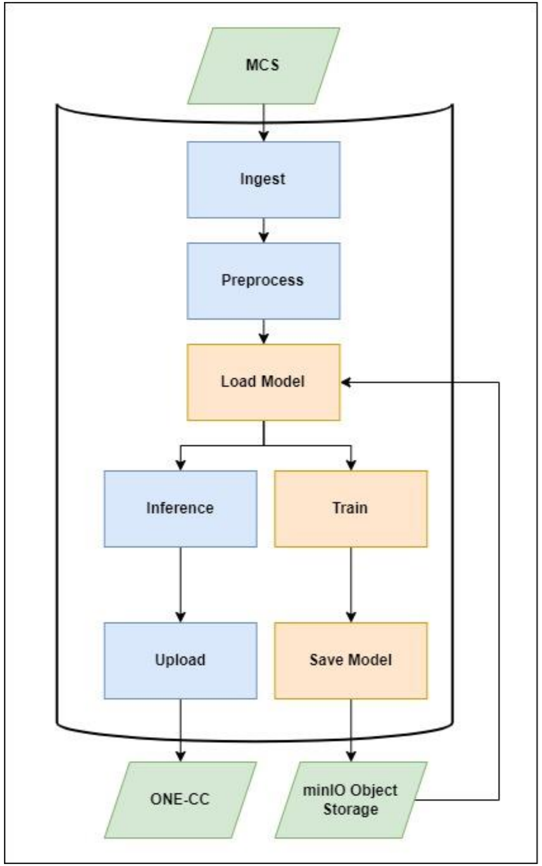
Platform-as-a-Service (PaaS) infrastructure hosted by ESOC, established by the ‘AI4Ops’ activity

Based on Kubeflow, an Open-Source MLOps platform for Kubernetes-based container orchestration

Kubeflow Pipelines approach selected for ML workflow modelling, providing:

- End-to-End Orchestration
- Easy experimentation
- Easy reuse

Individual operations/algorithmic steps modelled as tasks, connected in a graph.



SW Prototype 1: Terrain Classification

Model Overview

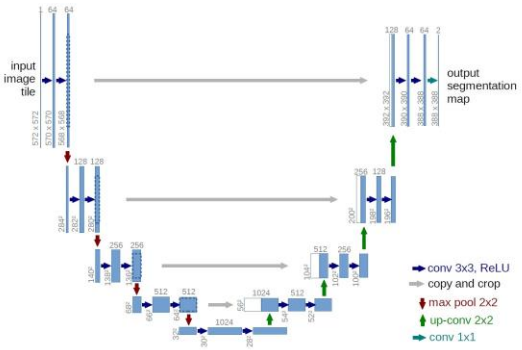
This Use Case aimed to consider an AI-based Terrain Classification model in the AI-CV Framework performing automated segmentation of provided rover images

Two Encoder-Decoder architectures were selected for implementation:

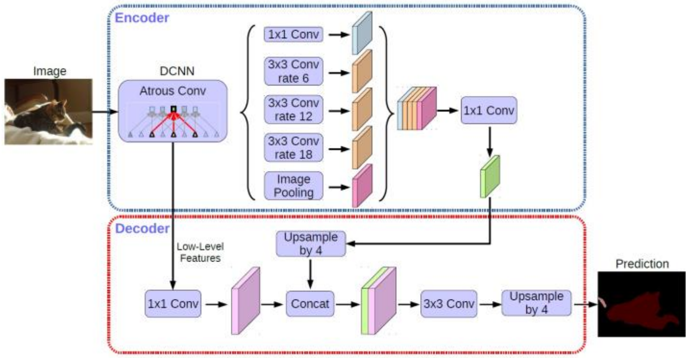
- U-Net
- DeepLabV3+

Several data processing techniques were evaluated and included as pre-processing steps

For training, conventional data augmentations (crop, flip, random zoom) were in addition to the GAN-generated images.



