# Terrain Risk Classification for Autonomous Rovers on Mars

William Scott
California State University, Sacramento
Sacramento, CA, USA
wscott@csus.edu

Byron Saylor
California State University, Sacramento
Sacramento, CA, USA
byronsaylor@csus.edu

## Abstract

Autonomous rover navigation on Mars requires accurate terrain assessment to ensure mission safety and success. While existing approaches using the AI4Mars dataset focus on pixel-level terrain classification with computationally intensive models like DeepLabv3, they do not assess traversability risk or predict the safety of navigating different terrain compositions. This research develops a lightweight deep learning model for terrain risk classification that assigns discrete risk levels (Low, Medium, or High) to rover imagery based on the safety of the terrain. Unlike traditional segmentation methods that identify terrain types at the pixel level, our approach evaluates overall scene safety, providing rovers with actionable intuition about navigational hazards ahead. By prioritizing model efficiency, this work delivers practical traversability assessment while maintaining computational feasibility for resource-constrained onboard rover systems. The proposed lightweight model attains 82.6% classification accuracy, achieving a 87.6% reduction in model size relative to DeepLabV3, thereby confirming the feasibility of real-time, scene-level risk assessment for computationally limited planetary rovers.

## CCS Concepts

• **Computing methodologies** → **Deep learning**; **Computer vision**; • **Computer systems organization** → *Autonomous robots*; • **Applied computing** → *Space systems.*

## Keywords

Mars rover navigation, terrain risk classification, traversability assessment, lightweight neural networks, autonomous planetary exploration, AI4Mars dataset, deep learning, space robotics


*CSUS Fall 2025, Sacramento, CA, USA*


## 1 Introduction

### 1.1 Problem Description

Accurate perception and characterization of Martian terrain are critical for ensuring the safety and autonomy of rover navigation systems. Deep learning provides a data-driven framework for extracting visual and contextual cues from rover imagery, enabling systems to infer terrain properties without explicit human supervision. During traversal, autonomous rovers must possess an effective form of *terrain intuition*—the ability to perceive and anticipate the traversability and potential risk of the terrain directly ahead. This capability is essential not only for safe path planning but also for preventing wheel damage and mechanical wear, thereby extending the rover's operational lifespan and maximizing scientific data collection. However, Martian environments introduce significant challenges for such perception models, including extreme illumination variability, limited labeled data, and stringent onboard computational constraints. These factors necessitate the development of lightweight yet robust neural architectures capable of real-time terrain understanding and risk prediction under operational mission conditions.

### 1.2 Approach Overview

This project develops and evaluates a lightweight deep learning model for terrain risk classification. The model estimates discrete risk levels (Low, Medium, High) that reflect the relative ease or difficulty of navigating terrain based on rover imagery from the AI4Mars dataset.

### 1.3 Contributions

Our key contributions include:

- A lightweight neural network architecture for terrain risk classification suitable for resource-constrained rover systems
- A scene-level risk assessment approach that moves beyond pixel-level terrain segmentation
- Comprehensive evaluation demonstrating the trade-offs between model efficiency and classification accuracy
- A practical framework for real-time traversability decision-making in autonomous planetary exploration

## 1.4 Paper Organization

The remainder of this paper is organized as follows: Section 2 formally defines the problem, Section 3 describes our system architecture and algorithm design, Section 4 presents our experimental methodology and results, Section 5 reviews related work, and Section 6 concludes with lessons learned and future directions.

# 2 Problem Formulation

**Input:** RGB images of Martian terrain captured by rover cameras, where each image $I \in \mathbb{R}^{H \times W \times 3}$ represents a scene containing various terrain types (soil, bedrock, sand, rocks).

**Output:** A discrete risk classification $r \in \{Low, Medium, High\}$ representing the traversability risk of the terrain depicted in the input image.

**Objective:** Learn a mapping function $f : I \rightarrow r$ that accurately predicts terrain risk while maintaining computational efficiency suitable for deployment on resource-constrained rover hardware.

