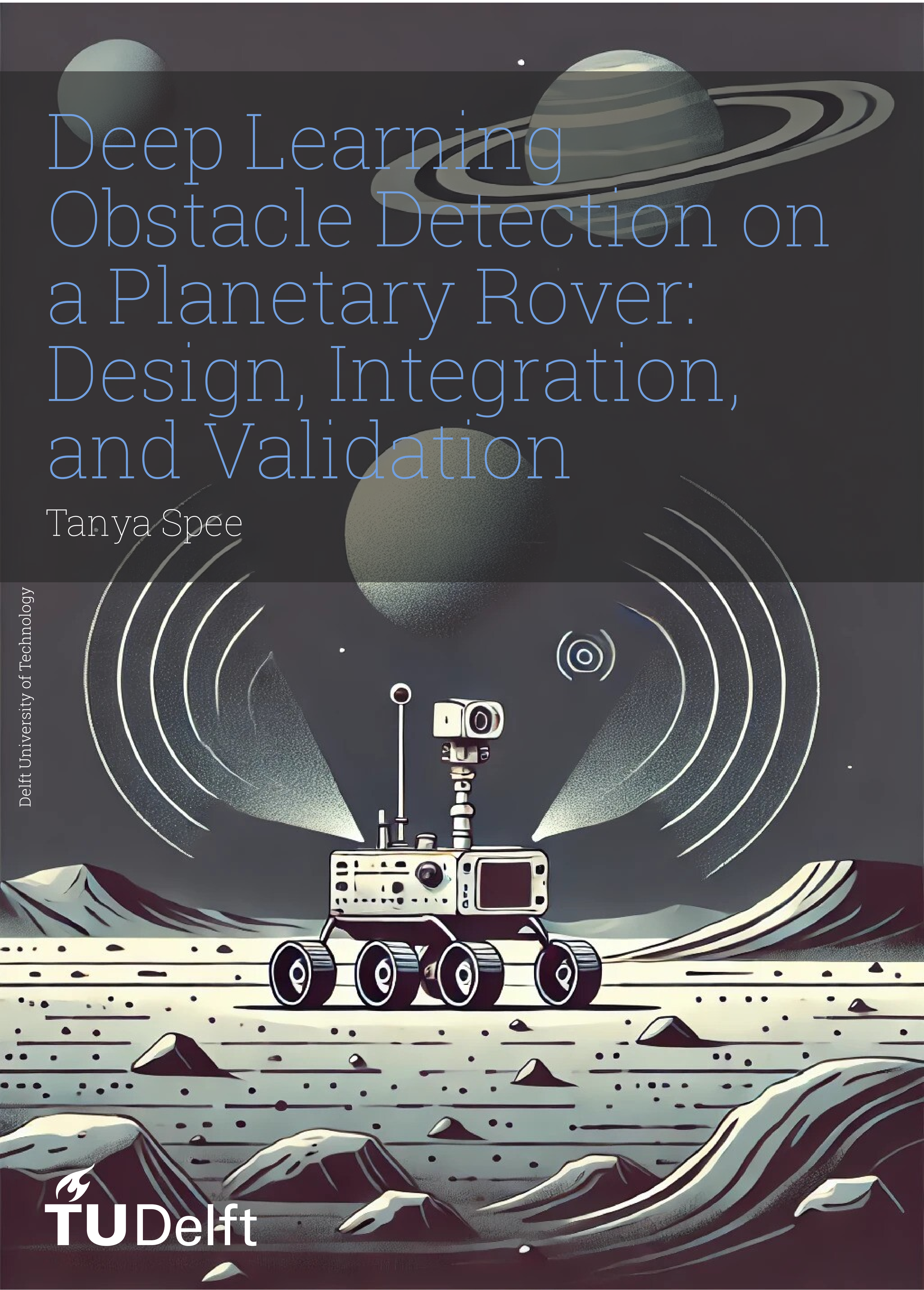
Deep Learning Obstacle Detection on a Planetary Rover: Design, Integration, and Validation

Tanya Spee

Delft University of Technology



Deep Learning Obstacle Detection on a Planetary Rover: Design, Integration, and Validation

by

Tanya Spee

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday, June 16, 2025, at 09:30 AM.

Student number:	4562739	
Project duration:	May, 2024 – June, 2025	
Thesis committee:	Dr. ir. R. Saathof,	TU Delft, Chair
	Prof. Dr. A. Menicucci,	TU Delft, Supervisor
	Prof Dr. J. Alonso-Mora,	TU Delft, Supervisor
	Ir. D. Rijlaarsdam,	Ubotica Technologies, Supervisor
	Dr. ir. C.J.M Verhoeven,	TU Delft, Examiner

Cover: OpenAI, personal communication

TU Delft

Preface

It feels incredible that with this thesis, which I am very proud of, I conclude my time in Delft by obtaining both a master's degree in Robotics and in Aerospace Engineering. I could not have asked for a better topic, better guidance, or more enthusiasm from the people involved.

I would like to express my gratitude to my supervisors. To David, who has always been very involved, ready for discussions on the topic, and who made me feel more than at home in Ubotica's team. To Alessandra, for her guidance and passion for the topic. I would also like to thank Lennart, Alex, and Martin from ESA's Automation & Robotics section for their help in the Planetary Robotics Lab, always ready to tackle any problem on the spot when the MaRTA rover was having breakdowns again.

Also, a thank you to all colleagues at both Ubotica and ESA for the fun times, good conversations, and valuable insights. Lastly, big thanks to my family, friends, and Dries for supporting me throughout both the thesis and my studies, and for making my time in Delft as fun as it was.

Here's to a new and exciting era of planetary exploration, and who knows you might see me in space in the future!

Tanya Spee
Delft, June 2025

Deep Learning Obstacle Detection on a Planetary Rover: Design, Integration, and Validation

Tanya Spee, David Rijlaarsdam, Alessandra Menicucci, Javier Alonso-Mora, Lennart Puck, Martin Azkarate

***Abstract*—Autonomous planetary rovers require obstacle detection capabilities to navigate hazardous terrain without Earth-based intervention, yet currently deployed methods are limited. This research develops and validates a complete deep learning system for autonomous rock detection on a resource-constrained planetary rover. We present a lightweight MobileNetV2-based U-Net architecture with dual attention mechanisms, optimized for edge deployment with only 0.31 million parameters. A new dataset MarsTanYard was created via a semi-automated dataset creation pipeline, enabling efficient annotation of Mars-analogue terrain imagery. Our deep learning network was trained on this dataset and integrated with a ROS2 navigation stack through a modular architecture that transforms segmentation masks into 2D occupancy grids that can be used for rover path planning. Our network achieves a 77% intersection-over-union accuracy for rock segmentation, and physical validation on a testing rover in a Mars analogue environment demonstrates a 94% detection rate for large rocks at close-range. An inference time of 4.49ms was achieved on the target rover hardware using model optimization techniques. The system maintains reliable operation across varying lighting conditions with less than 15% performance degradation. Results show theoretical collision probability of 7.8×10^{-7} per rock encounter, enabling months of autonomous operation for typical planetary missions. This work provides an end-to-end validation of the deep learning obstacle detection system, establishing a foundation for enhanced rover autonomy in future Mars exploration missions.**

I. Introduction

PLANETARY rovers serve as our robotic explorers on distant worlds, expanding our understanding of the solar system, and scouting regions where humans may set foot in the future. As missions become more ambitious, the need for greater autonomy in these rovers has become significant. This research focuses on a critical aspect of rover autonomy: obstacle detection, using deep learning as the core method, with an emphasis on end-to-end system integration and real-world testing.

Autonomous navigation is essential for maximising the scientific return of planetary rover missions, as continuous human intervention is impractical for exploring celestial bodies like Mars and the Moon [1]. Several factors drive this need for autonomy:

- Time delays in communication (10-45 minutes round trip for Mars) and limited windows of opportunity due to planetary rotations and relay orbiter availability cause slow operations and significant rover idle time [2].
- Scientific objectives can be achieved faster and more efficiently using autonomous navigation, as the rover's limited bandwidth can be spared for science-related data rather than image data needed for manual navigation [3].
- Autonomous hazard detection enhances rover safety, as it enables rovers to react in real-time to dangerous terrain or obstacles encountered along pre-defined paths that may not have been visible to human operators in earlier imagery [4].

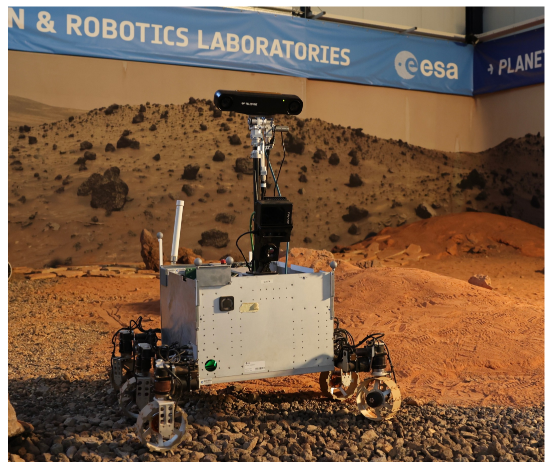


Fig. 1: Rover platform "MaRTA" in the European Space Agency's Planetary Robotics Laboratory's Mars Yard.

Currently deployed rover navigation systems like NASA's (National Aeronautics and Space Administration) Enhanced Navigation (ENav) on the Perseverance rover have already improved autonomous capabilities compared to previous missions. However, these systems still rely on geometric approaches for obstacle detection that have critical limitations [5] [6], as outlined in Section II.

Deep learning (DL) approaches offer promising solutions to these challenges. Recent advances in semantic segmentation and terrain classification using convolutional neural networks (CNNs) have demonstrated superior performance in identifying various terrain types and obstacles [7]. These approaches can improve hazard detection accuracy, distinguishing between visually similar but mechanically different textures [8] [9], reduce computation time through optimised architectures [10], and enhance robustness and adaptability to unseen terrain [11].

Despite the potential of deep learning for planetary obstacle detection, research gaps remain in adapting these systems for actual rover deployment with computational limitations, system integration, and real-world validation. This research therefore aims to answer the following main question: "How can deep learning models for obstacle detection be optimised, deployed, and integrated on a planetary rover to enable low-

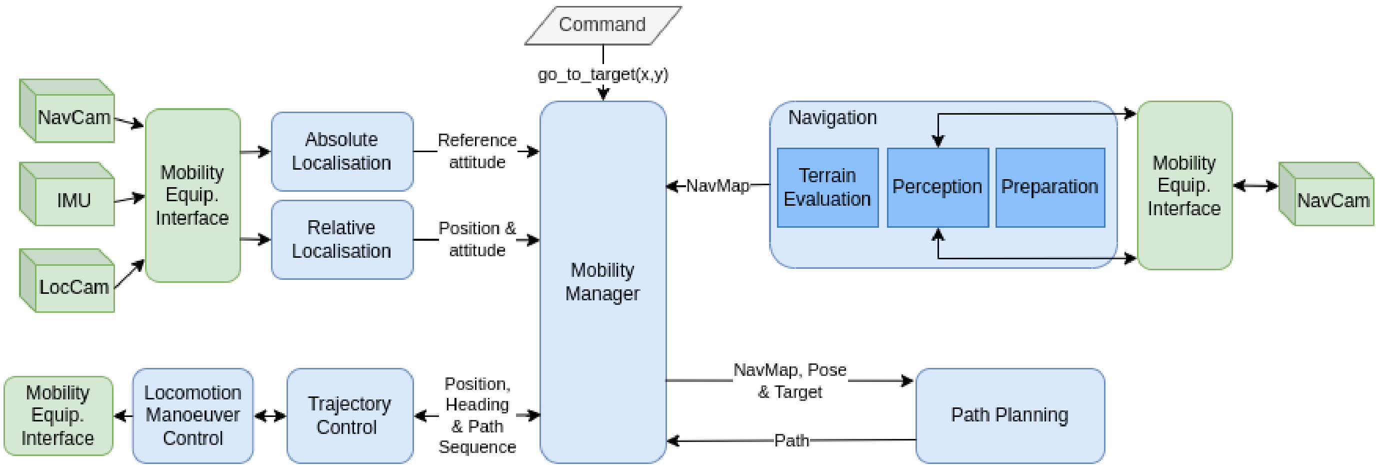


Fig. 2: Simplified ExoMars Rover Mobility Functional Architecture, as shown in [12].

latency, on-board inference?" This research question is supported by three sub-questions:

1) What DL architectures and optimisations enable accurate and low-latency obstacle detection on a resource-constrained planetary rover?
2) How can the deep learning obstacle detection system best be integrated with other rover software subsystems and be deployed on the rover hardware?
3) How does the end-to-end system using the integrated DL obstacle detection method perform on a planetary rover test platform in a realistic environment?

Answering these questions, the contributions of this study are: (1) design and optimization of a lightweight deep learning model for obstacle detection, (2) creation of a Mars analogue dataset for model training and evaluation, (3) integration of the DL system within a rover navigation stack, (4) hardware deployment and hardware-in-the-loop evaluation, (5) validation on a rover test platform in a Mars analogue environment (Figure 1), and (6) recommendations for future work.

II. Related Work

This section summarizes key developments in autonomous planetary rover navigation systems, obstacle detection algorithms, deep learning approaches, and deployed deep learning systems in space.

A. Current Rover Obstacle Avoidance Systems

Recent planetary rovers employ Guidance, Navigation, and Control (GNC) systems to autonomously navigate planetary surfaces between the waypoints set by human operators. Such GNC systems typically consist of four interconnected subsystems: navigation, path planning, localisation, and trajectory control [13] [14]. The European Space Agency's (ESA) ExoMars rover (to be launched in 2028) GNC functional architecture represents the current state-of-the-art in autonomous rover system design [12], shown in Figure 2. The navigation subsystem detects obstacles and assesses terrain traversability. Path planning then uses this assessment to generate safe routes, while localisation tracks the rover's position using techniques such as Visual Odometry (VO), and trajectory control translates the planned paths into motor commands. This review focuses on subsystem perception within the navigation pipeline, and how it interfaces with the other subsystems to enable autonomous navigation [15].

The rover's subsystems are managed by the 'Mobility Manager', which enables the structuring and hierarchizing of the different subsystems, and decides on the transitioning between modes when its requirements are satisfied. For example, traverse mode can only begin when actuators are warmed up and a safe path is available [14]. Currently deployed rovers operate primarily in supervised autonomy mode, where human operators make decisions about functional mode transitions and route selection. [13].

Within the navigation pipeline, nearly all recent planetary rovers perform terrain geometry analysis to determine if areas are safe to drive through. This approach is used by NASA's rovers (Spirit, Opportunity, Curiosity, and Perseverance), as well as other missions such as China's Zhurong rover. This geometric method follows a clear pipeline: Stereo cameras capture pairs of images from slightly different positions. Stereo matching algorithms then calculate the disparity between corresponding pixels in each image pair by matching corresponding features between the left and right images. This disparity data is then converted into a 3D point cloud via triangulation mathematics and processed into Digital Elevation Models (DEMs). These DEMs are height maps that show the 2.5D shape of the terrain in front of the rover [16]. The rover then applies straightforward geometric thresholds to these DEMs as a rule-based safety analysis. Areas with slopes steeper than 30° are typically marked as unsafe, as are regions with height differences larger than the rover's ground clearance [17]. This approach is often able to detect obvious hazards like large rocks, steep slopes, or deep holes.

NASA's "GESTALT" algorithm, used on Spirit, Opportunity, and Curiosity, is an example of this geometric approach [17]. For each stereo image pair, GESTALT processes an individual DEM and creates a local map marking areas as safe or unsafe. The algorithm treats the rover as a simple disk shape and expands all detected hazards by the rover's radius. This ensures safety but means the rover cannot drive over or between obstacles, even when it might physically be able to do so. This conservative design was necessary due to hardware limitations. Early rovers used slow processors (20MHz RAD6000 for Spirit and Opportunity, 133MHz RAD750 for Curiosity), making it impossible to combine the computationally expensive DEMs from multiple images [17] while performing accurate localisation using Visual Odometry. Instead, GESTALT only tracked the simple safe/unsafe decisions, which required less computation but lost detailed information about the terrain.

