Exploration of Large language model assisted boulder detection from Lidar data

Lingli Zhu, Emilia Hattula, Jere Raninen

National Land Survey of Finland, Vuorimiehentie 5, 02150 Espoo, Finland

Keywords: Large language model, Boulder detection, Lidar data, Claude 3.7 Sonnet, Gemini 2.5 Pro, OpenAI o1

Abstract

In recent years, large language models (LLMs) have revolutionized many aspects of life and work, and their impact is expected to continue transforming professional practices in the near future. Artificial intelligence is poised to become a standard tool in our workflows. This paper investigates the comprehension and reasoning capabilities of LLMs for boulder detection from high-density Lidar data (20 points/m²) and its derivatives, such as DEM, DSM, slope, and roughness, evaluating their potential to achieve reliable results. Three LLMs with notable reasoning and coding capabilities—Claude 3.7 Sonnet, Gemini 2.5 Pro, and OpenAI o1—were selected for this study. Due to the complexity of working and availability with very high-resolution data for boulder detection, few studies have explored this area. As a result, this research highlights the potential of LLMs in innovative applications and underscores their role in advancing collaborative research efforts to enhance scientific capabilities.

1. Introduction

In recent years, Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, including question answering, summarization, image and video generation, and the development of intelligent, code-based solutions (Li, 2024). This versatility has sparked growing interest in exploring their potential across diverse research domains. In remote sensing, LLMs have been applied to tasks such as image captioning, text-based image generation, text-based image retrieval (TBIR), visual question answering, scene classification, semantic segmentation, and object detection (Li et al., 2024). These developments highlight the expanding role of LLMs in enhancing remote sensing research and applications.

Beyond traditional multimodal tasks, LLMs also support prompt-based class discovery and exhibit strong performance in coding-related activities, including code generation, completion, and interpretation across multiple programming languages. A notable example is Claude 3.7 Sonnet, the latest iteration of Anthropic's LLM (Anthropic, 2025), which introduces a "hybrid reasoning" approach. This mechanism enables the model to switch between rapid responses and detailed, step-by-step reasoning, significantly improving its effectiveness in solving complex programming and logic problems.

Boulder detection is relevant across a range of disciplines and environmental settings, encompassing both planetary and terrestrial contexts. On Earth, boulders provide insights into geological processes such as glacial transport, landslides, and tsunami deposits. In planetary exploration, boulder detection plays a vital role in identifying safe landing sites, enabling autonomous navigation of rovers, and interpreting surface processes. Boulder distributions provide valuable clues about crater formation, erosion, mass-wasting events, and even seismic activity, such as moonquakes.

Boulder studies require data with very high spatial resolution. In the literature, high-resolution imagery with a ground sampling distance of approximately 25 cm/pixel has been widely used in planetary exploration (Rothrock et al., 2016; Palafox et al., 2017). For terrestrial applications, various high-resolution data sources have been employed, including side-scan sonar with resolutions of 0.1–0.2 m (Feldens et al., 2019), geophysical survey data such as electrical resistivity with resolutions ranging from 1 to 5 m (Su et al., 2021; Gomo et al., 2023), topo-bathymetric Lidar data with point densities between 5 and 20 points/m² (Hansen et al., 2021, 2022), and aerial orthophotos with 30 cm spatial resolution (Jiang et al., 2019).

Most image-based boulder detection studies are limited to two-dimensional information, unless stereo or multi-view imagery is available. These methods also perform best in open, vegetation-free areas. Detecting boulders in forested environments remains challenging when relying solely on optical imagery. In contrast, Lidar, an active remote sensing technology, has the advantage of being able to penetrate vegetation canopies and capture ground-level features. However, its effectiveness depends on the wavelength of the laser system used.

Topo-bathymetric Lidar typically operates at a wavelength of 532 nm, which is capable of penetrating clear water and is therefore well-suited for mapping riverbeds, coastal zones, and shallow aquatic environments. Due to power and atmospheric absorption limitations, 532 nm systems are typically used at lower altitudes. Topographic Lidar systems, on the other hand, commonly operate at 1064 nm. These systems support higher power outputs and longer ranges, making them suitable for large-scale surveys from airborne or satellite platforms. The 1064 nm wavelength provides strong returns from dry surfaces and offers good ground penetration in vegetated areas, making it ideal for mapping boulders, forest structures, buildings, roads, and terrain models.