**Constraints:**

- Model must be lightweight (minimal parameters and computational requirements)
- Inference must be fast enough for real-time navigation decisions
- Must generalize to unseen Martian terrain variations

# 3 System and Algorithm Design

## 3.1 System Architecture

Our system performs automatic terrain-risk classification from raw rover images by combining a data-preprocessing pipeline, a label-generation module, and a lightweight deep neural network based on EfficientNetB0. A high-level conceptual flow is:

Raw Rover Image → Image Preprocessing Module (resize, normalize, augment) → Label Generation Module (convert pixel masks → risk labels) → Neural Network (EfficientNetB0 + custom head) → Predicted Risk Class (Low / Medium / High)

The model operates on 224×224×3 RGB images, processes them through the EfficientNetB0 feature extractor, and outputs class probabilities indicating the dominant terrain-risk category.

## 3.2 Data Preprocessing

The AI4Mars dataset was preprocessed through a structured pipeline to ensure consistent image-mask pairing and risk-label generation. The following steps summarize the procedure:

(1) **Dataset Organization**
Raw image and mask data were obtained from the AI4Mars MER and MSL subsets (Spirit, Opportunity, and Curiosity Rovers). Directory paths were standardized for training and testing partitions: *MER* images were drawn from `images/train` and `images/test`, while corresponding masks were located under `labels/train` and `labels/test`. A similar structure was adopted for *MSL* Navcam imagery and annotations.

(2) **File Validation and Matching**
All image and mask filenames were filtered by extension (`.jpg`, `.jpeg`, `.png`). Each mask was matched to its corresponding image by removing dataset-specific suffixes (e.g., `_merged`, `_T0_merged`) using a regular-expression rule. Pairs lacking a valid match were excluded and logged for verification. This ensured one-to-one alignment between imagery and annotations across all sources.

(3) **Terrain Label Extraction**
Each mask contained pixel-level labels corresponding to terrain classes (*soil*, *bedrock*, *sand*, *big_rock*, *null*). Masks were resized to 224×224 px using nearest-neighbor interpolation to preserve discrete class boundaries. Pixel counts were computed for each terrain type and used to estimate the overall composition of each scene.

(4) **Risk Classification Mapping**
A rule-based heuristic was applied to assign each scene a discrete risk category:
   - **Low risk:** soil > 65%, sand < 10%, and rock < 15%.
   - **High risk:** sand > 40% or rock > 35%, or (sand + rock) > 50%.
   - **Medium risk:** all other cases.

   The resulting risk label was one-hot encoded into three classes (*low*, *medium*, *high*) for model training.

(5) **TensorFlow Dataset Construction**
File paths, rather than full images, were loaded into memory to minimize GPU memory overhead. Each image–mask pair was processed through a two-stage TensorFlow pipeline:
   (a) Mask parsing and risk computation (executed on CPU).
   (b) Image decoding and resizing to 224 × 224 px (executed on GPU).

   The pixel values were retained in the original 0–255 range and cast to `float32` for model input. The resulting tensors were batched, shuffled, and prefetched for efficient training.

(6) **Train–Validation–Test Partitioning**
Data were divided into training, validation, and expert-labeled test subsets. The expert test set consisted exclusively of high-agreement (*gold*) masks from both MER and MSL datasets. A total of 24,893 image–mask pairs were used, with 20,711 samples for training (83.2%), 3,656 for validation (14.7%), and 526 expert-labeled pairs for testing (2.1%). The split was generated using a fixed random seed.

(7) **Quality Verification**
Random samples from the pipeline were visualized to confirm correct image–mask alignment, class-color integrity, and accurate risk computation. Visual inspection confirmed preprocessing consistency across rover sources.

## 3.3 Neural Network Architecture

*3.3.1 Model Design* A lightweight convolutional neural network (CNN) was developed using transfer learning with a pre-trained

*EfficientNetB0* backbone and a custom classification head tailored for three terrain-risk categories.

**Backbone:** The model uses *EfficientNetB0*, initialized with ImageNet weights and configured as a fixed feature extractor by removing its fully connected top layers (`include_top=False`) and applying global average pooling to the final convolutional output. EfficientNetB0 employs mobile inverted bottleneck convolution (MBConv) blocks and compound scaling to balance network depth, width, and input resolution. The input shape is $224 \times 224 \times 3$, consistent with the standard Keras configuration.