Perseverance's Enhanced Navigation (ENav) system represents a computational upgrade while keeping the same geometric principles [18]. The key hardware improvement is the Vision Compute Element (VCE), a dedicated image processing computer that includes a RAD750 processor paired with Field Programmable Gate Array (FPGA) acceleration. This hardware partly enables 'thinking while driving', unlike previous rovers that had to stop moving while processing images [16]. Furthermore, unlike GESTALT's simple disk model, ENav's Approximate Clearance Evaluation (ACE) algorithm analyzes where each individual wheel would be positioned and calculates safety margins for the rover's belly clearance. This allows the rover to safely drive over small rocks or through narrow passages that GESTALT would avoid, enabling navigation through more complex terrain.

However, these geometric methods have fundamental limitations beyond computational constraints. Stereo vision fails in poor lighting conditions or regions with poor feature density, resulting in elevation maps with gaps or inaccurate data [5]. Furthermore, geometric analysis cannot distinguish between terrain types that appear geometrically similar but behave very differently. For example, a sand slope and a rock slope of a similar angle may pose entirely different risks, as sand can trap the rover [19]. Geometric methods miss the semantic understanding of the environment to understand material properties, while human operators naturally consider terrain properties when planning routes. Even with faster processors and better cameras, the current terrain analysis approach will still rely on human operators to identify non-geometric hazards and specify areas to avoid [13].

Beyond NASA's missions, other space agencies have deployed rovers with similar autonomous navigation capabilities. Within ESA's ExoMars perception subsystem, a two-level approach is used with both "efficient navigation" for simple terrain and "full navigation" with SLAM (Simultaneous Localisation and Mapping) capabilities for complex environments. The efficient navigation acts like an "electronic bumper" analyzing only the region directly in front of the rover. It avoids performing computationally expensive terrain reconstructions, but relies directly on disparity values and precomputed lookup tables to determine if the terrain is safe [14]. Although much faster, it uses the same geometric approach as the earlier discussed systems and faces the same limitations. China's Zhurong rover (Mars, Tianwen-1 mission, 2021), Yutu rovers (Moon, Chang'e 3 and 4 missions, 2013 and 2019), and India's Pragyan rover (Moon, Chandrayaan-3 mission, 2023) all employ similar geometric approaches [20] [21] [22] [23]. They also perform disparity calculations using stereo vision, and either DEM generation or rule-based obstacle filtering, though detailed algorithmic specifications remain limited in publicly available literature. The convergence on stereo vision-based geometric analysis across different space agencies demonstrates both the maturity of this approach and the lack of viable alternatives for current planetary rover operations.

B. Alternative Approaches for Obstacle Detection

The limitations of geometric methods have motivated research into alternative approaches for planetary obstacle detection. Classical machine learning methods extract handcrafted features such as slope, roughness, texture, and color from terrain data, then train classifiers like Support Vector Machines or Random Forests to distinguish safe from hazardous areas. While more capable than pure geometric thresholds, these approaches rely on human experts to extensively define relevant features and struggle with novel terrain types.

Sensor fusion approaches combine geometric analysis with limited semantic understanding, using visual texture analysis to distinguish rock types or incorporating extra sensors to assess surface properties. Thermal imaging, Light Detection and Ranging (LiDAR) and spectrometers can be integrated to build richer terrain representations. These approaches face challenges in sensor synchronization and calibration, while many sensors are currently unsuitable for the space environment.

Deep learning (DL) represents a fundamentally different paradigm that addresses the core limitations of previous methods. Rather than applying predetermined rules to geometric features, deep neural networks learn complex mappings between raw sensor data and terrain properties directly from training examples. This data-driven approach can enable the rover to understand material properties from visual appearance, adapt to previously unseen environments through learned representations, and reduce computation time [19]. Where geometric methods use explicit mathematical models with deterministic outputs, deep learning uses implicit feature representations learned through gradient-based optimization of millions of parameters, trading interpretability for adaptability. While geometric approaches have predictable computational requirements and bounded behavior, deep learning offers greater flexibility to capture complex terrain relationships, but at a cost of transparency in the decision-making processes.

C. DL architectures and Datasets for Planetary Obstacle Detection

Semantic segmentation has become the predominant deep learning approach for planetary obstacle detection, with recent architectures making different design choices to balance accuracy, computational efficiency, and deployment constraints (shown in Table I). Since these methods are evaluated on

Architecture	Key Features	Main Strength	Primary Limitation
MobileNetV2 U-Net [10]	MobileNetV2-based encoder	Massive parameter reduction	Reduced accuracy and precision
RSU-Net [24]	SENet channel attention on U-Net backbone	High benchmark performance	Large parameter count
DAM-U-Net [25]	Channel and spatial attention mechanisms	Comprehensive attention coverage	Implementation complexity
Y-shaped Network [26]	One encoder, segmentation and depth decoders	Multi-task scene understanding	Task interference and loss balancing
RockFormer [27]	Full transformer U-shaped architecture	Global context through self-attention	Transformer computational overhead
RockSeg [28]	ResNet blocks combined with transformers	Local and global feature fusion	Hybrid architecture complexity
YOLOv5 (modified) [9]	YOLO detection with attention mechanisms	Fast inference for detection	Bounding boxes, no pixel-level masks

TABLE I: Summary of recent proposed deep learning architectures for planetary obstacle detection

different datasets with varying annotation quality, image complexity, and planetary environments, direct quantitative comparisons are challenging, and this analysis therefore focuses on the qualitative trade-offs that distinguish these approaches.

Standard U-Net architectures achieve excellent segmentation through their encoder-decoder structure with skip connections but require too many parameters for rover deployment. MobileNetV2 U-Net [10] tackles this problem by replacing the standard encoder with MobileNetV2's depthwise separable convolutions, significantly reducing the parameter count. This enables deployment on systems like Raspberry Pi while maintaining acceptable segmentation quality.

Attention mechanisms in recent works follow two distinct strategies. RSU-Net [24] uses channel attention to emphasize informative feature channels, which works well for distinguishing rocks from similar-colored terrain. DAM-U-Net [25] instead combines both channel and spatial attention for more comprehensive feature enhancement.

Multi-task networks attempt to extract additional scene information beyond segmentation. The Y-shaped Network [26] combines rock segmentation with depth estimation using a depth-aware spatial attention module. The mode contains one shared encoder, and two distinct decoders for the two tasks. The depth decoder branch provides 3D spatial information that improves boundary detection and small rock identification. However, training multiple tasks simultaneously introduces complexity in loss balancing and potential task interference.

Hybrid CNN-transformer architectures address the limited receptive fields of pure CNN approaches. RockSeg [28] combines ResNet blocks with transformer modules to capture both local features and global context while maintaining efficiency. RockFormer [27] uses a full transformer-based U-shaped architecture that prioritizes global context through self-attention. RockSeg suits real-time applications better with its lighter network, while RockFormer excels at detecting small, scattered rocks across complex terrain but is heavier.

YOLOv5 with attention mechanisms [9] provides fast obstacle detection suitable for immediate navigation decisions, only outputting bounding boxes around rocks and craters. Segmentation methods like the U-Net variants provide pixel-level rock boundaries essential for detailed path planning around irregular obstacles. The choice between segmentation and detection using bounding boxes depends on whether rapid detection or precise spatial understanding is more critical for the navigation system.

All DL planetary obstacle detection methods described conduct performance evaluation exclusively on pre-collected datasets and in simulations without addressing software and hardware integration challenges or system-level validation.

Deep learning models for planetary terrain segmentation rely on three types of datasets: real (e.g., AI4MARS [19], MarsRock [24]), synthetic (e.g., SimMars6K [29], SynMars [30]), and analogue (e.g., MADMAX [31], Katwijk Beach [32]). Real datasets provide authentic planetary imagery but may be limited in diversity or annotation quality, and lack precise ground truth information. Synthetic datasets are computer-generated datasets that offer perfect ground truth information, but do not fully capture the real-world complexity and textures, introducing a domain gap. Analogue datasets consist of images taken on Earth that look like planetary surfaces. They provide a balance of real-world complexity and controlled environments with ground-truth information, but do not capture both as perfectly.

D. DL Deployment in Space

Deploying deep learning models on planetary rovers requires hardware that balances computational capabilities with the constraints of space environments. Traditional space-grade processors like the RAD750 used in Curiosity and Perseverance offer limited performance for AI workloads due to minimal parallel processing capabilities [16]. Specialized AI accelerators, such as radiation-tolerant FPGAs and Vision Processing Units (VPUs), provide more promising alternatives for neural network inference while meeting space-grade requirements [33] [34]. Key constraints for space hardware are:

- Radiation tolerance of the electronics
- Operation across extreme temperature ranges
- Strict power constraints
- Minimal mass and volume

Deep learning deployment in space remains limited, with only a handful of systems operating on small satellites and space station robots in low-earth orbit. ESA's PhiSat-1 mission (2020) pioneered onboard deep learning by using CNNs on the Intel Myriad 2 for cloud filtering, reducing data transmission [35]. Its follow-on PhiSat-2 extends this to a library of in-orbit ”apps” for cloud-masking, vessel detection, and image processing tasks [36]. Multiple CubeSat missions have demonstrated lightweight CNNs on platforms like Intel Myriad-2 and NVIDIA Jetson for image classification [37] [38]. These recent examples show that deep learning systems, specifically with image data, can be successfully deployed in space environments.

Concerning planetary applications, the company Mission Control developed MoonNet, a deep learning model intended for lunar surface segmentation on the ispace M1 mission in collaboration with the Emirates Lunar Mission Rashid rover [39]. Although the spacecraft did not successfully land, MoonNet was confirmed to be operating nominally in lunar orbit, making it the world's first and only deployed deep learning system outside Earth orbit. It has not been specified what hardware the deep learning system was deployed on. Recent research has been done on the validation of Ubotica Technologies' CogniSat XE2 board using the Myriad X for deep learning planetary obstacle detection. It was demonstrated to be successful in segmenting obstacles in a pre-collected and annotated dataset [40].

E. Research Gaps

Despite advances in both geometric obstacle detection for rovers and in deep learning approaches, three key research gaps were identified: (1) computational optimization of deep learning planetary obstacle detection models for resource-constrained rover hardware, (2) end-to-end system integration with existing rover navigation stacks, and (3) real-world validation on physical platforms in realistic environments rather than simulation-only testing. This research addresses these gaps through the research questions and contributions outlined in the introduction.

III. System Engineering & Architecture Design

This chapter presents the system engineering approach and architectural design for developing a deep learning-based obstacle detection system for planetary rovers.

A. V-Model Development Framework

For the design of the deep learning-based obstacle detection system, the V-model systems engineering approach was deployed as shown in Figure 3, based on literature [41]. This framework provides an iterative and systematic method for decomposing high-level mission requirements into detailed subsystem specifications, while verifying the designed subsystems and integrated systems. The left side of the V represents the decomposition path from mission needs to subsystem requirements, while the right side focuses on subsystem design and integration into bigger systems. Verification is done to check whether the (sub)systems meet their specifications, and validation is done to check whether the system meets the mission needs. The V-model aligns with the Autonomy Requirements Engineering (ARE) methodology, a framework developed by ESA for space missions to address the challenges of autonomous systems [42]. ARE emphasizes the importance of defining verifiable autonomy properties at different system levels, supporting the choice of the V-model approach. In this research, the focus is on ensuring that the perception-to-navigation chain can be verified at each subsystem level and that the system is validated in real-world conditions, as simulations alone cannot fully mimic the interactions between hardware, perception systems, and interfaces with the other software components [43].

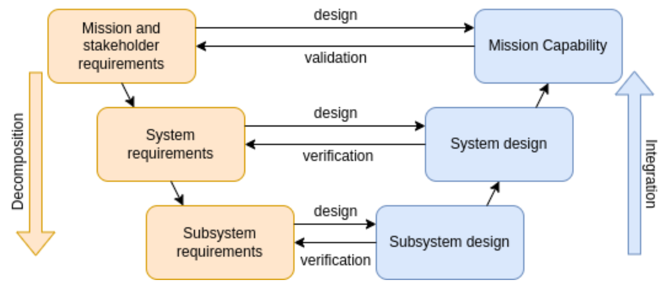


Fig. 3: V-model development framework, based on [41]

B. Concept of Operations and System Requirements Derivation

For the design of the deep learning obstacle detection system and its subsystems, requirements are derived following a systematic approach shown in Figure 4, based on space mission design methodologies [42]. The mission statement and stakeholder analysis are taken as starting point. The mission statement is to develop a low-latency, on-board obstacle detection system for planetary rovers that enables safe, autonomous navigation over planetary surfaces. Three stakeholders are identified:

1) Ubotica Technologies: A company specializing in commercial off-the-shelf (COTS) AI accelerators for space applications. Their interest lies in advancing the understanding and feasibility of running deep learning inference directly on planetary rovers.
2) European Space Agency (ESA): ESA aims to enhance rover autonomy to reduce dependence on Earth-based control, particularly for Mars missions where communication delays are critical. They support this research by sharing expertise and providing access to the Mars Yard, an analogue Martian environment.
3) Lunar Zebro (TU Delft): This TU Delft initiative is developing sub-1.5 kg micro-rovers designed to operate autonomously in swarms. The research specifically targets lightweight obstacle detection models that can run on such ultra-constrained platforms, supporting the scalability and autonomy of the Zebro swarm concept.

These stakeholders share a common goal of achieving reliable autonomous obstacle detection in planetary environments with minimal computational resources, specifically using deep learning. This convergence of interests shaped the requirements for a deep learning perception system that is robust, lightweight, and modular to adapt across different rover platforms.

The Concept of Operations (CONOPS) defines how the deep learning obstacle detection system integrates operationally with rover missions to enable safe, autonomous navigation across planetary surfaces. The CONOPS for this research can be summarised as follows:

1) Ground operators define science objectives and general waypoints based on orbital imagery
2) The rover executes the planned traverses while continuously monitoring terrain ahead and detecting potential

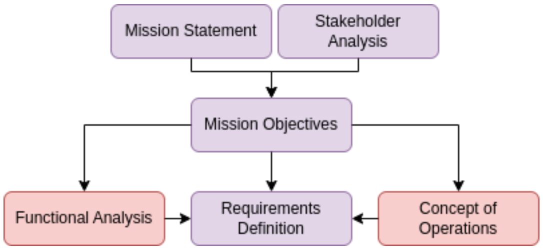


Fig. 4: Requirements derivation methodology, based on space mission design literature [42]

hazards.

3) When obstacles are detected that interfere with the planned path, the system triggers local replanning and continues operations without requiring ground intervention. Only rock obstacles are considered within the scope of this research.
4) During communication windows, the rover transmits relevant data to ground operators such as obstacle maps, performance telemetry, and important imagery for mission assessment
5) If critical obstacles block all viable paths or system anomalies occur, the rover safely halts and flags the situation for manual control by ground operators.

The operational concept defines clear roles for both autonomous rover operation and human control. Human operators provide high-level mission objectives, review system performance, and can override rover decisions, while the rover autonomously makes real-time local navigation decisions and traverses the surface while maintaining safety.

The complete list of mission objectives and subsequent system requirements is provided in Appendix A. The requirements were formulated according to SMART principles (Specific, Measurable, Achievable, Relevant, Time-bound) as far as possible and relevant, and categorised as functional, performance, design, interface, and validation requirements. The requirements were established within the constraints of the available testing platform and environment, as detailed in Subsection D. Each requirement was assigned a priority level and a verification method following ECSS standards (test, analysis, inspection, or review) [44]. The functional requirements define what the system must do, the performance requirements specify how well it must perform, the interface requirements describe how it connects with other systems, the design requirements establish the architectural and implementation constraints, and the validation requirements define how the system's performance will be evaluated.