As Lidar technology has advanced, point cloud densities have increased substantially—from approximately 0.5 points/m² in the early 2000s, to 5 points/m² in the late 2010s, and up to 20 points/m² or more in national-scale datasets by the 2020s. Despite this progress, research on boulder detection remains limited, largely due to the demanding spatial resolution and point density requirements.

In parallel, the emergence of Large Language Models (LLMs) in recent years has opened new avenues for data analysis and automation in remote sensing. Their potential applications in tasks such as code generation, data annotation, multimodal analysis, and prompt-driven class discovery are actively being explored by the research community.

This paper explores the novel application of large language models (LLMs) for boulder detection from Lidar data, addressing a gap in current research. In this study, two prompts were designed for the latest LLMs: Claude 3.7, Gemini 2.5, and ChatGPT 4o to test its intelligence. The first set of Prompts defined available datasets and the outputs. The algorithms are fully designed by the LLMs. In the second set of Prompts, we provided the detailed steps of the algorithm that we designed. By evaluating the potentials of different LLMs for this challenging task, it offers a fresh perspective on leveraging LLMs in geospatial analysis despite their reliance on existing knowledge.

.

2. Related work

The detection and classification of boulders have progressed from manual rock count methods to advanced machine learning pipelines incorporating semantic segmentation, object detection, and 3D geophysical modelling. This evolution reflects the increasing need for automated and scalable approaches across both planetary and terrestrial domains.

• Boulder Detection and Characterization on Planetary Surfaces
Early investigations into Martian surface characterization laid the groundwork for assessing landing site safety and understanding surface morphology. For instance, Golombek et al. (2008) analysed rock size-frequency distributions at the Phoenix landing site using analogue data from terrestrial environments, offering insights into Martian surface processes through comparative geology.
Subsequent research has increasingly leveraged automated methods. Rothrock et al. (2016) introduced SPOC, a deep learning-based terrain classifier capable of supporting real-time rover navigation on Mars. Palafox et al. (2017) applied convolutional neural networks (CNNs) to identify geological landforms, demonstrating the utility of deep learning in reducing manual annotation.
Recent advancements have further emphasized automation at scale. Hood et al. (2022) developed the Martian Boulder Automatic Recognition System (MBARS), integrating image processing and pattern recognition for large-scale boulder field extraction. Similarly, Zhu et al. (2021) adapted Earth-based object detection models such as YOLOv5—augmented with attention mechanisms—for boulder detection in planetary imagery, confirming the adaptability of these frameworks to extraterrestrial contexts.
The applicability of deep learning to high-resolution imagery has been further demonstrated by Prieur et al. (2023), who extracted boulders from satellite data, underscoring the feasibility of large-scale planetary surface analysis. In addition, studies by Nagle-McNaughton et al. (2019) and Bickel et al. (2019) exemplify hybrid approaches combining manual and automated techniques to investigate impact-driven boulder fields and rockfalls across Mars and the Moon.

• Earth-Based Analogues and Geophysical Methods
Terrestrial environments serve as essential analogues for planetary research. Feldens et al. (2019) employed neural networks to detect boulders in side-scan sonar imagery of marine settings, while Hansen et al. (2021) utilized topo-bathymetric Lidar to classify boulders in coastal areas.
In the context of civil engineering and infrastructure, boulder detection is critical for mitigating risks in tunnelling operations. Yang et al. (2024) highlighted the impact of undetected boulders during super-large diameter shield tunnelling, advocating for robust pre-construction geophysical surveys. Su et al. (2021) addressed similar challenges in urban subway construction by integrating 3D resistivity tomography with AI-driven data fusion techniques. Furthermore, Gomo et al. (2023) proposed a multi-sensor geophysical workflow tailored for boulder delineation in mining applications, demonstrating the value of integrated approaches for subsurface imaging.