**Custom classification head:**

- Dense layer with 256 neurons and `ReLU` activation, learning task-specific representations from extracted features.
- Dropout layer with a rate of 0.6 for regularization and mitigation of overfitting.
- Output layer: Dense layer with 3 neurons and `softmax` activation, producing normalized class probabilities corresponding to low-, medium-, and high-risk terrain categories.

$$
\begin{aligned}
&\text{Input } (224 \times 224 \times 3) \\
&\quad \rightarrow \text{EfficientNetB0 (pre-trained, no top)} \\
&\quad \rightarrow \text{Global Average Pooling} \\
&\quad \rightarrow \text{Dense}(256, \text{ReLU}) \\
&\quad \rightarrow \text{Dropout}(0.6) \\
&\quad \rightarrow \text{Dense}(3, \text{Softmax}) \\
&\quad \rightarrow \text{Output (3-class probabilities)}
\end{aligned}
$$

*3.3.2 Training Procedure* **Loss Function:** Categorical cross-entropy was used, as the task involves multi-class classification with mutually exclusive labels.

**Optimizer:** The Adam optimizer was selected for its adaptive learning rate and suitability for fine-tuning deep networks.

**Training Phases:** Training was conducted in two stages:

(a) *Feature extraction phase:* The EfficientNetB0 backbone was frozen, and only the custom classification head was trained.
(b) *Fine-tuning phase:* Selected layers of the EfficientNetB0 backbone were unfrozen and trained with a reduced learning rate to refine high-level features for the terrain classification task.

**Hyperparameters:** The model was trained with a batch size of 32. The feature extraction phase was run for 50 epochs, followed by 100 epochs of fine-tuning with early stopping to prevent overfitting.

**Regularization and Callbacks:** A dropout rate of 0.6 was used in the classification head. Early stopping and model checkpointing (`ModelCheckpoint`) were applied based on validation accuracy to retain the best-performing weights.

**Data Pipeline and Preprocessing:** All input images were resized to $224 \times 224$ pixels and processed using the `tf.keras.applications.efficientnet.preprocess_input` function, ensuring pixel scaling consistent with the EfficientNet normalization scheme.

# 4 Experimental Evaluation

## 4.1 Methodology

**Dataset:**
We use the AI4Mars dataset, which contains images of Martian terrain collected by the MER (Spirit/Opportunity) and MSL (Curiosity) rovers. Our dataset is constructed by matching raw rover images with their corresponding pixel-level terrain labels, using the directory structure:

- `mer/images/train`, `mer/images/test`
- `mer/labels/train/merged-unmasked`,
- `mer/labels/test/masked-gold-min1-100agree`
- analogous directories for `msl`

A custom filename-matching pipeline aligns each image with its mask, producing a final set of valid (image, mask) pairs used to derive the 3-class terrain-risk labels. The dataset used in our experiments contains a total of **24,893** samples.

**Data Split:**
The matched dataset is divided into 83.2% training, 14.7% validation, and 2.1% testing, with no file overlap between splits.

**Experimental Setting:**
All experiments are performed using TensorFlow/Keras, with models defined using the EfficientNet API (`tf.keras.applications.efficientnet`). Training is executed on a standard GPU-accelerated environment (e.g., NVIDIA GPU) with batch processing using TensorFlow `tf.data` pipelines. Images are resized to $224 \times 224 \times 3$. Training is conducted in two phases:

- 50 epochs (feature extraction)
- 100 epochs (fine-tuning)

**Evaluation Metrics:**
We evaluate model performance using:

- Classification accuracy
- Precision, recall, and F1-score per terrain-risk class
- Confusion matrix analysis to identify class confusion patterns
- Model size (number of trainable parameters of EfficientNetB0 + custom head)
- Inference time, measured as the forward-pass latency for a single $224 \times 224$ input image

**Baseline Methods:**
We compare our lightweight EfficientNet-based classifier against:

- DeepLabv3, adapted for patch-level classification
- ResNet-based classifier

- Other conventional CNN baselines included during experimentation

## 4.2 Results

Table 1 summarizes the classification performance of both models evaluated on the expert-labeled test set. The baseline custom CNN achieved an overall accuracy of 80.6%, while the transfer learning model using EfficientNetB0 achieved a slightly higher accuracy of 82.5%. This indicates that the pre-trained EfficientNet backbone provided slightly improved generalization, with a large increase in model size.