These mission objectives and system requirements are validated and verified throughout this research paper. The complete verification overview is summarized in Appendix refapp:verification and will be referred to during the respective verification activities.

C. Testing Platform

The system is designed primarily for lightweight rover platforms and was validated on ESA's MaRTA (Martian Rover Testbed for Autonomy) rover in the Mars Yard at ESA's Planetary Robotics Laboratory (PRL) [45], both shown in Figure 5. MaRTA provides an ideal testing platform, larger and with more computational power than nano-rovers, but still representative of the target application with its modular design.

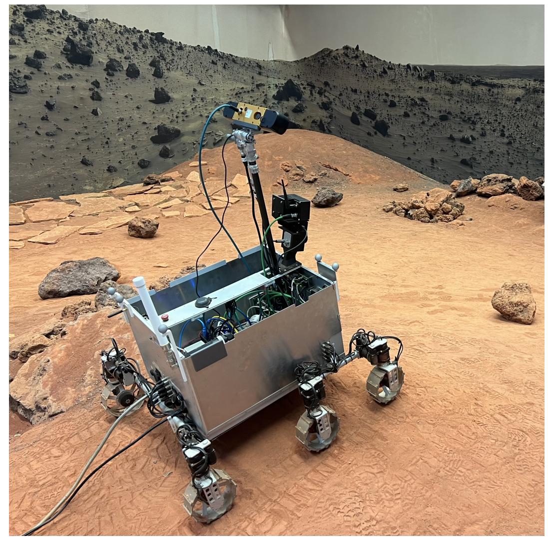


Fig. 5: Test rover "MaRTA" in the Mars Yard test environment at ESA's Planetary Robotics Laboratory.

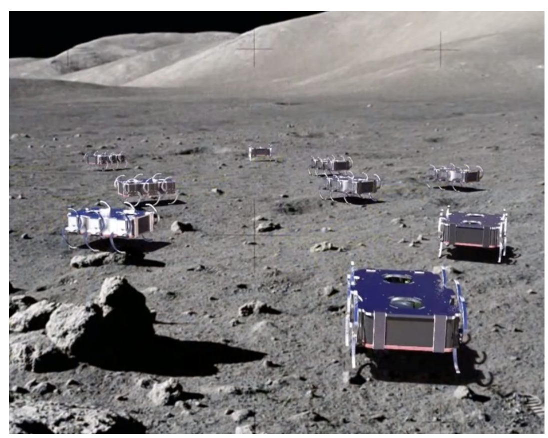


Fig. 6: Conceptual illustration of the Lunar Zebro swarm [46]

MaRTA is a 32 kg testing rover platform, developed as a scaled-down version of the ExoMars rover (Rosalind Franklin). It features a Teledyne Bumblebee X stereo camera mounted on a pan-tilt unit, providing a 60° horizontal field of view. The rover uses EtherCAT (100 Mbit/s) **??** for internal communication between all motors and houses an NVIDIA Orin AGX for onboard computing, making it well-suited for deep learning inference. Its triple-bogie suspension with 6-wheel steering enables navigation over Mars-like terrain at speeds up to 10 cm/s and can traverse rocks up to its wheel diameter of 15 cm, even with vertical surfaces.

The Mars Yard provides a realistic analogue environment with varying terrain types, including sandy areas and rocky sections with obstacles of different sizes. The controlled

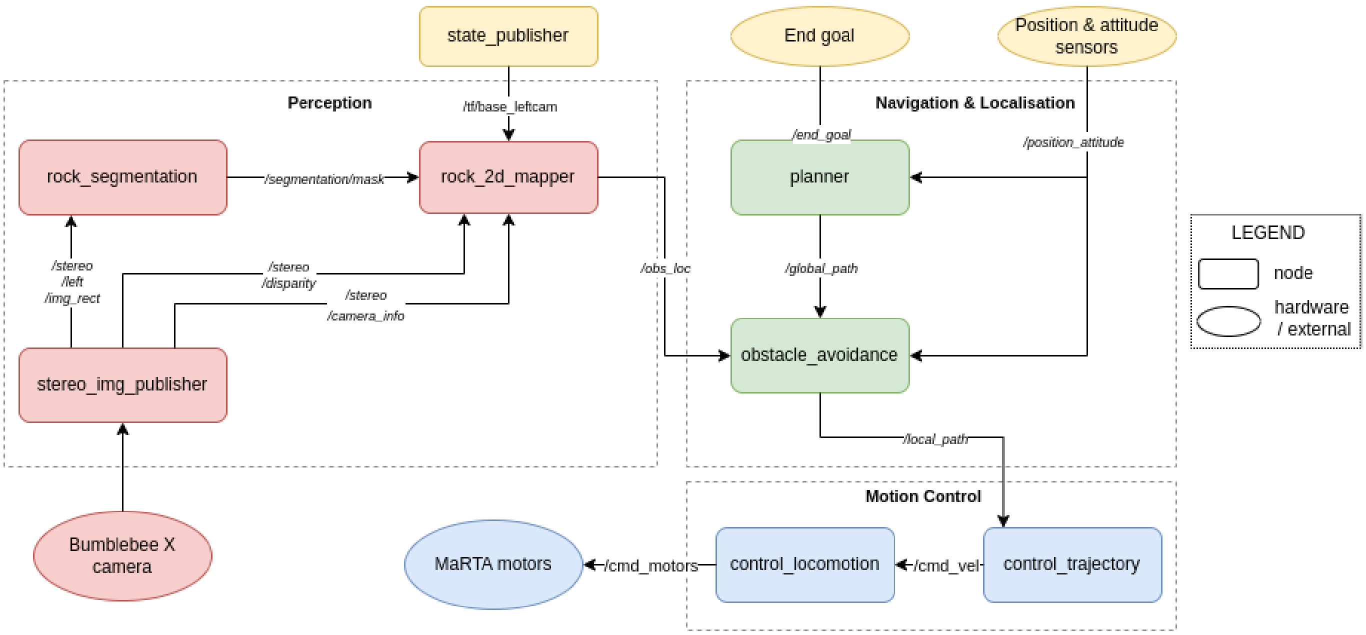


Fig. 7: ROS2 functional node graph showing the perception pipeline components and their interactions within the overall rover system architecture.

environment allows for repeatable experiments, while variable lighting conditions enable testing the robustness of the perception system under different illumination scenarios.

For future lightweight applications, the Lunar Zebro platform (Figure 6) represents the type of resource-constrained rover that could benefit from this technology. While current versions of these small rovers have limited onboard computing capabilities, future iterations with more efficient hardware could leverage DL algorithms similar to the one developed in this work, as was already tested in earlier research [40].

D. System Architecture Design

The obstacle detection system encompasses both hardware and software components, designed to integrate with existing rover platforms while meeting the performance requirements specified in Appendix A. The system boundaries are the stereo camera sensors, onboard computing hardware, and the complete software stack for perception and obstacle mapping.

The primary constraint is that the system must operate within MaRTA's existing hardware configuration, meeting requirement IR-04, while ensuring software portability to other rover platforms. MaRTA's Teledyne Bumblebee X stereo camera and NVIDIA Orin AGX computing platform provide the sensor input and processing capabilities required, with EtherCAT communication enabling the data flow necessary for low-latency operation.

The software architecture is built on Robot Operating System 2 (ROS2), a middleware framework providing standardized communication infrastructure for robotic applications, thereby satisfying requirement FR-03. The modular design enables component-wise development and facilitates transfer to different rover platforms without major architectural changes. The high-level ROS2 functional node graph for this research is shown in Figure 7, highlighting the interactions between perception, navigation, and locomotion control pipelines. Within the perception subsystem, the rock segmentation and rock mapper nodes represent the core contributions of this research.

The system architecture directly supports the CONOPS through its sense-plan-act implementation. The stereo camera system continuously captures imagery at 1Hz to meet requirement PR-02, providing terrain coverage ahead of the rover. Raw stereo imagery undergoes the preprocessing, obstacle segmentation, and obstacle mapping to generate the local occupancy grids that integrate with global path planning algorithms as shown in Figure 7. The modular architecture ensures that individual components can be validated independently while supporting end-to-end system validation on the MaRTA platform. More details on the software implementation and integration testing will be provided in Section V.

IV. Deep Learning Model Design

This section details the development of a lightweight deep learning model for planetary rock segmentation, covering dataset selection and creation, neural network architecture design, training methodology, and performance evaluation.

A. Dataset Selection and Creation

Obstacle segmentation on planetary surfaces presents unique challenges due to monochromatic environments and similar textures. The dataset selection to train the deep learning model on, requires consideration of environmental representation, annotation quality, size and diversity and accessibility. Since the scope of this research is to perform rock segmentation, only this annotation is needed. No information is needed for other terrain properties such as bedrock, sand, and craters.

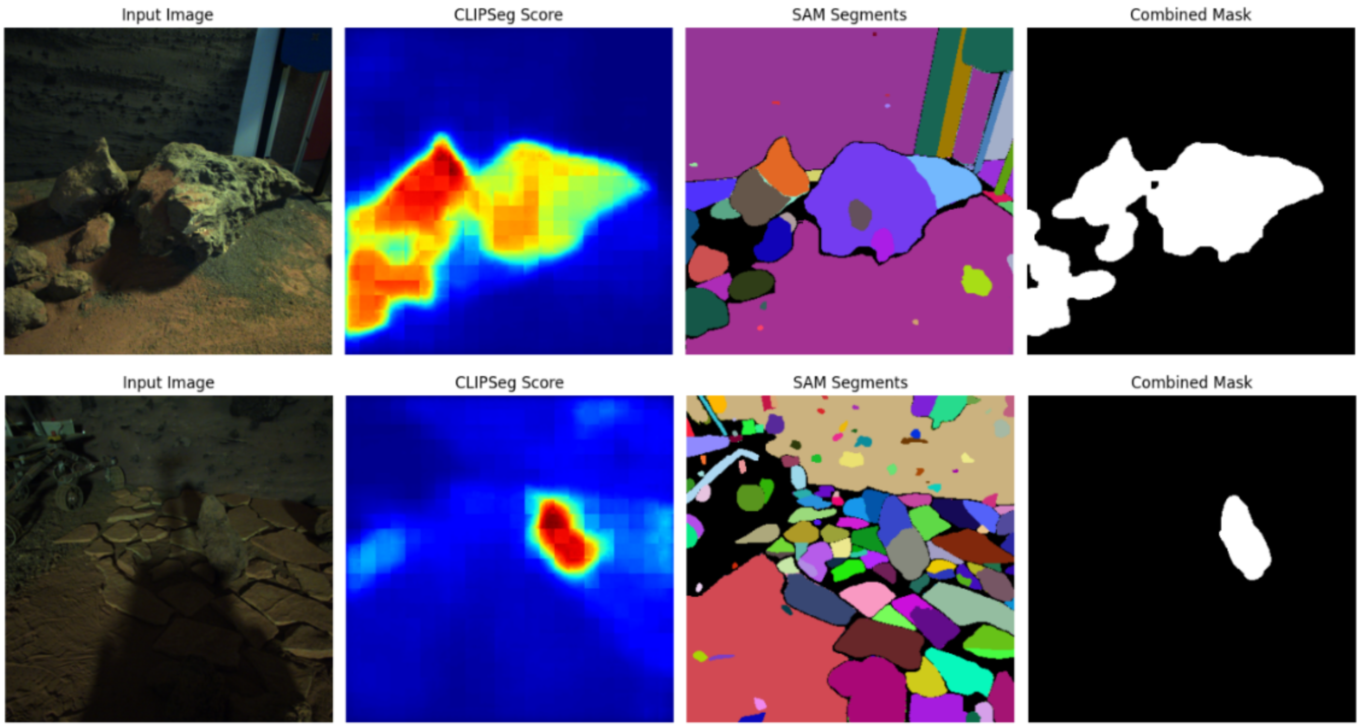


Fig. 8: Example images during MarsTanYard dataset creation, showing the integration of CLIPSeg and SAM models for semi-automated segmentation mask generation.

TABLE II: Comparison of shortlisted datasets for planetary rock segmentation

Dataset	**Type**	**Size**	**Colour**
AI4MARS [19]	Real	326,000	Grayscale
MarsData-V2 [47]	Real	8,000	RGB
MarsRock [24]	Real	1,194	Grayscale
SimMars6K [29]	Synthetic	6,000	RGB
SynMars [30]	Synthetic	60,000	RGB
Artificial Lunar [48]	Synthetic	9,766	RGB

The available real, analogue, and synthetic Mars datasets were researched and evaluated. The comparison table of shortlisted datasets can be found in Table II. The real-world dataset, MarsData-V2, was chosen due to its real RGB image content, its annotation quality, binary class structure, and dataset size.

While the MarsData-V2 dataset provided valuable resources, an additional mission-specific dataset was highly desirable as there was still a significant domain gap between the existing datasets and the deployment conditions. The mission-specific dataset, named "MarsTanYard", contains images from MaRTA's left camera in the analogue Mars environment that would eventually be used for testing. This approach ensures that the model is optimised for its deployment. Traditional dataset creation for rock segmentation typically requires extensive manual labour, drawing all rock contours by hand. Because of time and resource constraints, as only one human was performing this task, an efficient semi-automated approach was developed using two state-of-the-art deep-learning foundation models:

- CLIPSeg [49]: A language-vision model that identifies regions matching text descriptions. It was trained on image-text pairs and it provides zero-shot capabilities for recognising concepts without specific training on them.
- Segment Anything Model 2 (SAM2) [50]: This instance segmentation model provides precise object boundaries, though it lacks semantic understanding of what the objects are.

Images were collected using MaRTA's left camera as it was driven over the Mars Yard with a gamepad controller, with the setup shown in Figure 9. The Mars Yard is illuminated by a single light source referred to as the 'sun' as seen in the top right of the figure, positioned at a 15-degree angle relative to the horizontal surface, measured from the Mars Yard centre point. To capture a wide range of shadow directions, MaRTA was positioned in various orientations and locations across the Mars Yard, resulting in shadows being cast in different directions. Terrain variability was further enhanced by combining different terrain types and rock formations within individual scenes, partially occluding each other. The different terrain types included in the dataset are: flat rocks (seen on the left part of the image), pebbles (seen on the bottom part), small and big rock boulders, and sand (throughout the image).

Two examples during the rock labelling workflow can be seen in Figure 10, and the system diagram of the labelling pipeline is shown in Figure 10. On the input image, the CLIPSeg identified rock regions using the prompt 'rock taller than 15cm', after experimenting with various prompts like "hazard," "boulder," and "stone". CLIPSeg provides a

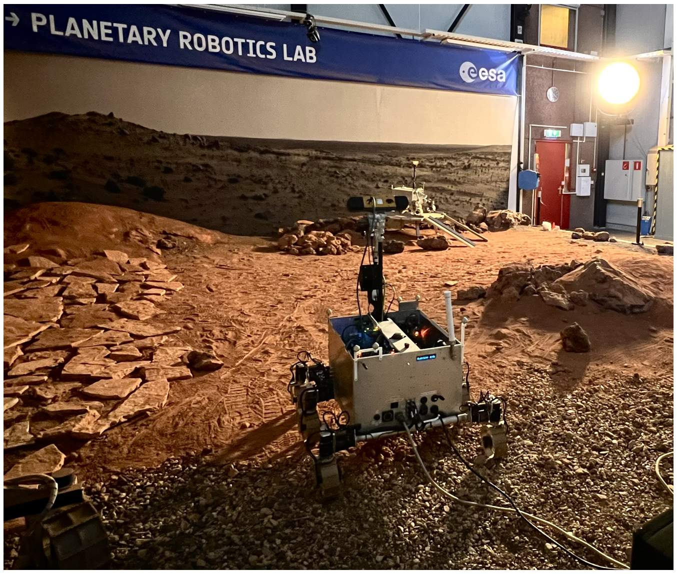


Fig. 9: Setup used for the MarsTanYard dataset generation, including a single light source as the 'sun', and different terrain types: flat rocks, pebbles, small and big rock boulders, and sand.

heatmap-like output with confidence scores. The bottom-row example in Figure 10 demonstrates CLIPSeg's ability to distinguish the single upright rock from the flat rocks beneath it in shadowy conditions, while the top example shows it distinguishes rocks from sand. SAM provided precise rock boundaries on the same input image. The results of the models were then combined, keeping SAM objects that overlapped with CLIPSeg rock predictions above a certain threshold. For each image, masks were generated at four different CLIPSeg threshold values (0.2, 0.3, 0.4, 0.5), and the resulting masks were analysed, with the best mask selected manually by expert opinion and discarding the others. This automated approach worked satisfactorily, as measured by the same expert opinion, for 78% of the images. For the remaining 22% of the images, a custom tool was built that did not use CLIPSeg and allowed manual selection of SAM segments with simple clicks, saving the combined clicked segments as a binary mask. The resulting dataset, "MarsTanYard," contains 364 images captured during the test and their corresponding binary masks.