U-Net Architecture



DeepLabV3+ Architecture

SW Prototype 1: Terrain Classification

Datasets

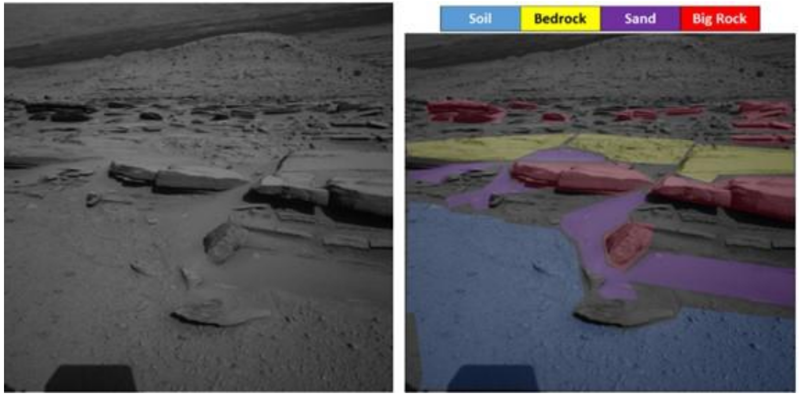
Selected the AI4Mars dataset, MSL subset ~16k images, 90% training, 10% validation. Three hold-out test sets of 322 images held.

Labels composed of: [Soil, Bedrock, Sand, Big Rock], with quite different proportions.

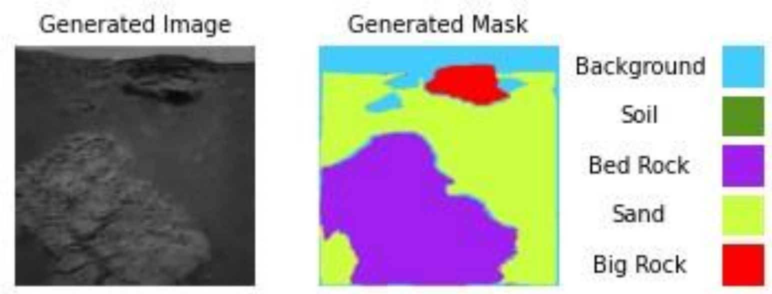
We considered a fifth class was considered for other labelled pixels: rover hardware, horizon etc.

Use of SemanticStyleGAN model as an augmentation to generate further image data and corresponding masks.

Augmented datasets G1, G2 created containing ~4k images.



Example Image and Mask from AI4Mars Dataset (Courtesy: NASA)



Example of generated image and mask using SemanticStyleGAN

SW Prototype 1: Terrain Classification

Results & Discussion

Overall, DeepLabV3+ significantly outperformed U-Net in both pixel accuracy and inference times, but achieved similar mIoU scores on testing sets

DeepLabV3+ achieved a maximum segmentation accuracy of 95% on the M3 hold-out subset of AI4Mars

DeepLabV3+ struggled to identify the “Big Rock” class, even with augmentation to increase the number of examples

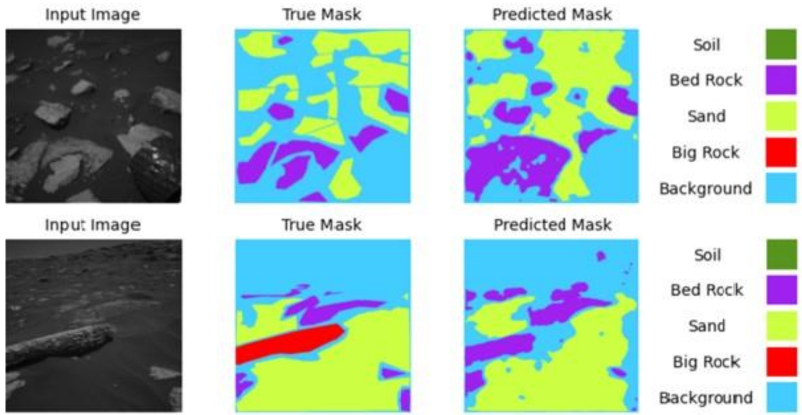
Use of the GAN-generated set (G2) was shown to improve the mIoU of the DeepLabV3+ model by ~3%

U-Net and DeepLabV3+ comparison results

Model	Backbone	Test set	Acc	Infer. time	mIoU	IoU			
						Soil	Bedrock	Sand	Big Rock
U-Net	MobileNetV2	M1	0.80	33	0.56	0.76	0.63	0.74	0.11
		M2	0.86	32	0.54	0.77	0.53	0.77	0.10
		M3	0.92	32	0.45	0.63	0.36	0.71	0.08
U-Net	VGG16	M1	0.82	77	0.57	0.77	0.62	0.75	0.12
		M2	0.88	78	0.55	0.78	0.52	0.79	0.12
		M3	0.92	78	0.45	0.64	0.35	0.71	0.10
DeepLabv3+	Resnet50	M1	0.84	**4**	0.57	0.78	0.64	0.77	0.10
		M2	0.90	**4**	0.55	0.79	0.54	0.79	0.08
		M3	0.95	**4**	0.44	0.63	0.37	0.69	0.07
DeepLabv3+	Resnet101	M1	**0.85**	10	**0.58**	0.78	0.66	0.77	0.09
		M2	**0.91**	10	**0.55**	0.79	0.54	0.80	0.08
		M3	**0.95**	10	**0.45**	0.64	0.35	0.72	0.07