• Machine Learning and Semantic Segmentation Techniques
Recent developments in deep learning have enabled precise pixel-level classification of boulders and rocks in complex terrains. Jiang et al. (2019) applied semantic segmentation to extract geological features from heterogeneous natural landscapes. Similarly, Maharaja et al. (n.d.) introduced a transfer learning framework for detecting boulders and craters, illustrating the benefits of leveraging pre-trained models for cross-domain generalization.
Ensemble learning methods have also been explored. Hansen et al. (2021, 2022) implemented a random forest classifier using Lidar-derived features for boulder classification in coastal environments, demonstrating robust performance in noisy, unstructured data.

• Terrain Classification and Autonomous Navigation
Beyond boulder detection, terrain classification is central to autonomous navigation in unstructured environments. Birk et al. (2008) provided a comprehensive review of terrain classification methods for planetary rovers and autonomous systems, highlighting the intersection of robotics, computer vision, and safety-critical operations. Building upon this foundation, Rothrock et al. (2016) applied deep learning in the SPOC framework to support real-time, autonomous terrain assessment—critical for planetary exploration.

3. Materials

The Lidar dataset was acquired in 2023 by the National Land Survey of Finland using a Leica TerrainMapper-2 system over the Heinävesi region, Finland. The survey was conducted at a flight altitude of 900 meters above mean ground level, with an airspeed of 130 knots. The laser scanner operated at a pulse repetition frequency of 1700 Hz, resulting in an average point density of 20 points/m². The maximum point spacing was 0.45 meters in the flight direction and 0.2 meters in the direction of the mirror movement. Other available data include Lidar derivatives: Lidar-DTM, Lidar-DSM, Lidar-Slope, and Lidar-Roughness. The Lidar data were classified into ground, low vegetation, and high vegetation using Terrascan software (Terrasolid, Finland). Two sets of test data (mapsheets: N5243G1_6 and N5233E3_5) from different environment were selected. Fig. 1 shows the example of 20 points/m² Lidar point cloud with undulating terrain and cliffs, predominantly covered by vegetation. Each set covers an area of 1 km². As shown in the right image of Fig. 1, stones are on top of undulating terrain, highlighting the complexity of the detection task in such environment.

.

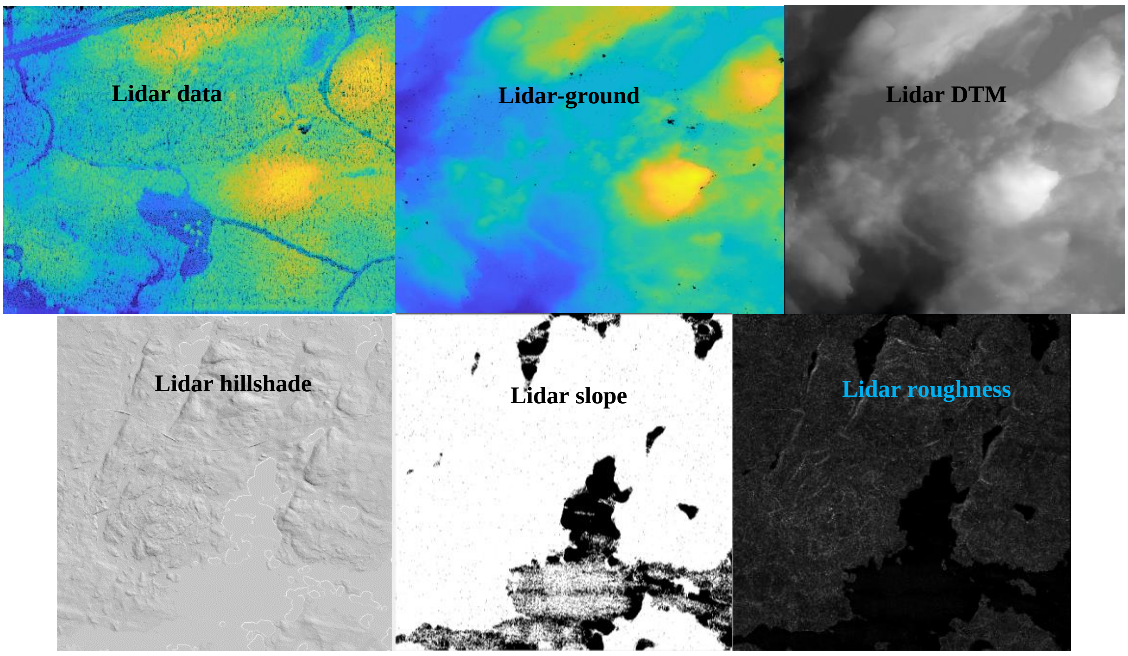

Figure 1. Example of test data.
Upper layer from mapsheet: N5243G1_6, lower layer from mapsheet: N5243G