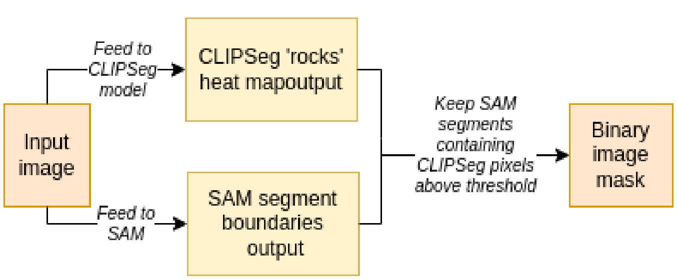


Fig. 10: System diagram of labelling pipeline for MarsTanYard's binary rock masks. Threshold manually set by expert opinion to 0.2, 0.3, 0.4 or 0.5.

Before feeding images to the deep learning model, several preprocessing steps were implemented to enhance model performance, efficiency, and robustness, and to ensure consistency between the different datasets. These steps are: centre square cropping, resizing to 256x256 (meeting requirement IR-02), colour conversion to RGB, normalization of pixel values to [0,1] floating-point range by dividing by 255, and data augmentation. The augmentation strategy was designed to represent realistic scenarios. For instance, horizontal flips were included as they represent plausible terrain orientations, while vertical flips were excluded as they would create unrealistic scenarios of rocks hanging from the sky. The data augmentations shown in Table III were applied, each with a 20% probability of being applied per image, and with values ranging between realistic boundaries.

TABLE III: Data Augmentation Parameters

Augmentation Type	**Value Range**	**Rationale**
Horizontal Flip	–	Different viewing angles
Brightness	0.8–1.2	Varying lighting conditions
Contrast	0.8–1.2	Enhanced feature visibility
Rotation	$\pm 15°$	Camera orientation variations
Noise Addition	$\sigma = 0$–0.03	Sensor noise simulation
Random Crop	80–100 %	Partial view learning
Gamma	0.8–1.2	Exposure variations

B. Neural Network Architecture

Selecting the appropriate neural network architecture for planetary terrain rock segmentation required consideration of both performance requirements and computational resource constraints. The architecture development and training optimization were conducted on a HP Victus laptop equipped with an Intel i5-11400H processor, 16 GB random access memory, and an 'NVIDIA GeForce RTX 3050' praphics processing unit (GPU).

Based on the literature review, several state-of-the-art architectures for planetary rock segmentation mentioned in the "Related Works" section were evaluated. As the goal of this research is to develop a very lightweight resource-efficient rock segmentation system that can be used on small rovers, a U-net with pre-trained MobileNetV2 encoder (ImageNet weights) and lightweight custom decoder was used as starting point, as the usage of MobileNetV2's depthwise separable convolutions greatly reduces the parameter count without significantly sacrificing performance. The classification is binary, 'hazard rock' or 'background', to comply with requirement FR-04. Skip connections were implemented by extracting intermediate feature maps from MobileNetV2 blocks at specified depths (blocks 1, 3, 6, and 13) chosen to provide features at different scales, preserving the pre-trained weights. From there, the influence of different architectural modifications was analyzed via an ablation study to optimize the model by balancing segmentation performance with computational constraints. Four targeted ablations were conducted on the training split of the MarsData-V2 dataset [47], with each configuration tested across 5 independent runs. The configurations are shown in Table IV and are outlined below:

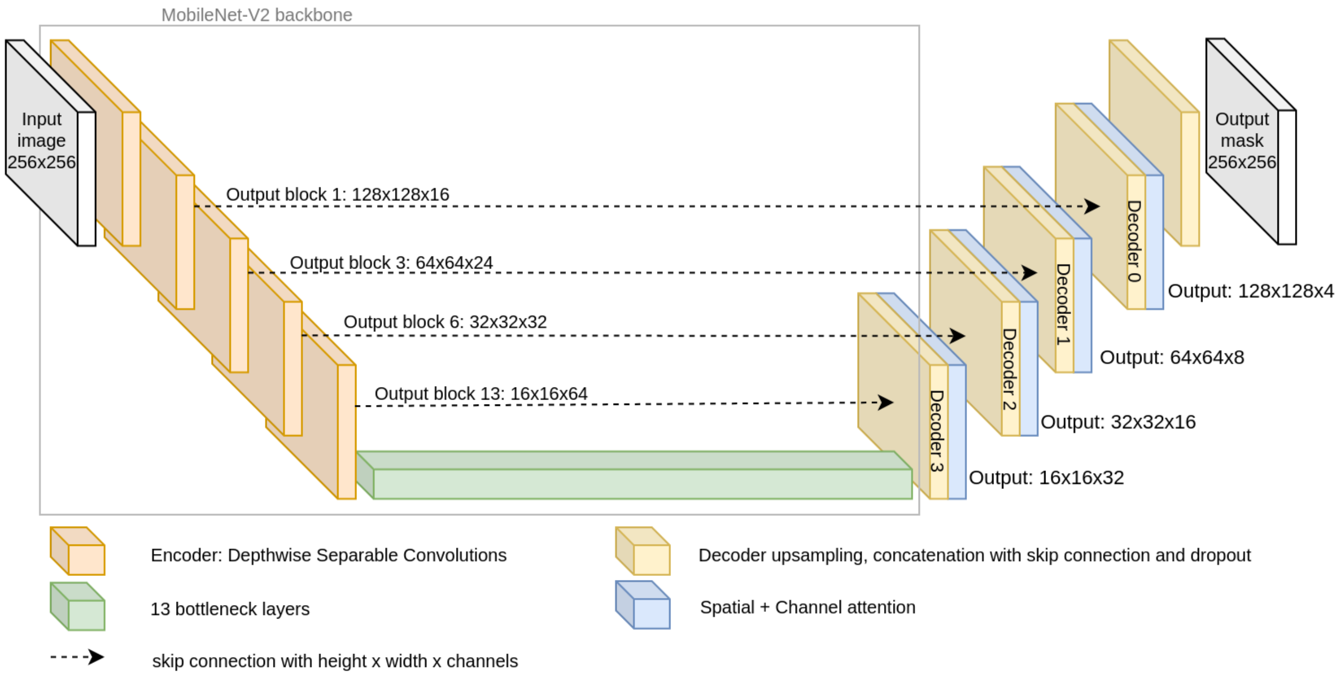


Fig. 11: Simplified architecture of the proposed model

1) Depth Multiplier (α): tested 3 α values to determine optimal encoder width, with the baseline being 0.50. This hyperparameter controls the number of channels (filters) in each layer of the network. α =0.5 was chosen as it provides the best balance between model capacity and parameter efficiency.
2) Attention Mechanisms: Custom spatial and channel attention modules were integrated into each decoder block, after the skip connection fusion, while keeping the pre-trained MobileNetV2 encoder unchanged. Four configurations were tested: no attention (baseline), spatial-only, channel-only, and combined attention. Spatial attention helps the model focus on rock boundaries by emphasizing important spatial regions, while channel attention improves feature representation through adaptive channel weighting. The combined approach was selected for its improved IoU with minimal parameter overhead (3K parameters).
3) Decoder base filters: In the decoder, the number of filters decreases progressively by a factor of 2 across four stages. The starting number of base filters was tested: 16, 32 (baseline) and 64. 32 base filters were chosen as this still showed a high IoU without unnecessary parameter increase.
4) Encoder bottleneck layers: 13 (baseline) layers were chosen after analyzing parameter growth patterns, as increasing beyond block 13 resulted in a significant 50% parameter increase with minimal IoU improvement.

TABLE IV: Ablation study results for architectural components

Component	Configuration	Parameters	Val IoU
Width multiplier α	0.35	144,115	0.532
	0.50	300,203	0.614
	0.75	454,101	0.628
Attention mechanisms	None	300,203	0.614
	Spatial	301,859	0.651
	Channel	301,427	0.643
	Both	303,083	0.672
Decoder base filters	16	255,379	0.598
	32	300,203	0.672
	64	389,851	0.685
Encoder bottleneck layers	10	265,155	0.639
	13	300,203	0.672
	16	458,347	0.688
	19	621,891	0.692

The final model architecture is shown on Figure 11. The configuration features a MobileNetV2 encoder with α =0.5 and 13 bottleneck blocks, spatial and channel attention mechanisms, double convolution blocks in the decoder and a base filter count of 32. A dropout rate of 0.1 was applied to prevent overfitting. The resulting model has 303,083 parameters (1.16 MB model size, below 5 MB of requirement DR-02).

C. *Training Methodology*

To find the optimal training configuration of the model, multiple options were systematically evaluated.

The following loss functions were implemented and evaluated to determine which would be most effective:

- Dice loss: Directly optimises for region overlap, appropriate for imbalanced segmentation tasks

- Focal loss: Addresses class imbalance by down-weighting easy examples.
- Combined loss: Weighted combination of Dice and Focal losses.
- Binary cross-entropy (BCE): Standard loss for binary classification problems.

The results are shown in Table V. Dice loss was selected as the primary training objective because it performed best on all tested metrics, including mean IoU and F1 score.

TABLE V: Loss function comparison

Loss function	Mean Val IoU	Std-dev	F1 score
Dice	0.665	0.169	0.735
Combined	0.600	0.094	0.667
BCE	0.472	0.084	0.641
Focal	0.458	0.157	0.628

For optimisation strategies, multiple algorithms were evaluated:

- Adam: Adaptive moment estimation with momentum and bias correction
- RMSprop: Maintains per-parameter learning rates divided by a moving average of squared gradients
- SGD: Standard stochastic gradient descent with momentum (dropped after initial testing due to not converging)

RMSprop showed as the optimal optimiser, delivering the highest mean IoU (0.720) and F1 score. Adam showed larger variance and occasional divergence, as shown in Table VI. For learning rate, values between 0.0005 and 0.01 were tested, with 0.001 providing the best balance between stability and convergence speed. A cosine learning rate decay was compared to no decay, the latter showed better fine-tuning of the model and increased final IoU. Memory constraints limited the batch size to 4.

TABLE VI: Optimizer comparison

Optimizer	Mean Val IoU	Std-dev	F1 score
RMSprop	0.720	0.131	0.781
Adam	0.458	0.243	0.543

Based on the ablation studies performed, the final training configuration is summarised in Table Table VII.

TABLE VII: Final training configuration

Hyper-parameter	Value
Loss function	Dice
Optimizer	RMSprop
Initial LR	10^{-3}
LR schedule	Cosine decay (end ratio = 0.01)
Batch size	4
Epochs / early-stop	100 / patience = 20
Dropout	0.10
Mixed precision	Yes (`float16`)

After establishing the optimal architecture and training configurations, transfer learning was explored to further improve model performance, especially for adapting to the specific target environment. While ideally ablation studies would be repeated for each dataset, the architectural optimizations were assumed to generalize across Mars datasets due to similar image data, similar annotations, and the same preprocessing and segmentation tasks. Transfer learning is particularly valuable in planetary exploration contexts where labelled data is scarce. A two-stage transfer learning strategy was developed:

1) Base Training: The model was first trained on the MarsData-V2 training set due to its real Mars images, dataset size and annotation quality.
2) Specialization: The base model was then fine-tuned on the MarsTanYard training set, testing varying numbers of frozen encoder layers. This allows the model to specialize to the visual conditions encountered during deployment. The best results were achieved with a completely unfrozen encoder, allowing all weights to be retrained.

Two model variants were compared: the transfer learning model (trained on MarsData-V2 then fine-tuned on MarsTanYard) versus a direct learning model (trained only on MarsTanYard). Both approaches started with ImageNet-pretrained MobileNetV2 encoder weights, and both models were evaluated on the validation sets of both MarsData-V2 and MarsTanYard to assess cross-domain performance.

The results in Table VIII show that direct training on MarsTanYard achieved better performance on the target domain, the validation set of MarsTanYard (0.815 vs 0.795 IoU). However, the transfer learning approach demonstrated better generalization to MarsData-V2 (0.212 vs 0.063 IoU), as expected since this dataset was seen during base training. Additionally, the transfer learning approach made the model converge significantly faster on the MarsTanYard dataset, compared to training it on MarsTanYard directly. The approach converged to a (dice) loss below 0.5 in 4 epochs on average (over 5 runs), and below 0.2 in 27 epochs on average. During training the model directly on MarsTanYard, achieving these losses took on average 32 and 48 epochs respectively.

Despite the theoretical advantages of transfer learning, empirical results showed that direct training on the mission-specific MarsTanYard dataset yielded the best segmentation accuracy for this particular application. This suggests that the domain shift from real Mars imagery to MarsTanYard is substantial enough that transfer learning can introduce harmful priors, and that the MarsTanYard dataset contains sufficient information to train a robust model. For the specific application targeting MarsTanYard deployment, the direct approach was selected based on its superior performance on the target domain, while acknowledging that for missions with greater uncertainty in deployment conditions, the transfer learning approach would offer better generalization.

TABLE VIII: Cross-domain IoU performance on validation sets

Training Strategy	MarsTanYard IoU	MarsData-V2 IoU
Transfer learning (MarsData-V2 $\rightarrow$ MarsTanYard)	0.795	0.212
Direct learning (MarsTanYard only)	0.815	0.063

Fig. 12: Example input image, ground truth segmentation mask, and prediction of the final model on the MarsTanYard dataset.

D. Model Performance Evaluation

An evaluation of the final model's performance was conducted. An experimental setup was defined to ensure reliable performance assessment of the final trained model. The evaluation was performed using the MarsTanYard validation set, which consists of 40 images that were set aside before training to ensure the model had never seen these images during training. The validation set was specifically chosen to represent diverse environmental conditions including varying lighting scenarios, terrain types (rocky, sandy, pebbles, mixed), rock sizes, and camera angles to ensure comprehensive performance assessment.

For evaluation, the trained model processed each validation image through the complete inference pipeline: image preprocessing, neural network inference, and post-processing to generate binary segmentation masks. The evaluation measures the following standard segmentation metrics: Dice score, IoU (Intersection over Union) measuring overlap between predicted and true rock regions, precision quantifying false positive rates, recall measuring missed rock detection, and rock-specific IoU focusing on foreground class performance. These metrics are critical for planetary applications where false positives could trigger unnecessary avoidance maneuvers and false negatives could lead to collisions.

The performance metrics on the MarsTanYard validation set are shown in Table IX. The model achieves 77.0% IoU, demonstrating effective rock segmentation capability and verifying requirement FR-02. The class-specific Rock-IoU of 73.7% confirms that the network can really identify rock regions well and not only the background considering there is a class imbalance, and recall (85.3%) indicate robust detection without excessive false positives or missed rocks. For planetary navigation, the 81.0% precision means approximately 19% of pixels classified as rocks are false positives, while 85.3% recall indicates about 15% of true rock pixels are incorrectly classified as background. This balance favours safety by slightly over-detecting potential rock pixels rather than missing them. The performance in detecting complete physical rocks will be tested and validated in Section VI. The achieved IoU and dice scores meet requirement PR-03. However, an important limitation of our model is that it only detects rocks as obstacles, while planetary rovers face diverse hazards including sand traps, steep terrain, bedrock formations, and crater edges. Our model may miss important terrain features that could impact rover mobility and safety. Another limitation is that it classifies all rocks as a single class rather than detecting individual rock instances. This can result in rock clusters being treated as single large obstacles and occluded rocks being poorly represented in the obstacle map.