DeepLabV3+ results using GAN generated images. Backbone = ResNet50, image size = 256x256

Model	Trained on	Test set	Acc	mIoU	IoU			
					Soil	Bedrock	Sand	Big Rock
DeepLabv3+	AI4Mars	M1	0.84	0.57	0.78	0.64	0.77	0.10
		M2	0.90	0.55	0.79	0.54	0.79	0.08
		M3	0.95	0.44	0.63	0.37	0.69	0.07
DeepLabv3+	AI4Mars+G1	M1	0.86	0.58	0.78	0.66	0.78	**0.11**
		M2	0.91	0.55	0.78	0.54	0.79	**0.1**
		M3	0.95	0.44	0.63	0.35	0.7	**0.08**
DeepLabv3+	AI4Mars+G2	M1	0.86	**0.61**	0.78	0.66	0.77	0.09
		M2	0.91	**0.58**	0.79	0.54	0.80	0.06
		M3	0.95	**0.47**	0.62	0.35	0.71	0.05



SW Prototype 2: Global Localisation

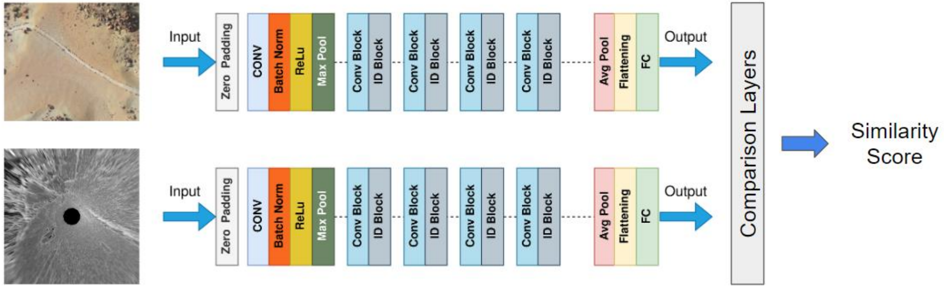
Model Overview

Based on a Siamese Architecture with 2 parallel CNNs

The two ResNet50 chosen as backbone receive as input the rover's reprojection and an orbital image, both covering a 50x50m aerea at 128px resolution.

Several preprocessing steps to compute the reprojections and filter data.

Output predictions as a confidence score. Best candidate selected from N to signify the most likely global position.



Evaluated backbone CNN selection over common pretrained nets: ResNets, VGG, DenseNet, MobileNet with retraining at increasing depths

Evaluated various image resolutions, colour, and image type (raw rover frame or reprojection)

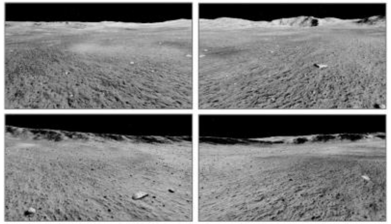
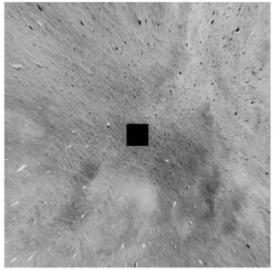
SW Prototype 2: Global Localisation

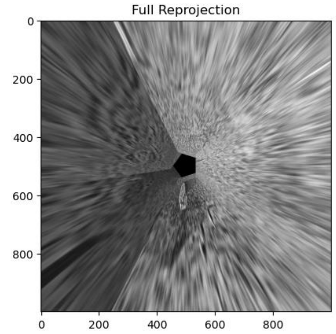
Datasets

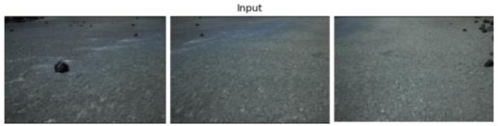
Considers both Real-World and Synthetic datasets for training → Main contribution of this work was to develop a pipeline based on a DL approach that could run well on realistic data not seen before