**Table 1: Comparison of model performance on the expert-labeled test set**

| Model | Accuracy (%) | Model Size (MB) |
|---|---|---|
| Custom CNN | 80.6 | 3.6 |
| EfficientNetB0 | 82.5 | 29.0 |

**Table 2: Per-class Precision, Recall, and F1-score for Custom CNN and EfficientNetB0 models**

| Risk Category | Metric | Custom CNN | EfficientNetB0 |
|---|---|---|---|
| Low | Precision | 0.81 | 0.83 |
| | Recall | 0.93 | 0.86 |
| | F1-score | 0.87 | 0.85 |
| Medium | Precision | 0.75 | 0.76 |
| | Recall | 0.77 | 0.85 |
| | F1-score | 0.76 | 0.81 |
| High | Precision | 0.89 | 0.90 |
| | Recall | 0.66 | 0.73 |
| | F1-score | 0.76 | 0.81 |

Figure 1 presents the confusion matrices for both models. The EfficientNetB0 model shows a noticeable reduction in misclassification between the *Medium* and *High*-risk classes, suggesting better separation of complex terrain patterns. Both models correctly identified *Low-risk* terrain with high precision, while most errors occurred between visually similar *Medium* and *High*-risk samples.

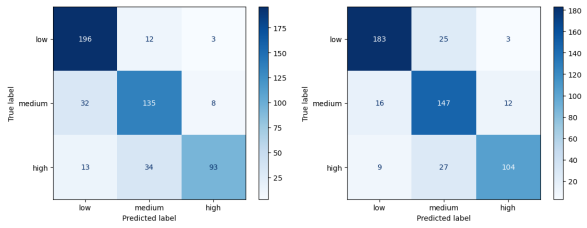


**Figure 1: Confusion matrices for the baseline CNN (left) and EfficientNetB0 (right) models.**

Overall, the results demonstrate that transfer learning with EfficientNetB0 provides a modest improvement in generalization to unseen Martian terrain, yielding higher accuracy in distinguishing *Medium* and *High*-risk classes. Both models confirm the feasibility of real-time terrain risk assessment on resource-constrained rover platforms, maintaining small model sizes suitable for onboard deployment.

# 5 Related Work

## 5.1 Terrain Segmentation on Mars

Recent work on Martian terrain understanding has been driven largely by the AI4MARS dataset, a large-scale collection of ~35,000 rover images paired with over 326,000 pixel-level semantic segmentation labels obtained through crowdsourcing and expert annotation [8, 9]. This dataset has enabled a series of studies that frame Mars terrain perception primarily as a supervised semantic segmentation problem, typically using encoder–decoder architectures such as DeepLabV3/V3+, PSPNet, or more recent lightweight and transformer-based variants [9]. These models predict per-pixel terrain classes (e.g., soil, sand, bedrock, big rock) and are evaluated using metrics such as mean IoU and pixel accuracy. Follow-up work has focused on improving segmentation efficiency, adapting terrestrial state-of-the-art models to rover imagery, and exploring semi- and self-supervised approaches to reduce reliance on expensive pixel annotations [9].

Our project diverges from these prior formulations by shifting the problem from pixel-wise classification to terrain *risk* classification. Instead of predicting discrete terrain labels, we estimate a discrete risk level relative to the difficulty of traversing the imaged terrain. This formulation aligns more directly with the needs of onboard planning and navigation systems, which consume continuous cost signals rather than categorical labels [2]. While existing segmentation models require additional mapping from classes to planner-specific costs, our approach produces planner-ready outputs that more naturally support ranking, thresholding, and uncertainty-aware decision making.

Methodologically, our model emphasizes computational efficiency by using a lightweight convolutional backbone with a compact regression head, in contrast to the heavy segmentation architectures commonly used in prior work [9]. This design choice addresses practical rover constraints on compute, memory, and power, which have motivated recent research into efficient Mars segmentation architectures [9]. To leverage the extensive AI4MARS labels without requiring new expert annotations, we compute traversability scores from the original mask labels using an area-weighted scheme over terrain classes, enabling continuous supervision derived from existing pixel annotations [8, 9].

This problem formulation is valuable because it (1) aligns perception outputs directly with the requirements of autonomous navigation [2], (2) reduces the computational burden relative to full semantic segmentation [9], (3) increases robustness by avoiding brittle class-boundary errors, and (4) leverages the strengths of existing labeled

datasets while addressing downstream autonomy needs that pixel-level methods do not directly capture [9].