Qualitative assessment does confirm that our model accurately detects rocks of varying sizes, shapes, and textures across different lighting conditions, which is what it was designed for and complies with our requirements. An example is shown in Figure 12.

TABLE IX: Final model performance metrics

Metric	Value
IoU	0.770
Rock-IoU	0.737
Dice	0.858
Precision	0.810
Recall	0.853

The benchmark comparison in Table X shows the only meaningful quantitative comparison available, as RockFormer is the only model evaluated on the identical MarsData-V2 dataset. Other proposed DL model approaches, as mentioned in the related work section, cannot be directly compared due to the usage of different datasets, validation splits, and evaluation protocols.

TABLE X: Model segmentation benchmark comparison on MarsData-V2 dataset

Model	Params [M]	Prec. %	Recall %	IoU %
Ours	0.31	94.6	90.7	89.9
RockFormer [27]	6.88	99.0	98.6	93.3

Our model achieves 89.9% IoU compared to RockFormer's 93.3% IoU, this 3.4 percentage point difference represents the cost of our 22-factor parameter reduction (0.31×10^6 vs $6.88 \times$

10^6 parameters). This demonstrates competitive segmentation quality while achieving high computational efficiency. The precision-recall trade-off shows our model at 94.6% precision and 90.7% recall versus RockFormer's 99.0% precision and 98.6% recall. Next to having expected lower values for both, the difference between our false positive and false negative rate (4.4 percentage points) shows more caution, as collision-safety is most important to our rover according to our CONOPS and objectives.

V. Software and Hardware Integration

This section details the integration of the deep learning-based obstacle detection system with the MARTA rover's software and hardware components. The integration encompasses developing software interfaces between the segmentation model and existing rover systems, and optimizing the model for efficient edge inference on the target hardware.

A. Software Integration

The obstacle detection system must interface seamlessly with other software components of the MARTA rover, particularly the camera input system and path planner. The integration required the development of a dedicated ROS2 node that serves as a bridge between the deep learning segmentation model and the navigation stack, to meet requirement IR-01.

A Python-based ROS2 node, named Rock2DMapper, was implemented to transform segmentation masks from the deep learning model into occupancy grid representations usable by the navigation system. The node subscribes to both the segmentation mask topic and the disparity image from the stereo camera, and to the transformation of the camera to the rover base, as can also be seen in the earlier shown ROS2 node functional graph in Figure 15. It then processes this information to generate an occupancy grid that represents obstacles on the surface in front of the rover.

The conversion from 2D segmentation masks to 3D world coordinates involves stereo vision calculations and coordinate frame transformations. First, the stereo camera system provides disparity images where each pixel value represents the horizontal displacement between corresponding points in the left and right camera images. The depth Z_{cam} in the camera coordinate frame is calculated using the standard stereo vision formula:

$$Z_{cam} = \frac{f_x \cdot B}{d \cdot s} \tag{1}$$

where f_x is the focal length in pixels (1304 pixels for the Bumblebee X camera, B is the stereo baseline (known to be 0.240234375 m for the Bumblebee X camera), d is the disparity value from the disparity image, and s is the disparity scale factor (0.015625 for the Bumblebee X camera). Once the depth is known, the 2D pixel coordinates (u, v) of detected rock centers are converted to 3D coordinates in the camera frame using the pinhole camera model [51]:

$$X_{cam} = \frac{(u - c_x) \cdot Z_{cam}}{f_x} \tag{2}$$

$$Y_{cam} = \frac{(v - c_y) \cdot Z_{cam}}{f_y} \tag{3}$$

where (c_x, c_y) are the principal point coordinates and (f_x, f_y) are the focal lengths in the x and y directions, all adjusted for the image preprocessing pipeline that crops and resizes the original camera images to 256x256 pixels.

The 3D coordinates in the camera frame are then transformed to the robot's base frame using the Transform Framework 2 (TF2) system provided by ROS2. TF2 provides the necessary rotation and translation to convert points from the camera's coordinate system to the robot's base coordinate system.

Finally, the system generates a 2D occupancy grid as seen in Figure 13 rather than a full 3D representation for computational efficiency. The 2D approach projects all detected obstacles onto a flat ground plane, assuming relatively flat terrain without hills or big elevation changes. The 3D obstacle coordinates are mapped onto a flat 200x200 cell grid with 0.05 m resolution (complying with requirement FR-05), covering a 5m x 5m area with the robot positioned at the bottom center. Each detected rock is represented by marking corresponding grid cells as occupied, with the number of cells determined by the rock's apparent size in the segmentation mask. While this flat ground assumption reduces mapping accuracy on uneven terrain, it maintains the safety requirements of the system. A nearby rock will be detected regardless of whether the rock or rover is on a slope, ensuring that the rover stops or deviates from its path when approaching obstacles. This mapping accuracy trade-off is acceptable because the primary goal is collision avoidance rather than precise 3D terrain reconstruction.

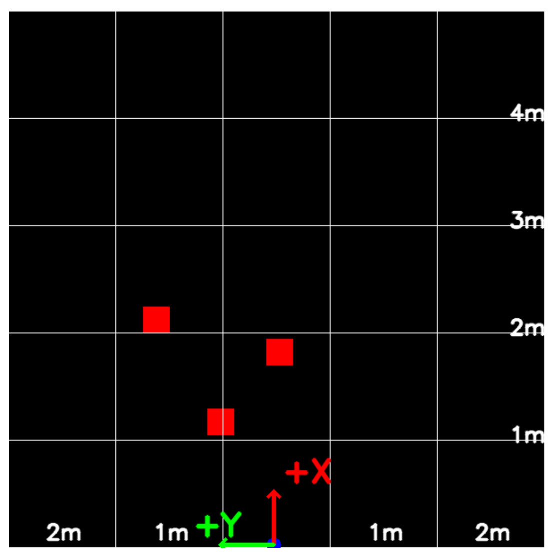


Fig. 13: Example of a 5 x 5 meter occupancy grid in the rover frame generated by the system, at a resolution of 5 cm. The occupied pixels represent the obstacles.

To handle temporal inconsistencies in rock detection caused by segmentation noise, occlusions, or varying lighting conditions, a persistence mechanism was implemented in the occupancy grid. Each cell in the occupancy grid has an associated persistence value that tracks how long an obstacle should remain marked, complying with requirement FR-06 of maintaining an internal map. When a rock is detected, the persistence value is set to a threshold of 3 cycles (3 mapping actions, so 3 seconds when running at 1Hz), and decreases with each cycle if the same rock is not re-detected. Cells are marked as free space when their persistence value reaches zero. This mechanism accounts for the rover's movement speed but assumes no wheel slippage as a simplification. The system ensures that temporarily missed detections do not immediately clear recent obstacle information, while also preventing outdated obstacle data from persisting indefinitely.

B. Target Hardware Integration

For the MaRTA rover, the NVIDIA Jetson AGX Orin was selected as the target hardware platform due to its balance of computational performance and power efficiency for edge AI applications. The Jetson AGX Orin features an Advanced RISC Machine'-based (ARM) central processing unit (CPU) with 12 cores, an NVIDIA 'Ampere architecture GPU with 2048 'Compute Unified Device Architecture' (CUDA) cores and 64 Tensor cores, 32GB of unified memory, and up to 275 Trillions of Operations Per Second (TOPS) of AI performance.

Docker containerization was employed to facilitate system integration and deployment, meeting requirement DR-03. The existing MaRTA ROS2 stack was already containerized for x86 architecture, but deploying on the ARM64-based Jetson platform required creating a new container based on NVIDIA's Linux for Tegra (L4T) images. The new custom-built container incorporates ARM64 architecture support, the NVIDIA JetPack Softare Development Kit (SDK), ROS2 Humble, and the MaRTA ROS2 stack.

The system maximizes on-device GPU utilization. All numerically intensive operations (CNN inference and pixel-level preprocessing) execute on the GPU and Tensor Cores, while the CPU handles ROS2 messaging, TF2 transforms, and the lightweight occupancy grid updates. Unified memory on the Jetson AGX Orin eliminates explicit data copies between CPU and GPU domains. This computational split prioritizes GPU resources for deep learning inference while keeping decision-making tasks on the CPU, an approach that remains valid for future hardware architectures.

For future planetary rovers that may adopt a CPU coupled with an AI accelerator such as Lunar Zebro, the computational distribution would remain conceptually similar. The DL model would execute on the accelerator such as a vision processing unit (VPU), while the CPU would continue managing interfacing between subsystems, coordinate transformations, and occupancy grid operations. Similarly, path planning algorithms would execute on the CPU because their graph search and optimization operations are better suited to general-purpose processors than specialized AI accelerators.

C. Model Optimization

The deep learning model was optimized for efficient inference on the Orin AGX GPU through a multi-stage conversion pipeline, following requirement FR-07. The model was first converted from its original Keras format [52] (the representation used for development with the Keras application programming interface (API) and TensorFlow backend), to Open Neural Network Exchange (ONNX), an intermediate representation that standardizes the model and facilitates optimization. The ONNX model was then converted to TensorRT, NVIDIA's runtime optimizer that significantly accelerates inference on NVIDIA GPUs through optimizations such as layer fusion, kernel auto-tuning, and precision calibration. During the TensorRT conversion, the model was quantized to half-precision floating point (FP16) to reduce memory usage and increase inference speed, with minimal impact on segmentation accuracy (0.1% decrease). The inference performance results using the different model formats are discussed in the next section.

VI. Experimental Results and Analysis

This section presents the experimental validation of the developed obstacle detection system, evaluating both the model performance on the target hardware and the complete system validation through real-world testing on the MaRTA rover platform.

A. Integrated Model Performance

The performance of the optimised and integrated deep learning obstacle detection system was evaluated using two primary metrics: segmentation accuracy and computational efficiency. Segmentation accuracy was assessed across the three model deployment formats: the original Keras model, the ONNX conversion, and the TensorRT engine. To evaluate segmentation quality, the same metrics as earlier were calculated on the validation set of 40 MarsTanYard images, as described earlier.

TABLE XI: Segmentation performance across model formats

Model	IoU	Dice	Precision	Recall	Accuracy
Keras	0.793	0.878	0.877	0.889	0.993
ONNX	0.793	0.878	0.877	0.889	0.993
TensorRT	0.792	0.877	0.876	0.889	0.993

As shown in Table XI, the segmentation performance remained consistent across all three model formats with negligible variation (less than 0.1% difference), confirming that the optimization process preserved prediction quality despite format conversions. The high precision and recall values indicate effective rock detection with minimal false positives and false negatives.

Secondly, the computational efficiency of the model formats was analysed in terms of inference time, throughput, memory usage, and thermal characteristics during execution. Inference time was measured using system timers that capture the complete model execution cycle. The measurement boundaries were defined from immediately before transferring the preprocessed input image to the model's execution environment until the final prediction is available for post-processing. A warm-up period of 10 inferences was performed before measurements to ensure a consistent GPU state. All inference times were collected over 100 runs to obtain statistically significant results.

TABLE XII: Computational performance metrics across model formats run on CPU and GPU

model	mean time	std dev	P95 time	throughput
Keras (CPU)	505.57 ms	20.32 ms	548.89 ms	2.0 fps
Keras (GPU)	150.72 ms	6.03 ms	160.56 ms	6.6 fps
ONNX (CPU)	101.88 ms	19.64 ms	128.66 ms	9.8 fps
ONNX (GPU)	48.56 ms	8.14 ms	62.64 ms	20.6 fps
TensorRT (GPU)	4.49 ms	0.91 ms	6.36 ms	222.5 fps

The results in Table XII demonstrate significant performance gains through model optimisations and through running on GPU. The ONNX model on GPU improved inference time by 3.1 times compared to the Keras model on GPU, because of its graph optimisations like operator fusion. TensorRT provided the most dramatic improvement, reducing inference time 33.5 times when compared to Keras on GPU and 10.8 times when compared to ONNX on GPU. The improvement between running the TensorRT model on GPU and the Keras model on CPU is over two orders of magnitude. The TensorRT implementation also showed superior stability and consistency with the lowest standard deviation and P95 time (meaning 95% of inferences complete within this time). This shows that the TensorRT model would leave ample resources for other rover subsystems, making it very suitable for integration with other processes.

The TensorRT implementation achieved a peak runtime memory footprint of only 76 MB (meeting the PR-06 requirement of max 100MB), compared to 148 MB for ONNX and 215 MB for the Keras FP32 model, a 2.8-fold reduction from Keras to TensorRT. This improvement comes from both halving precision (FP32 to FP16) and the elimination of redundant intermediate tensors through operator fusion.

Thermal monitoring showed all implementations maintained safe temperatures on the Jetson platform. The TensorRT implementation recorded peak temperatures of 49.2°C in processing zones and 45.8°C in memory zones, nearly identical to Keras (49.3°C/45.3°C) and ONNX (49.4°C/45.8°C). These minimal differences of less than 0.2°C suggest that the brief inference operations have limited thermal impact on the Nvidia Orin AGX. Requirement PR-07 is met as the GPU temperature stays below 50°C.

With TensorRT processing frames in 4.49 ms on average, the obstacle detection operates well faster than the inference time requirement of 0.05 seconds (PR-01), consuming minimal computational resources and allowing other rover systems to run simultaneously on the same hardware.

B. Real-World Testing on Rover Platform

To verify and validate the system, the obstacle detection system was evaluated in real-world conditions: running on the MaRTA rover while encountering varying terrains on the Mars Yard. The approach is shown in Figure 14, with the activities explained in the next subsections.

1) Test Setup and Validation Methodology: Since the MaRTA rover does not yet have autonomous navigation nodes integrated, motor and joint commands were provided manually using a gamepad to control the rover's movement over terrain and adjust the pan-tilt camera directions. This setup preserved

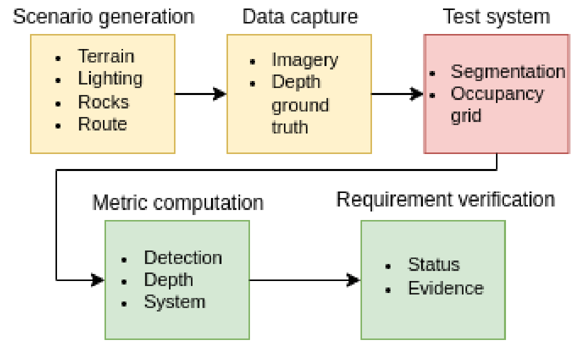


Fig. 14: Systematic framework used to verify the system requirements

the complete perception stack intact, enabling end-to-end testing with the same input and output interfaces that would be used in autonomous operation, as was shown in the earlier functional diagram in Figure 15.

The experimental setup consisted of the MaRTA rover positioned in the Mars Yard facility to have a semi-realistic Mars setup, partially meeting requirement VR-01, with two distinct lighting configurations: directional ”sun light” using a single bright lamp to create shadows at a 15% inclination as measured from the center of the Mars Yard, and even ”laboratory lighting” using distributed overhead illumination. Ground truth measurements of rock depths were collected using a laser rangefinder, measured from the left camera to the rock's front surface.

Data collection followed a systematic approach with 82 static images captured across controlled conditions, with MaRTA following the route outlined Figure 16. The lighting conditions within the dataset were distributed to be 50% sun light (sun being a bright lamp as single light source) and 50% laboratory lighting, containing each exact environmental configuration in both lighting conditions. The terrain type distribution present in the dataset is as follows: 15% pebbles, 20% flat rocks, 50% regular sandy ground, 15% mixed terrain. These environments were selected to represent the diverse conditions the rover might encounter during deployment and fulfill requirement VR-03.