Reference data were from Finnish National topographic database, in vector format with a shape file (Fig. 2).
The reference data was symbol-based indicator with evenly distributed boulder symbols. It demonstrated that there were boulders around the place. However, boulders can be identified visually from DEM data.

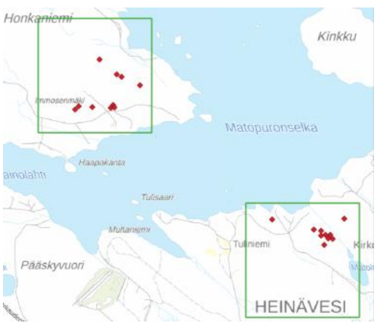

Figure 2. Boulder reference data from national topographic database.

3. LLMs

In the field of AI, a system's capacity for "reasoning" refers to more than just classification and prediction. It refers to its ability to analyse information, draw logical conclusions, incorporate context and nuance, and make informed decisions (Google Deep Mind). In this test, three recent published LLMs with powerful reasoning capacity were selected to explore their capacities for boulder detection. They are Claude 3.7 Sonnet, Gemini 2.5, and OpenAI o1 models.

3.1.1 Claude 3.7 Sonnet (Anthropic, USA)

Claude 3.7 Sonnet, released by Anthropic in February 2025, is an advanced hybrid reasoning AI model that integrates rapid response generation with extended, step-by-step reasoning capabilities. This flexibility allows users to tailor the model's performance to a wide range of tasks, from quick information retrieval to complex problem-solving scenarios (Anthropic, 2025).
Claude 3.7 Sonnet is state-of-the-art for agentic coding, and can complete tasks across the entire software development lifecycle—from initial planning to bug fixes, maintenance to large refactors. It offers strong performance in both planning and solving for complex coding tasks, making it an ideal choice to power end-to-end software development processes. In benchmarks, Claude 3.7 Sonnet has demonstrated improvements over Claude 3 Opus in coding-related tasks. For instance, it achieved a 96.2% score on the MATH benchmark, compared to Claude 3 Opus's 60.1%. Additionally, Claude 3.7 Sonnet has shown strong performance in software engineering tasks, outperforming models like GPT-4o and o1 in certain areas. It is accessible through various platforms (Anthropic, 2025).

3.1.2 Gemini 2.5 (Google, USA)

Google's Gemini 2.5, introduced in March 2025, represents a significant advancement in artificial intelligence, emphasizing enhanced reasoning and multimodal processing capabilities. Gemini 2.5 models are designed to reason through their thoughts before responding, resulting in improved performance and accuracy. The model can interpret various input types, including text, audio, images, video, and code, providing versatile and comprehensive responses. Gemini 2.5 Pro excels in generating visually compelling web applications and supports agentic programming, facilitating complex coding tasks with greater efficiency (Google DeepMind, 2025).

3.1.3 OpenAI o1 (OpenAI, USA)

OpenAI o1 is a reflective generative pre-trained transformer (GPT). A preview of o1 was released by OpenAI on September 12, 2024. o1 spends time "thinking" before it answers, making it

better at complex reasoning tasks, science and programming than GPT-4o (Metz, 2024). The full version was released to ChatGPT users on December 5, 2024. (OpenAI, 2025).
In tests, the o1 model has shown remarkable performance, surpassing human PhD-level accuracy on benchmarks in physics, chemistry, and biology. It also ranks in the 89th percentile on competitive programming questions. The o1 model is particularly effective for generating and debugging complex code efficiently. The o1 model has improved safety features, including better adherence to safety rules and guidelines1. It scored significantly higher on safety tests compared to previous models (OpenAI, 2025).