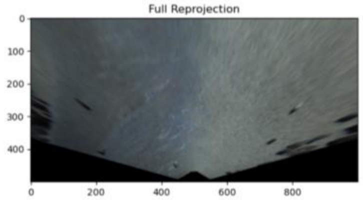
1. Synthetic datasets Moon generated in Unreal Engine, based on a NASA FDL Activity, using 4 rover frames.
2. CSA ENAV: Outdoor rover in Canada. 11 NavCam frames
3. ESA PRL: Tenerife field tests with ESA HDPR rover. 3 NavCam frames

Preprocessing to reproject rover camera frames into an <u>orthoimage</u> to match with <u>orbital</u> data







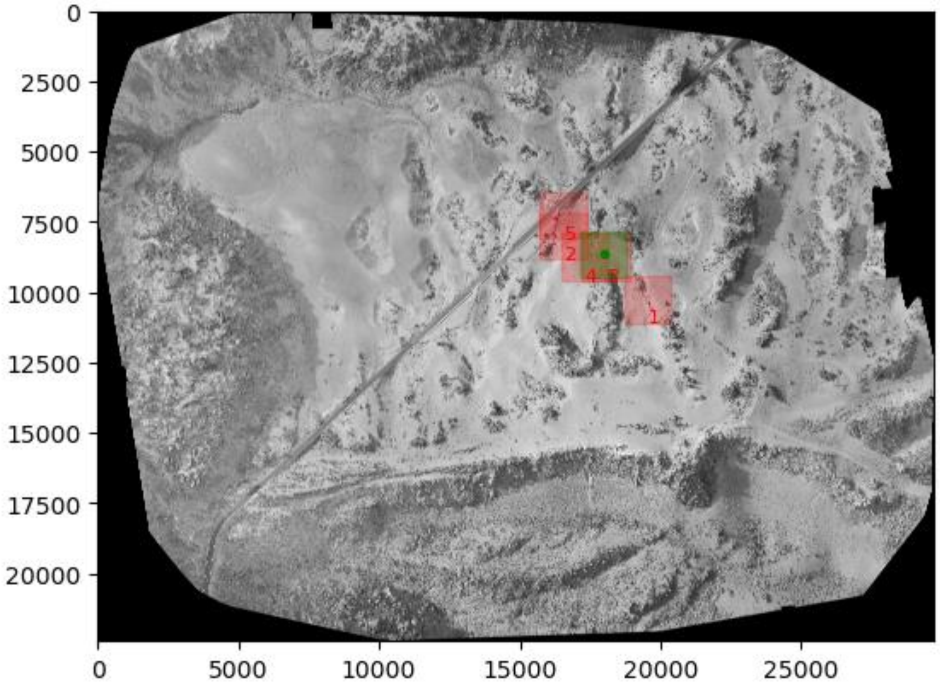
SW Prototype 2: Global Localisation

Results & Discussion

Results analysed on 2 main metrics
- Regular Accuracy of the SNN model
- Position estimation accuracy on the orbital DEM based on rover GT

Quantitative comparison with other Image Matching techniques:

SNN Evaluation with Random Sampling, SSD, and SAD, Position Score Backbone: ResNet50 until last classifier. Epochs: 50. Steps: 500. Dataset: PRL			
Random Sampling	**Sum of Squared Differences (SSD)**	**Sum of Absolute Differences (SAD)**	**Position Estimation Score**
186.66956747	149.69381958	114.90633132	58.6318445



Conclusions

ViBEKO established a new AI-CV Framework to automate knowledge extraction for Planetary Rover Operations

ViBEKO built on, and extended, existing ESA Mission Operations tools, combing an MCS, Robot Control Environment and an AI Platform

Two Software Prototypes were implemented and validated using the AI-CV Framework, demonstrating the overall concept.

The Terrain Classification prototype, in particular, demonstrated competitive results compared with the State-of-the-Art, and considered a new approach to data augmentation on the AI4Mars dataset

In general, both Software Prototypes suffered due to a lack of good, labelled and consistent datasets for training → The ESA "AI-Aided-XR" activity currently underway, is looking to address this for the Terrain Classification Use Case in particular.

There remains some visualisation integration work to be completed to fully view the ML outputs from the implemented Software Prototypes in the 3DROCS environment

Thank you

Steven Kay (GMV-UK)
skay@gmv.com

INNOVATING SOLUTIONS

UNIVERSITY OF SURREY