Rocks were distributed throughout test environments following a semi-structured approach, ensuring a representative range of scenarios relevant to the rover's operational environment.

1) Each test image contained on average 4 (between 2 and 8) rocks of various sizes, distributed as:
   - Small rocks ($<$ 10 cm height): 30% of rocks
   - Edge-case rocks (10-12 cm height): 20% of rocks
   - Large rocks ($>$ 12 cm height): 50% of rocks
2) Rocks were positioned at distances ranging from 0.5 m to 8 m from the rover, with higher concentration in the 1-4 m range.
3) Test configurations are:
   - Rocks of different colors placed on various background textures
   - Isolated rocks on uniform terrain
   - Clustered rocks

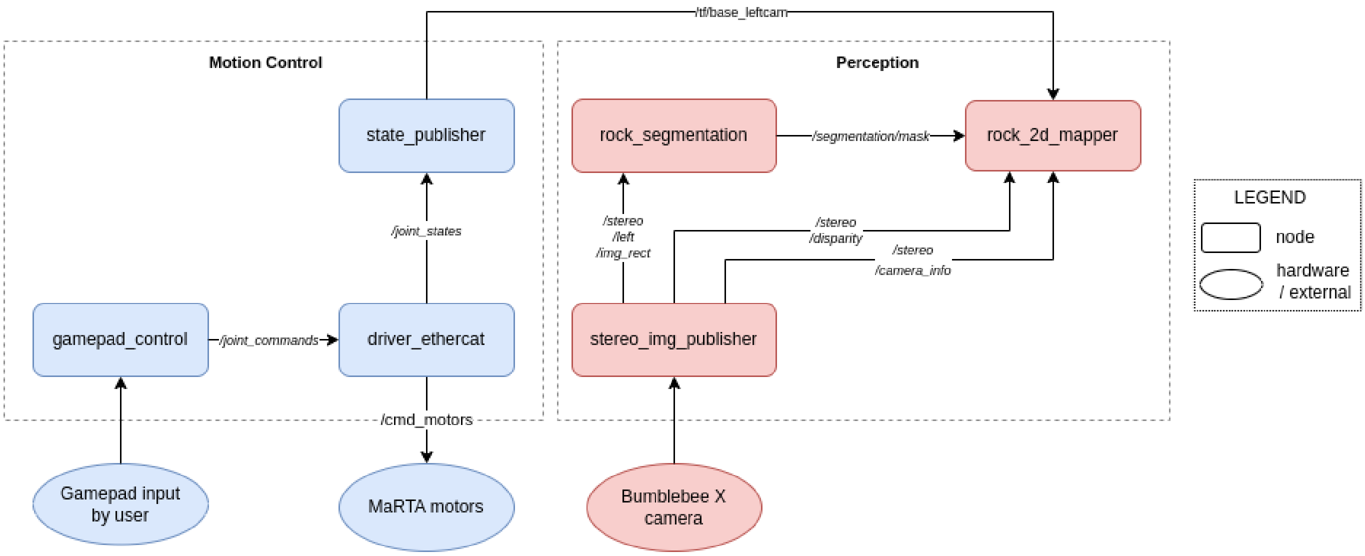


Fig. 15: ROS2 node graph used for this research, with manual rover control provided by gamepad, and showing the same perception pipeline components.

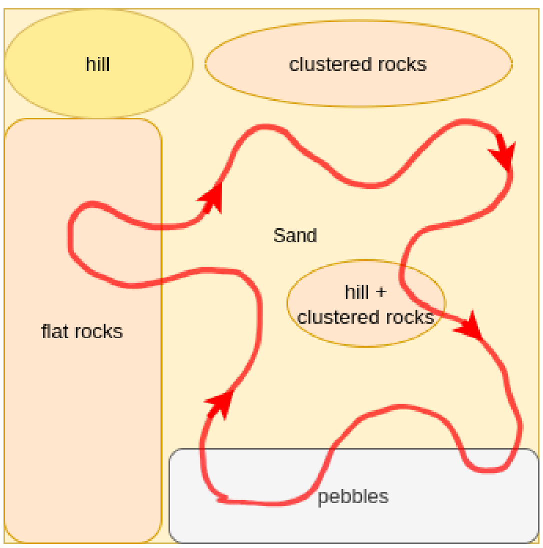


Fig. 16: Schematic map of the Mars Yard and MaRTA's route for data collection outlined in red.

- Rocks in shadow regions and under varying illumination

To establish consistent ground truth, explicit classification criteria were defined:

- Rocks with height $>$ 12 cm: Classified as obstacles that should be detected
- Rocks with height $<$ 12 cm: Classified as traversable (should not be detected)

The validation methodology systematically evaluated each component through controlled ground truth collection. Each processing stage was validated using the following methodologies:

1) Detection validation: Expert opinion of the test images identifying rocks that constitute true obstacles according to the classification criteria. This provided ground truth for evaluating object-level detection (whether a rock is detected at all).
2) Depth validation: Laser rangefinder measurements provided ground truth. The transformation from depth to 3D coordinates and subsequent mapping to the occupancy grid was performed to demonstrate the complete pipeline, but not explicitly validated with ground truth positioning. This was because precise positional ground truth would require specialized equipment not available for this study, and would primarily validate the transformation from camera to rover base_link frame, which was outside the scope of this research.
3) System-level validation: end-to-end performance was evaluated from camera input to mapper output, measuring the system's ability to identify and locate obstacles that should be avoided within safety constraints. This did not include actual path planning or avoidance manoeuvres.

This validation approach constitutes a systematic methodology to ensure reproducibility through several key characteristics: explicit test factors (terrain types, lighting conditions, rock sizes, distance ranges), controlled data captured under both lighting conditions with laser rangefinder ground truth and outlined procedures such as the route driven during data collection.

2) Performance Results and Analysis: Figure 17 shows examples of the complete processing pipeline, following requirement VR-02, for the same environment under both lab and sun light conditions. From left to right it shows the input image, disparity image with segmentation contours, mask with detected rocks and their depth values and coordinates, and the final map with rock position and sizes in the rover frame.

Table XIII presents object-level detection rates across rock categories and lighting conditions, focusing on whether rocks

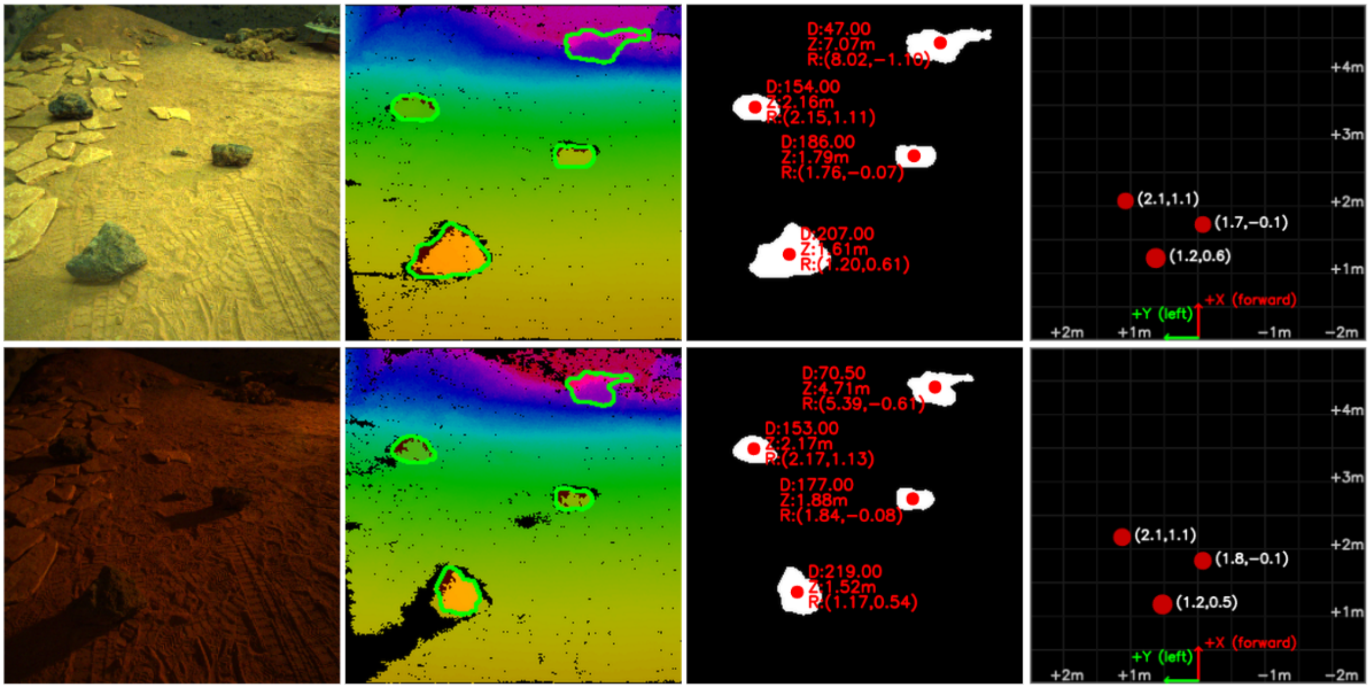


Fig. 17: Example results of the image pair of the same environment tested with lab lighting (above) and sun light (below). For each row, the first image shows the input image. The second image shows the DL-generated rock segment contours overlayed on the camera disparity image. The third image shows the DL rock segmentation with for each detected rock, the disparity, depth, and position values. The last image is a visualisation of where the detected rocks are placed in the terrain in front of the rover.

are detected rather than segmentation quality, as this directly matters to the rover concept of operations. The detection rate is averaged over all rocks in all images, and the min-max single img provides the minimum and maximum detection percentages within a single image.

TABLE XIII: Segmentation performance by rock characteristics, both light conditions

Rock size	avg detection	std dev	min-max img
sun light			
Too small (<10 cm)	8%	±12.5%	0% - 50%
Edge cases (10-12 cm)	40%	±25%	0% - 100%
Large (>12 cm)	94%	±3.8%	85% - 100%
lab light			
Too small (<10 cm)	30%	±18.8%	0% - 75%
Edge cases (10-12 cm)	74%	±12.5%	50% - 100%
Large (>12 cm)	82%	±8.5%	66% - 100%

Large rocks (>12 cm) exhibited high detection reliability in sun light (94%) with minimal variability (±3.8%), but showed a 12% performance degradation in laboratory lighting due to the domain gap. Edge-case rocks (10-12 cm) demonstrate significant differences between lighting conditions, with detection rates increasing from 40% in sun light to 74% in laboratory conditions. The high variability (±25% in sun light) indicates detection instability for this size category, which is expected behavior since these rocks represent edge cases for obstacle avoidance. Small rocks (<10 cm) appropriately showed low detection in sun light (8%) but higher false positive rates in laboratory lighting (30%). This suggests the model perceives these rocks as larger under even illumination.

Environmental sensitivity is a limitation. The pattern suggest that the segmentation model relies on shadow cues for large rock detection, which are more prominent in directional sun light. Conversely, the even illumination of laboratory lighting enhances the visibility of smaller rocks, making them look more like big rocks. While large rocks maintain reasonable consistency in sun light (±3.8%), their detection stability decreases in laboratory light conditions (±8.5%). Edge cases show extreme variability in both lighting conditions, with standard deviations reaching ±25% in sun light, indicating inconsistent detection. The high min-max ranges across all categories show that there is a substantial variance between the images due to environmental changes such as colors, light incident angles and textures. Overall, rocks over 12 cm in height are detected in sun light over 90% of the time, verifying requirement FR-01 and partially requirement PR-04. Also, rock detection degrades with less than 15% on average between different lighting conditions, verifying PR-05.

TABLE XIV: Mapper performance by rock distance, both light conditions

Real position	avg err	median err	std dev	min-max img
sun light				
Close (<2 m)	32%	20%	±48.5%	6% - 200%
Medium (2-3.5 m)	41%	38%	±15%	13% - 73%
Far (>3.5 m)	86%	48%	±126.8%	31% - 543%
lab light				
Close (<2 m)	18%	16%	±7.5%	4% - 34%
Medium (2-3.5 m)	34%	32%	±9%	15% - 51%
Far (>3.5 m)	82%	41%	±156.3%	35% - 670%

Depth estimation accuracy was evaluated by comparing the system's calculated depths with the laser rangefinder ground truth measurements, as summarized in Table XIV. Close-range accuracy shows high variability in sun light (±48.5%) with errors up to 200%, while laboratory conditions yield significantly more consistent results (±7.5%). The reason for this can be seen in Figure 17: in some sun light images, the disparity values are missing because the camera failed to match stereo features, especially in shadow regions. Although black disparity pixels are not used, this still caused less reliable depth estimation as fewer disparity pixels are available for calculation, sometimes even none at all. Since outliers affect the average error significantly, the median error shows that the impact is less critical than the average suggests.

The erroneous obstacle depth estimation is a limitation of the system. Far-range measurements (>3.5 m) show high average error rates with high inconsistency in both lighting conditions, with maximum errors exceeding 500%. The median error shows less serious errors. Laboratory lighting generally produced more accurate and consistent depth estimates, especially at close and medium ranges, likely due to more reliable disparity values. Overall, position error increases dramatically with distance, making far-field measurements unreliable for efficient rover navigation. Requirement FR-03 is verified because rocks within 4 meters are generally detected, although their position errors beyond 3.5m are significant.

The mapping process combines segmentation and depth estimation, inheriting limitations from both components. These results show operational constraints for rover deployment:

- Operational range: Given the extremely high position errors at distances beyond 3.5 m, the effective operational range must be limited to close and medium distances.
- Safety margins: The detection variability for edge-case rocks (10-12 cm), combined with high position error variability, may require increasing safety margins by classifying edge-case rocks as obstacles.
- Environmental adaptation: The significant performance differences between lighting conditions demonstrate that environmental adaptation will be necessary for deployment in varying illumination scenarios.

3) System-Level Validation: The primary requirements from the concept of operations were:

1) Safety: avoid collisions with obstacles that could damage the rover
2) Efficiency: detect obstacles at sufficient range for efficient path planning

For direct rover safety analysis addressing mission objectives MO-01 and MO-02, Table XV presents a cross-validation of detection and depth estimation performance for safety-critical scenarios involving large rocks at close range (< 2 m).

For large rocks at close range in sun light, the system demonstrated robust performance metrics with direct implications for operational safety:

- 94% detection rate (6% single-frame miss probability)

TABLE XV: Critical safety performance analysis at close range

Rock type	**Lighting condition**	**Detection rate (%)**	**Depth error (%)**	**Collision Prob. (per rock)**
Large	sun light	94	32	6.0e-8
Large	Lab	82	20	1.2e-3

- 32% average depth error, within acceptable limits for emergency stopping or avoidance when the rock is within 2m
- Temporal processing: 1Hz frame rate, with rocks typically visible for approximately 20 frames before reaching collision distance (0.0m)

The frame-to-frame independence is assumed to be 75%, measured from the rock detection results by the segmentation model. At a speed of 10 cm/s with processing at 1 Hz, the subsequent scenes captured by the rover's camera are similar, and the segmentation results showed that a false negative rock detection would persist for on average 75% of the time. This yields an adjusted miss probability of approximately 7.8×10^{-7} per rock encounter, using the following formula:

$$P_{\text{miss}} = (P_{\text{single frame miss}})^{n_{\text{frames}} \cdot f_{\text{independence}}} \quad (4)$$

$$= (0.06)^{20 \cdot 0.25} \quad (5)$$

$$\approx 7.8 \times 10^{-7} \text{ per rock encounter} \quad (6)$$

With an average of 2 significant obstacles per meter, translating to 720 obstacles per hour at maximum speed (10 cm/s), the theoretical collision probability is roughly one event every 149 days ($\approx$ 5 months) during continuous operation of 12 hours per day. For typical planetary rover missions like Lunar Zebro's 14-day duration [46], this would mean there is a 9.0% chance that the rover would collide with a rock during its mission.