## 5.2 Traversability Analysis for Rovers

Classical rover traversability research models mobility as a continuous function of terrain parameters such as soil cohesion, slip ratio, sinkage, wheel loading, and slope angle. These physics-based approaches estimate traversability through quantities like predicted slip or maximum climbable slope, often derived from terramechanics and empirical rover mobility testing [1,2]. In contrast, our system reformulates rover mobility assessment as a discrete *risk classification* problem, assigning each terrain patch to one of three categories: Low, Medium, or High risk. This supports a perceptual, image-driven evaluation pipeline suitable for real-time autonomous decision-making.

Extensive mobility studies from the Mars Exploration Rover and Mars Science Laboratory missions show that loose, sandy terrain produces the most severe slip-related hazards. During the well-documented Hidden Valley event, Curiosity experienced excessive slip on unconsolidated sand at an average slope of only four degrees [3]. Controlled mobility experiments further demonstrate a sharp transition in cohesionless sand, from moderate slip around 10° to wheel embedding at 17° [1]. In our taxonomy, these conditions correspond to the **High-risk** class, representing visually sandy, low-cohesion regions historically associated with immobilization events.

Bedrock terrain generally provides improved traction but introduces long-term mechanical hazards. Mobility testing indicates that Curiosity can climb slopes up to 22° on smooth bedrock [4], supporting a **Medium-risk** classification. However, operational analyses also note that repeated traversal over sharp or angular rocks fatigues the thin aluminum wheel skin, causing cracks and brittle fractures [5]. Physics-based traversability models characterize this hazard in terms of wheel stress and material fatigue, whereas our image-based classifier detects bedrock using structural and textural visual cues.

Large embedded rocks constitute the highest mechanical hazard, functioning as both puncture threats and physical obstructions. Rover operations reports emphasize the need to avoid sharp, point-like rocks to reduce wheel gouging and puncture likelihood [5], and wheel tread fractures observed on Curiosity have been directly attributed to interactions with embedded rocks [6]. Our system therefore assigns such rocks to the **High-risk** class, reflecting both obstruction risk and potential for immediate wheel damage.

Cohesive soils, by contrast, offer the most stable traversal conditions. Slope tests demonstrate that Curiosity can traverse cohesive soils at angles up to 28° [1], and geotechnical studies report friction angles of 34–39° for soil-like material at Mars Pathfinder sites [7]. These findings justify our **Low-risk** class, corresponding to visually uniform, fine-grained terrain with high bearing strength.

Overall, while classical traversability studies derive continuous mobility metrics from mechanical soil–wheel interactions, our framework translates these findings into discrete risk classes that can be predicted directly from image appearance. This allows rover mobility science to inform a lightweight CNN-based perception model capable of real-time, terrain-aware risk assessment.

# 6 Conclusion

This work demonstrates that lightweight deep learning models can provide effective terrain–risk classification for autonomous Mars rovers while remaining computationally feasible for deployment on resource-constrained hardware. By reframing terrain understanding as a scene-level risk prediction task rather than a pixel-wise segmentation problem, our approach produces planner-ready outputs that more directly support rover navigation and decision-making.

Our experiments show that both the custom CNN and the transfer-learning model based on EfficientNetB0 achieve strong performance on the expert-labeled test set, with accuracies of 80.0% and 82.5%, respectively. While EfficientNetB0 offers a modest improvement in generalization-particularly in reducing confusion between Medium and High-risk terrain—it does so with a larger model size. This highlights a clear trade-off: the custom CNN is more compact and efficient, whereas the transfer-learning model provides increased robustness to challenging terrain patterns. Notably, both models remain far lighter than traditional semantic segmentation architectures such as DeepLabv3, reinforcing the feasibility of real-time risk assessment under strict rover compute, memory, and power constraints.

These findings indicate that lightweight scene-level classifiers represent a practical direction for onboard traversability analysis. Their low computational footprint and simplified output structure reduce reliance on pixel-level segmentation and eliminate the need for downstream conversion from terrain classes to navigation costs. This can help avoid brittle boundary-level errors and streamline autonomous planning pipelines on future rover missions.

Future work may incorporate additional risk factors such as slope estimation, shading cues, or wheel–terrain interaction metrics to enhance prediction reliability. Integrating stereo depth or surface normal information could improve recognition of geometrically complex terrain. Long-term opportunities include continual learning in deployed environments, domain adaptation across rover platforms, and validation using simulated or real rover drives to assess robustness under mission-like conditions.

Overall, this study provides evidence that efficient, image-driven risk classifiers can serve as a practical bridge between perception and planning, helping advance the autonomy capabilities of next-generation planetary rovers.