However, this analysis assumes frame dependence, to which it is extremely sensitive, and assumes that 32% depth errors are manageable for obstacle avoidance, which is reasonable given the occupancy grid's safety margins. The significant environmental sensitivity, particularly to lighting conditions, suggests that unexpected behavior may arise in unseen environments.

The complete assessment of requirement compliance and mission objective achievement is presented in section B, which traces each requirement to its verification activity and demonstrates that the developed system successfully achieves all five core mission objectives while meeting the majority of requirements.

VII. Discussion and Conclusion

This research addressed the development of a lightweight deep learning obstacle detection system for planetary rovers, following a systems engineering approach from requirements specification to validation. This section discusses key findings, limitations, and future directions.

A. Research Questions and Contributions Revisited

The primary question examined how deep learning models for obstacle detection can be optimised, deployed, and integrated on planetary rovers to enable low-latency, onboard inference. The research demonstrated that:

1) A lightweight MobileNetV2-based U-Net with dual attention mechanisms provides a balance between accuracy and parameter efficiency. Deployment in TensorRT format facilitates low-latency inference at 4.49 ms.
2) The ROS2-based modular integration approach facilitated interfacing between the subsystems from camera input to obstacle mapping. The TensorRT model was optimised to run on the NVIDIA Jetson AGX Orin.
3) The integrated system validated on the MaRTA rover in the ESA Mars Yard demonstrated a 94% detection rate for large obstacles at close range under sunlight conditions, translating to a theoretical collision probability of approximately 7.8×10^{-7} per rock encounter.

The research contributed the following innovations to the field of autonomous planetary DL obstacle detection:

- Dataset Creation: The semi-automated annotation methodology combining CLIPSeg and SAM models significantly reduced manual effort while maintaining high annotation quality. This approach enabled the creation of the mission-specific MarsTanYard dataset, facilitating domain adaptation.
- Lightweight Architecture: The optimised model with 0.31×10^6 parameters achieved a 77.0% IoU a 85.8% Dice score).
- Deployment Optimization: The comprehensive optimization pipeline reduced not only inference time but also memory footprint (2.8× reduction), demonstrating a practical approach for edge deployment on resource-constrained hardware.
- Validation Methodology: A systematic testing framework was developed that evaluated performance across diverse terrain types, lighting conditions, and obstacle configurations. This methodology established quantifiable metrics for both segmentation quality, rock detection, and obstacle mapping, providing an evidence-based approach to validate the mission objectives and verify the system requirements.

B. Future Research Directions

Limitations of the current system and subsystems were identified and mentioned throughout this research. Proposed directions for future research are:

- Multi-Modal Sensing: Integration of thermal imaging or LiDAR could enhance detection robustness in challenging lighting conditions, providing complementary information where visual features are difficult to distinguish.
- Advanced Architectures: Exploration of newer backbone architectures like MobileNetV3 and different decoders could further improve the efficiency-performance trade-off.
- Space-Grade Hardware Deployment: Testing on radiation-tolerant hardware (e.g., Intel Myriad X VPU or radiation-hardened FPGAs) would provide critical insights into deployment feasibility for actual planetary missions.
- Uncertainty Estimation: Incorporating uncertainty quantification for hazards would enable more informed decision-making for path planning, allowing conservative approaches when confidence is low in the obstacle detection.
- Multi-Class Terrain Segmentation: Future research should expand beyond binary rock detection to include more classes that could pose risks for the rover's safety or mission efficiency. This would provide a better environmental understanding for autonomous planning.
- Instance-Level Rock Detection: Transitioning from semantic to instance segmentation would enable the detection of individual rocks and a more accurate occupancy grid representation. This approach would better handle rock clusters and occlusions, improving path planning efficiency.
- Interpretable Decision-Making: The current system makes black-box decisions about which rocks constitute obstacles based on learned features, without explicit size analysis. An alternative approach would be to detect all rocks first and then apply explicit size-based filtering, which would make the decision-making more transparent and controllable.

C. Concluding Remarks

This research has shown that deep learning-based obstacle detection can work effectively on a resource-limited planetary test rover. The system meets the main safety requirements with measurable performance results, achieving processing speeds well above what is needed. The obstacle detection rates and mapping precision ensure safe rover operation, but limit efficient path planning due to high mapping errors at rock distances beyond 3.5 m.

As planetary exploration continues to advance, the approaches developed here (model optimization, system integration, and testing methods) provide a basis for improved rover autonomy. Future work on combining multiple sensors, implementing on space-qualified hardware, and handling uncertainty will be essential for bringing these methods to real planetary missions. This will ultimately enable faster and better scientific outcomes through more efficient navigation and less dependence on Earth-based mission control.

References

[1] M. De Benedetti, S. Kay, J. Ocón, R. Jalvo, M. E. Cerezo, A. Gómez Eguíluz, M. Alonso, J. R. Fernandez, K. Buckley, R. Field, A. Cameron, V. Papantoniou, A. Papantoniou, C. P. Del Pulgar, K. Kapellos, and M. Azkarate, "RAPID & FASTNAV Projects: High-Speed Semi-Autonomous Rovers Enabling High Return Planetary Missions," in *2024 International Conference on Space Robotics (iSpaRo)*. Luxembourg, Luxembourg: IEEE, Jun. 2024, pp. 260–265. [Online]. Available: https://ieeexplore.ieee.org/document/10688088/

[2] D. Gaines, G. Doran, M. Paton, B. Rothrock, J. Russino, R. Mackey, R. Anderson, R. Francis, C. Joswig, H. Justice, K. Kolcio, G. Rabideau, S. Schaffer, J. Sawoniewicz, A. Vasavada, V. Wong, K. Yu, and A. Agha-mohammadi, "Self-reliant rovers for increased mission productivity," *Journal of Field Robotics*, vol. 37, no. 7, pp. 1171–1196, Oct. 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/rob.21979

[3] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Proceedings, IEEE Aerospace Conference*, vol. 5, Mar. 2002, pp. 5–5. [Online]. Available: https://ieeexplore.ieee.org/document/1035370/?arnumber=1035370

[4] J. E. Loh, G. H. Elkaim, and R. E. Curry, "Roughness Map for Autonomous Rovers."

[5] A. Rankin, T. Del Sesto, P. Hwang, H. Justice, M. Maimone, V. Verma, and E. Graser, "Perseverance Rapid Traverse Campaign," in *2023 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2023, pp. 1–16. [Online]. Available: https://ieeexplore.ieee.org/document/10115835/

[6] V. Verma, M. W. Maimone, D. M. Gaines, R. Francis, T. A. Estlin, S. R. Kuhn, G. R. Rabideau, S. A. Chien, M. M. McHenry, E. J. Graser, A. L. Rankin, and E. R. Thiel, "Autonomous robotics is driving Perseverance rover's progress on Mars," *Science Robotics*, vol. 8, no. 80, p. eadi3099, Jul. 2023. [Online]. Available: https://www.science.org/doi/10.1126/scirobotics.adi3099

[7] B. Kuang, C. Gu, Z. A. Rana, Y. Zhao, S. Sun, and S. G. Nnabuife, "Semantic Terrain Segmentation in the Navigation Vision of Planetary Rovers—A Systematic Literature Review," *Sensors*, vol. 22, no. 21, p. 8393, Jan. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8393

[8] J. Guo, X. Zhang, Y. Dong, Z. Xue, and B. Huang, "Terrain classification using mars raw images based on deep learning algorithms with application to wheeled planetary rovers," *Journal of Terramechanics*, vol. 108, pp. 33–38, Aug. 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0022489823000344

[9] T. Hu, F. Han, T. Cao, B. Zheng, Z. Qian, L. He, and Y. Lei, "Unstructured Small Target Obstacle Detection Technology on Lunar Surface Driven by Neural Network," in *2023 2nd Conference on Fully Actuated System Theory and Applications (CFASTA)*. Qingdao, China: IEEE, Jul. 2023, pp. 1022–1027. [Online]. Available: https://ieeexplore.ieee.org/document/10243276/

[10] G. Petrakis and P. Partsinevelos, "Lunar ground segmentation using a modified U-net neural network," *Machine Vision and Applications*, vol. 35, no. 3, p. 50, May 2024. [Online]. Available: https://link.springer.com/10.1007/s00138-024-01533-3

[11] W. Feng, L. Ding, R. Zhou, C. Xu, H. Yang, H. Gao, G. Liu, and Z. Deng, "Learning-Based End-to-End Navigation for Planetary Rovers Considering Non-Geometric Hazards," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4084–4091, Jul. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10138604/

[12] N. Silva, R. Lancaster, and J. Clemmet, "EXOMARS ROVER VEHICLE MOBILITY FUNCTIONAL ARCHITECTURE AND KEY DESIGN DRIVERS," 2013.

[13] C. Wong, E. Yang, X.-T. Yan, and D. Gu, "Adaptive and intelligent navigation of autonomous planetary rovers — A survey," in *2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. Pasadena, CA, USA: IEEE, Jul. 2017, pp. 237–244. [Online]. Available: http://ieeexplore.ieee.org/document/8046384/

[14] M. Azkarate, L. Gerdes, L. Joudrier, and C. J. Pérez-del Pulgar, "A GNC Architecture for Planetary Rovers with Autonomous Navigation Capabilities," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 3003–3009. [Online]. Available: http://arxiv.org/abs/1911.09975

[15] A. Shaukat, S. Al-Milli, A. Bajpai, C. Spiteri, G. Burroughes, Y. Gao, D. Lachat, and M. Winter, "NEXT-GENERATION ROVER GNC ARCHITECTURES," 2015.

[16] M. McHenry, N. Abcouwer, J. Biesiadecki, J. Chang, T. D. Sesto, A. Johnson, T. Litwin, M. Maimone, and J. Morri, "MARS 2020 AUTONOMOUS ROVER NAVIGATION."

[17] J. Carsten, A. Rankin, D. Ferguson, and A. Stentz, "Global Path Planning on Board the Mars Exploration Rovers," in *2007 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, 2007, pp. 1–11. [Online]. Available: http://ieeexplore.ieee.org/document/4161272/

[18] N. Abcouwer, S. Daftry, S. Venkatraman, T. d. Sesto, O. Toupet, R. Lanka, J. Song, Y. Yue, and M. Ono, "Machine Learning Based Path Planning for Improved Rover Navigation (Pre-Print Version)," Nov. 2020, arXiv:2011.06022 [cs]. [Online]. Available: http://arxiv.org/abs/2011.06022

[19] R. M. Swan, D. Atha, H. A. Leopold, M. Gildner, S. Oij, C. Chiu, and M. Ono, "AI4MARS: A Dataset for Terrain-Aware Autonomous Driving on Mars," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Nashville, TN, USA: IEEE, Jun. 2021, pp. 1982–1991. [Online]. Available: https://ieeexplore.ieee.org/document/9523149/

[20] M. Xiaoyan, M. Zhifu, C. Jianxin, L. Zhiping, T. Baoyi, and X. Yan, "Parallel Design and Implementation of Stereo Vision Algorithm of Zhurong Mars Rover."

[21] J. Wang, G. Hu, D. Li, S. Wang, S. Han, X. Li, X. Liu, Z. Cheng, H. Zhang, Z. Huang, X. He, and X. Wang, "Application of Computer Vision Technology in Collaborative Control of the "Zhurong" Mars Rover," in *Image and Graphics Technologies and Applications*, W. Yongtian and W. Lifang, Eds. Singapore: Springer Nature Singapore, 2023, vol. 1910, pp. 425–439, series Title: Communications in Computer and Information Science. [Online]. Available: https://link.springer.com/10.1007/978-981-99-7549-5_31

[22] J. Wang, J. Li, S. Wang, T. Yu, Z. Rong, X. He, Y. You, Q. Zou, W. Wan, Y. Wang, S. Gou, B. Liu, M. Peng, K. Di, Z. Liu, M. Jia, X. Xin, Y. Chen, X. Cheng, X. Feng, C. Liu, S. Han, and X. Liu, "COMPUTER VISION IN THE TELEOPERATION OF THE YUTU-2 ROVER," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-3-2020, pp. 595–602, Aug. 2020. [Online]. Available: https://isprs-annals.copernicus.org/articles/V-3-2020/595/2020/

[23] N. J. Kanu, E. Gupta, and G. C. Verma, "An insight into India's Moon mission – Chandrayan-3: The first nation to land on the southernmost polar region of the Moon," *Planetary and Space Science*, vol. 242, p. 105864, Mar. 2024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S003206332400028X

[24] P. Tian and M. Yao, "RSU-Net: An Attention U-Net for Martian Rock Segmentation," *Journal of Physics: Conference Series*, vol. 2762, no. 1, p. 012001, May 2024. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/2762/1/012001

[25] S. Sethy, S. K. Behera, J. Ramadevi, P. K. Sethy, and P. Biswas, "Rock Segmentation of Real Martian Scenes Using Dual Attention Mechanism-Based U-Net," in *Proceedings of 4th International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*, V. K. Gunjan and J. M. Zurada, Eds. Singapore: Springer Nature Singapore, 2024, vol. 873, pp. 117–124. [Online]. Available: https://link.springer.com/10.1007/978-981-99-9442-7_11

[26] C. Ma, Y. Li, J. Lv, Z. Xiao, W. Zhang, and L. Mo, "Automated Rock Detection From Mars Rover Image via Y-Shaped Dual-Task Network With Depth-Aware Spatial Attention Mechanism," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–18, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10453623/

[27] H. Liu, M. Yao, X. Xiao, and Y. Xiong, "RockFormer: A U-Shaped Transformer Network for Martian Rock Segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–16, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10012398/

[28] L. Fan, J. Yuan, X. Niu, K. Zha, and W. Ma, "RockSeg: A Novel Semantic Segmentation Network Based on a Hybrid Framework Combining a Convolutional Neural Network and Transformer for Deep Space Rock Images," *Remote Sensing*, vol. 15, no. 16, p. 3935, Aug. 2023. [Online]. Available: https://www.mdpi.com/2072-4292/15/16/3935

[29] C. Ma, Y. Li, Z. Xiao, W. Zhang, L. Mo, and A. Li, "SimMars6K," Mar. 2023. [Online]. Available: https://zenodo.org/records/7707898

[30] "CVIR-Lab/SynMars," Aug. 2024, original-date: 2022-09-03T06:29:33Z. [Online]. Available: https://github.com/CVIR-Lab/SynMars

[31] L. Meyer, M. Smíšek, A. Fontan Villacampa, L. Oliva Maza, D. Medina, M. J. Schuster, F. Steidle, M. Vayugundla, M. G. Müller, B. Rebele, A. Wedler, and R. Triebel, "The MADMAX data set for visual-inertial rover navigation on Mars," *Journal of Field Robotics*, vol. 38, no. 6, pp. 833–853, Sep. 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/rob.22016

[32] M. Azkarate, "Katwijk Beach Planetary Rover Dataset – ESA Robotics Datasets." [Online]. Available: https://robotics.estec.esa.int/datasets/katwijk-beach-11-2015/

[33] D. Rijlaarsdam, T. Hendrix, P. T. T. González, A. Velasco-Mata, L. Buckley, J. P. Miquel, O. A. Casaled, and A. Dunne, "The Next Era for Earth Observation Spacecraft: An Overview of CogniSAT-6," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, pp. 2450–2463, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10772069/

[34] D. Ali, A. U. Rehman, and F. H. Khan, “Hardware Accelerators And Accelerators For Machine Learning,” in *2022 International Conference on IT and Industrial Technologies (ICIT)*. Chiniot, Pakistan: IEEE, Oct. 2022, pp. 01–07. [Online]. Available: https://ieeexplore.ieee.org/document/9989124/

[35] G. Giuffrida, L. Fanucci, G. Meoni, M. Batic, L. Buckley, A. Dunne, C. Van Dijk, M. Esposito, J. Hefele, N. Vercruyssen, G. Furano, M. Pastena, and J. Aschbacher, “The -Sat-1 Mission: The First On-Board Deep Neural Network Demonstrator for Satellite Earth Observation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9600851/

[36] A. Marin, C. Coelho, F. Deconinck, I. Babkina, N. Longepe, M. Pastena, N. Longepe, and M. Pastena, “-Sat-2: Onboard AI Apps for Earth Observation,” 2021.