# 7 Work Division

Project responsibilities were structured to ensure balanced contributions across data preparation, model development, and evaluation,

while allowing each team member to focus on specific technical components. Both members collaborated throughout all stages of the project, coordinating progress and integration through Discord.

### 7.1 Data Preparation and Preprocessing (Joint)

William Scott and Byron Saylor jointly managed the loading, inspection, and preprocessing of the AI4Mars dataset. Both team members implemented a TensorFlow-based streaming data pipeline (`tf.data`) to efficiently load and preprocess image–mask pairs directly from disk without exhausting GPU memory. The pipeline matched rover imagery with corresponding pixel-level terrain masks, applied resizing operations, and computed discrete risk labels (*Low*, *Medium*, *High*) from the terrain composition of each mask. This approach enabled on-the-fly decoding, batching, and prefetching, ensuring continuous GPU utilization during training.

### 7.2 Model Architecture and Development

William Scott took primary responsibility for designing the convolutional neural network (CNN) architecture using the Keras Sequential API. His work included selecting and implementing the *EfficientNetB0* backbone for transfer learning and designing the custom classification head for three risk categories. Byron Saylor focused on model optimization and engineering trade-offs, including evaluation of lightweight configurations, parameter efficiency, and inference speed. Both members jointly iterated on architecture variations, reviewed results, and refined the final design.

### 7.3 Training, Hyperparameter Tuning, and Validation (Joint)

Both team members shared responsibility for training and validating the models. Tasks included executing training runs, monitoring epochs, and conducting hyperparameter changes over learning rate, batch size, and dropout rate. Early stopping and checkpointing strategies were tested collaboratively to prevent overfitting and ensure reproducibility across training sessions.

### 7.4 Evaluation, Efficiency Testing, and Visualization

Byron Saylor led the evaluation of model performance, including computation of classification metrics such as accuracy, precision, recall, and F1-score, as well as inference latency and model size analysis. William Scott led the creation of visualizations, including image-mask pair verification and confusion matrices.

## 8 Learning Experience

This project began with the goal of developing a deep learning model that could be applied to a meaningful real-world dataset. One of the initial challenges was identifying a dataset that was both sufficiently large to enable effective training and distinct from topics previously explored by other professionals. In retrospect, projects related to space exploration inherently present difficulties due to the scarcity of publicly available data and limited prior research. Much of the existing work in this area is conducted by specialized teams at NASA and the Jet Propulsion Laboratory (JPL), where datasets are often highly specific, minimally documented, and focused on environments that remain unexplored by humans, such as the Martian surface.

Another major lesson we learned was the computational difficulty of training deep neural networks on personal hardware. Even lightweight architectures require significant processing power and time to achieve convergence. As students without access to high-performance GPUs, we observed that training times remained lengthy and that careful scheduling was needed to allow sufficient epochs for the model to stabilize. With additional computational resources and more time for hyperparameter tuning, it is likely that we could have achieved higher classification accuracy and overall model performance.

## References

[1] Arvidson, R. E., et al. (2013). *Traverse Performance Characterization for the Mars Science Laboratory Rover.* Journal of Field Robotics.
[2] Rankin, A., et al. (2022). *Field Mobility Work Instructions.* NASA Jet Propulsion Laboratory. NASA Jet Propulsion Laboratory Technical Report, 2022.
[3] ScienceDirect (2024). *Curiosity Rover – Hidden Valley Incident.*
[4] Lakdawalla, E. (2020). *Curiosity Wheel Damage: The Problem and Solutions.* The Planetary Society.
[5] Lakdawalla, E. (2020). Sections on wheel fatigue and sharp-rock punctures. *Curiosity Wheel Damage: The Problem and Solutions.*
[6] NASA/JPL (2017). *Breaks Observed in Rover Wheel Treads.*
[7] Peters, G. H., et al. (2006). *Mars Soil Mechanical Properties and Suitability of Mars Soil Simulants.* Journal of Aerospace Engineering.
[8] NASA Open Data Portal (2021). *AI4MARS: A Dataset for Terrain-Aware Autonomous Driving on Mars.* https://www.kaggle.com/datasets/yash92328/ai4mars-terrainaware-autonomous-driving-on-mars/data
[9] M. Ono et al. (2021). *AI4MARS: A Dataset for Terrain-Aware Autonomous Driving on Mars.* IEEE Robotics and Automation Letters.