[37] “The Discovery Campaign on OPS-SAT experiments.” [Online]. Available: https://www.esa.int/Enabling_Support/Preparing_for_the_Future/Discovery_and_Preparation/The_Discovery_Campaign_on_OPS-SAT_experiments?utm_source=chatgpt.com

[38] “Lockheed Martin and USC to Launch Jetson-Based Nanosatellite for Scientific Research Into Orbit,” Aug. 2020. [Online]. Available: https://developer.nvidia.com/blog/lockheed-martin-usc-jetson-nanosatellite/

[39] M. Cross, A. J. Macdonald, B. Bonham-Carter, H. Burd, L. Chavier, T. Heydrich, S. Pillay, E. Smal, M. Maharib, K. Raimalwala, M. Faragalli, and M. Battler, “Enabling Autonomy and Operations for Lunar Surface Missions: An Overview of Demonstrated Capabilities.”

[40] T. Bolscher, “Designing the Brain of an Intelligent Lunar Nano-rover,” Master’s thesis, 2023. [Online]. Available: https://repository.tudelft.nl/record/uuid:8938dd1c-683f-49b5-a607-75fe8c4b6447

[41] T. Weilkiens, J. G. Lamm, S. Roth, and M. Walker, *Model-Based System Architecture*, 1st ed. Wiley, Sep. 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119051930

[42] M. A. Viscio, N. Viola, R. Fusaro, and V. Basso, “Methodology for requirements definition of complex space missions and systems,” *Acta Astronautica*, vol. 114, pp. 79–92, Sep. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0094576515001745

[43] E. Allouis, R. Blake, S. Gunes-Lasnet, T. Jorden, B. Maddison, H. Schroeven-Deceuninck, M. Stuttard, P. Truss, K. Ward, R. Ward, and M. Woods, “A FACILITY FOR THE VERIFICATION & VALIDATION OF ROBOTICS & AUTONOMY FOR PLANETARY EXPLORATION.”

[44] “verification | European Cooperation for Space Standardization.” [Online]. Available: https://ecss.nl/item/?glossary_id=2690

[45] “Planetary Robotics Laboratory.” [Online]. Available: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Planetary_Robotics_Laboratory

[46] “Lunar Zebro | Nano Rover TU Delft.” [Online]. Available: https://zebro.tudelft.nl/

[47] M. Yao, “MarsData-V2, a rock segmentation dataset of real Martian scenes,” Sep. 2022. [Online]. Available: https://ieee-dataport.org/documents/marsdata-v2-rock-segmentation-dataset-real-martian-scenes

[48] “Artificial Lunar Landscape Dataset.” [Online]. Available: https://www.kaggle.com/datasets/romainpessia/artificial-lunar-rocky-landscape-dataset

[49] “CLIPSeg.” [Online]. Available: https://huggingface.co/docs/transformers/en/model_doc/clipseg

[50] “Meta Segment Anything Model 2.” [Online]. Available: https://ai.meta.com/sam2

[51] K. Ikeuchi, Ed., *Computer Vision: A Reference Guide*. Boston, MA: Springer US, 2014. [Online]. Available: http://link.springer.com/10.1007/978-0-387-31439-6

[52] N. Ketkar, “Introduction to Keras,” in *Deep Learning with Python*. Apress, Berkeley, CA, 2017, pp. 97–111. [Online]. Available: https://link.springer.com/chapter/10.1007/978-1-4842-2766-4_7

Appendix A
Mission objectives and system requirements tables

This appendix outlines the mission objectives and key requirements for the deep learning-based obstacle avoidance system for planetary rovers, categorized in functional requirements, performance requirements, design requiremetns, interface requirements and validation requirements. While many additional requirements would be necessary for an actual mission, these represent the core requirements that guided the research and were directly addressed in this thesis. Each requirement contains a priority level and a verification method following ECSS standards (T-test, A-analysis, I-inspection, or R-review) [44].

ID	Mission Objective	Verification	Criticality	Rationale
MO-01	The system shall enable autonomous obstacle detection for rover navigation by identifying rocks that should be avoided on the planned path.	R	Critical	Core mission objective ensuring safe rover navigation through obstacle detection.
MO-02	The system shall prevent rover damage from collisions with non-traversable obstacles through a rock detection of 90% in each analysed frame.	R	Critical	Prevents mission failure by protecting rover hardware from collision damage.
MO-03	The system shall employ deep learning techniques for rock segmentation, achieving a minimum intersection over union of 0.70.	R	Critical	Justifies the research approach and addresses limitations of current geometric detection methods.
MO-04	The system shall operate within the computational constraints of lightweight planetary rovers.	R	Critical	Ensures practical deployment feasibility on resource-constrained rover platforms.
MO-05	The system shall demonstrate end-to-end integration and validation on a rover platform in Mars-analogue conditions.	R	Critical	Validates real-world applicability beyond simulation-only testing.

TABLE XVI: Mission Objectives for the Deep Learning Obstacle Detection System

ID	Requirement	Verification	Criticality	Rationale (Traceability)
Functional Requirements				
FR-01	The system shall detect and segment rocks larger than 12 cm in height per frame as obstacles to avoid.	T	Critical	Rocks of this size cannot be safely traversed by the target rover platform (MO-01, MO-02).
FR-02	The system shall process stereo camera images to generate pixel-wise rock segmentation masks.	T	Critical	Core deep learning functionality for obstacle identification (MO-03).
FR-03	The system shall detect obstacles within a forward range of 4 meters and 60° horizontal field of view.	T	High	Provides validated operational range for safe navigation and path planning (MO-01, MO-02).
FR-04	The system shall segment 2 categories: 'rocks-to-avoid' and 'other'.	T	Medium	Binary classification capturing the essential information to avoid rocks (MO-03).
FR-05	The system shall generate a local occupancy grid with spatial resolution of 5 cm within the detection range.	T	High	Provides sufficient spatial detail for path planning while being computationally efficient (MO-01).
FR-06	The system shall maintain an internal map of detected obstacles within the required view ahead.	T	Medium	Allows for planning with recently detected obstacles that may have been missed in the new segmentation mask (MO-01).
FR-07	The system shall include procedures for converting trained models to formats compatible with rover hardware.	T, I	High	Ensures trained models can be effectively deployed on the target rover hardware (MO-04).
Performance Requirements				
PR-01	The system shall achieve maximum inference time of 0.05 seconds per image frame on target hardware.	T	Critical	Ensures real-time operation capability for responsive obstacle detection (MO-04).
PR-02	The system shall capture and process images at a minimum rate of 1 Hz.	T	Critical	Minimum processing frequency to ensure safe rock detection at the rover's operational speed (MO-01, MO-02).
PR-03	The system shall achieve an Intersection over Union (IoU) score of at least 0.70 and a Dice coefficient of at least 0.80 for rock segmentation.	T	Critical	Quantitative accuracy thresholds demonstrating superior performance (MO-03).

PR-04	The system shall maintain detection rate above 90% for rocks larger than 12 cm at distances up to 4 meters under sunlight conditions.	T	Critical	Operational safety requirement ensuring reliable obstacle detection within validated range (MO-01, MO-02).
PR-05	The system shall maintain rock detection performance with maximum 15% degradation across varying lighting conditions.	T	High	Robustness requirement for real-world deployment in varying environmental conditions (MO-05).
PR-06	The deep learning model shall achieve maximum memory footprint of 100 MB during inference operations.	T	High	Ensures compatibility with limited memory resources on rover platforms (MO-04).
PR-07	The system shall maintain thermal stability with peak GPU temperatures below 50°C during continuous operation.	T	Medium	Validates thermal management compatibility for extended rover operations (MO-05).
Design Requirements				
DR-01	The compressed model shall not exceed 5 MB disk storage for deployment compatibility.	I	High	Ensures compatibility with limited onboard storage resources (MO-04).
DR-02	The system shall employ containerized deployment using Docker for reproducible integration.	I	Medium	Facilitates reliable deployment and integration across different platforms (MO-05).
Interface Requirements				
IR-01	The system shall integrate with the ROS2 navigation stack using standardised topic messages for data transfer.	T, I	High	Ensures compatibility with standard robotics navigation middleware (MO-05).
IR-02	The system shall accept rectified RGB images from the camera as input at resolution not bigger than 1920 x 1080 and output pixel-wise classification maps at 256 x 256 resolution.	T	Medium	Defines input/output interface specifications for integration (MO-03).
IR-03	The system shall employ a modular design architecture that allows component reuse and adaptation across different rover platforms.	R, I	Medium	Enables reuse and adaptation for different platforms with minimal modification (MO-05).
IR-04	The system shall integrate with existing rover hardware without requiring additional sensors or computers.	I	High	Ensures deployability on existing platforms without hardware modifications (MO-04, MO-05).
Validation Requirements				
VR-01	The system shall be validated using realistic Mars images.	T	Critical	Ensures performance validation in representative deployment conditions (MO-05).
VR-02	The system shall demonstrate end-to-end operation from camera input to occupancy grid output on rover platform.	T	Critical	Validates complete system integration rather than component-only testing (MO-05).
VR-03	The system shall be tested across diverse terrain types including rocky, sandy, and mixed surface conditions.	T	High	Ensures robustness across expected operational environments (MO-05).

TABLE XVII: System requirements for the deep learning obstacle detection system

Appendix B
Mission objective validation and requirement verification

This section provides a systematic validation of mission objectives and verification of the system requirements defined in section A, against the experimental results and system performance demonstrated throughout this research. The process follows the V-model development framework presented in Figure 3.

First, the high-level mission objectives are validated to demonstrate achievement of the research goals; second, the detailed system requirements are verified to ensure technical compliance and traceability from objectives to implementation. The verification status is categorized as follows: **PASS** indicates the requirement has been fully met with supporting evidence; **PARTIAL** indicates the requirement has been partially satisfied but with identified limitations or gaps; and **FAIL** would indicate non-compliance (no requirements fell into this category). Each verification entry includes evidence and references to the relevant sections, tables, or figures where detailed results can be found.

ID	Mission Objective	Validation Evidence	Status	Reference
MO-01	The system shall enable autonomous obstacle detection for rover navigation by identifying rocks that should be avoided on the planned path.	System successfully detects and maps rocks >12cm with 94% reliability, generates occupancy grids for navigation	**PASS**	Table XIII, section V
MO-02	The system shall prevent rover damage from collisions with non-traversable obstacles while driving.	Achieved a rock miss probability low enough to continue operations for multiple years, depending on environment characteristics	**PASS**	section VI
MO-03	The system shall employ deep learning techniques for rock segmentation, achieving a minimum intersection over union of 0.70.	Deep learning model achieved 0.770 IoU and 0.858 Dice coefficient, exceeding minimum threshold	**PASS**	Table IX
MO-04	The system shall operate within the computational constraints of lightweight planetary rovers.	Model uses only 0.31M parameters (1.2MB), 4.49ms inference time, 76MB memory footprint	**PASS**	Table XII, section VI
MO-05	The system shall demonstrate end-to-end integration and validation on a rover platform in Mars-analogue conditions.	Complete system tested on MaRTA rover in ESA Mars Yard with varying terrains and lighting conditions	**PASS**	section VI

TABLE XVIII: Mission objectives validation results showing achievement of high-level research goals

ID	Requirement	Verification Evidence	Status	Reference
Functional Requirements				
FR-01	The system shall detect and segment rocks larger than 12 cm in height per frame as obstacles to avoid.	94% detection rate for rocks >12cm under sunlight conditions	**PASS**	Table XIII
FR-02	The system shall process stereo camera images to generate pixel-wise rock segmentation masks.	Binary segmentation masks successfully generated with 77.0% IoU	**PASS**	Table IX, Figure 12
FR-03	The system shall detect obstacles within a forward range of 4 meters and 60° horizontal field of view.	Effective range validated to 4m, but beyond 3.5m: 86% position error	**PARTIAL**	Table XIV
FR-04	The system shall segment 2 categories: 'rocks-to-avoid' and 'other'.	Binary classification model implemented and tested	**PASS**	Figure 11
FR-05	The system shall generate a local occupancy grid with spatial resolution of 5 cm within the detection range.	5cm resolution occupancy grid implemented	**PASS**	section V
FR-06	The system shall maintain an internal map of detected obstacles within the required view ahead.	Persistence mechanism with 3-cycle threshold implemented but not validated	**PARTIAL**	section V
FR-07	The system shall include procedures for converting trained models to formats compatible with rover hardware.	Keras→ONNX→TensorRT pipeline verified with performance testing	**PASS**	Table XI, Table XII
Performance Requirements				
PR-01	The system shall achieve maximum inference time of 0.05 seconds per image frame on target hardware.	4.49ms average inference time achieved with TensorRT	**PASS**	Table XII
PR-02	The system shall capture and process images at a minimum rate of 1 Hz.	Camera captures at 1Hz and system demonstrates 222.5 fps process capability (far exceeds requirement)	**PASS**	Table XII
PR-03	The system shall achieve an Intersection over Union (IoU) score of at least 0.70 and a Dice coefficient of at least 0.80 for rock segmentation.	IoU: 0.770, Dice: 0.858 achieved on validation set	**PASS**	Table IX
PR-04	The system shall maintain detection rate above 90% for rocks larger than 12 cm at distances up to 4 meters under sunlight conditions.	94% detection rate achieved for large rocks under sunlight, but fails in lab lighting (82%)	**PARTIAL**	Table XIII

PR-05	The system shall maintain rock detection performance with maximum 15% degradation across varying lighting conditions.	Laboratory vs sunlight: 82% vs 94% detection (12% degradation)	**PASS**	Table XIII
PR-06	The system shall achieve maximum memory footprint of 100 MB during inference operations.	76MB peak memory usage with TensorRT FP16 optimization	**PASS**	section VI
PR-07	The system shall maintain thermal stability with peak GPU temperatures below 50°C during continuous operation.	Peak temperature: 49.2°C during continuous operation	**PASS**	section VI
Design Requirements				
DR-01	The compressed model shall not exceed 5 MB disk storage for deployment compatibility.	1.2MB TensorRT model size	**PASS**	section V
DR-02	The system shall employ containerized deployment using Docker for reproducible integration.	Docker container with L4T, ROS2, and MARTA stack implemented	**PASS**	section V
Interface Requirements				
IR-01	The system shall integrate with the ROS2 navigation stack using standardised topic messages for data transfer.	Standard occupancy grid messages implemented	**PASS**	section V
IR-02	The system shall accept rectified RGB images from the camera as input at resolution not bigger than 1920 x 1080 and output pixel-wise classification maps at 256 x 256 resolution.	Input preprocessing and 256x256 output resolution verified	**PASS**	section IV
IR-03	The system shall employ a modular design architecture that allows component reuse and adaptation across different rover platforms.	ROS2 node-based architecture enables cross-platform deployment	**PASS**	Figure 7
IR-04	The system shall integrate with existing rover hardware without requiring additional sensors or computers.	Successfully deployed on existing MaRTA Jetson AGX Orin platform using the Bumblebee camera	**PASS**	section V
Validation Requirements				
VR-01	The system shall be validated using realistic Mars-like images.	MarsTanYard dataset created in ESA Mars Yard, but limited to analogue environment	**PARTIAL**	section V, section VI
VR-02	The system shall demonstrate end-to-end operation from camera input to occupancy grid output on rover platform.	Complete pipeline from camera to occupancy grid tested	**PASS**	section VI
VR-03	The system shall be tested across diverse terrain types including rocky, sandy, and mixed surface conditions.	Rocky, sandy, pebbles, and mixed terrain tested in controlled lab environment.	**PASS**	section VI

TABLE XIX: Requirements verification results showing evidence and status for each system requirement