4. Methods

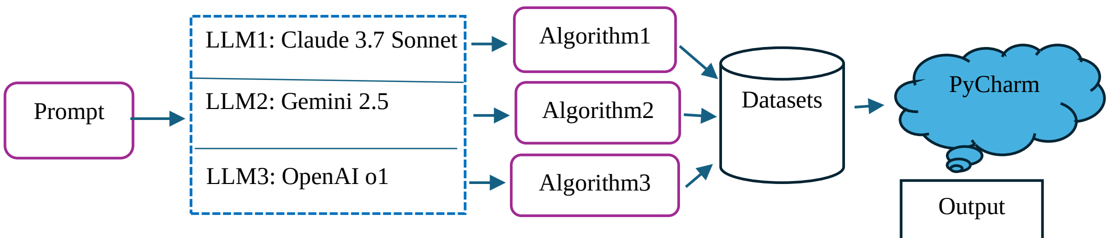

Figure 3. Workflow

In this study, a series of prompts were carefully developed to assess the performance of Large Language Models (LLMs) in the task of boulder detection. The initial set of prompts provided detailed information regarding the available datasets, their respective formats, and the expected output structure. Specifically, these prompts outlined the information to be included in the outputs and the required formats for the generated files. Following this, the LLMs were tasked with autonomously designing algorithms based on their inherent knowledge, subsequently generating Python code for the task.
The prompts were inputted into three distinct LLMs: Claude 3.7 Sonnet, Gemini 2.5, and OpenAI's GPT-4 (o1 model). The code produced by each model was then copied into the PyCharm Integrated Development Environment (IDE) and executed. The results of the execution were stored in shapefiles in a local directory for subsequent evaluation. The workflow, from input prompts to output generation, is illustrated in Fig.3.
This experiment aims to explore the capacity of LLMs to interpret dataset structures and apply existing knowledge to develop novel algorithms for the boulder detection task. To optimize the LLMs' performance, iterative testing and refinement of the prompts were necessary, ensuring that each prompt was as detailed and clear as possible. The insights and considerations drawn from our experience were helpful in shaping effective prompt design, highlighting the importance of precise and well-informed input for maximizing model performance. The following shows the final Prompt we designed:

Prompt: You act as a scientist to solve problems: you have a set of Lidar point cloud with point density of 20 points per square meter in .laz format. A laz file contains x, y, z, classification, intensity, number of echoes, and echo number for each point. Each point cloud file covers an area of one square kilometer. The number of points in one file is about 80M. The file size is between 50-500 MB. The point cloud has been classified into ground (in class 2), low vegetation (in class 3), and high vegetation (in class 4) using Terrascan software. Besides, other available data with 25cm spatial resolution include Lidar DEM, Lidar DSM, Lidar Slope, Lidar Roughness, and Lidar Hillshade, covering the same area as the Lidar point cloud. The above Lidar products were derived by TerraScan. The test area covers lots of vegetation. It means that some potential boulder points under vegetation might be in the last echoes. Besides, according to the articles (Hansen et al., 2021, 2022), curvature features might be useful. Please consider: "Local height differences in ground points might contain boulders, but some of them might be cliffs, not boulders. Lidar DSM might be useful for boulders in open areas, but they might not be useful in vegetation areas if boulders are under vegetation". Now you detect Boulders from above available data with each dimension greater than 0.5m. Can you provide an algorithm using Python to detect the Boulders with good accuracy and high efficiency? The result should be saved in '.shape' file. All data is in coordinate projection system: EPSG:3067. A figure containing Lidar ground points and detected boulders should be plotted.

In the figure, the detected boulders should be marked as a red circle.

5. Results and evaluation

In the experiment, datasets from two areas (N5243G1_6 and N5243G2_4) were tested with prompt for Claude 3.7 Sonnet, Gemini 2.5 Pro, and OpenAI o1 respectively. The algorithms were suggested by LLMs in Table 1.

In the Table, Claude 3.7 Sonnet utilized all datasets except Lidar hillshade. First the local height differences were estimated. Then by calculating the local normal to detect the height variation of the surfaces. The slopes and curvatures as important features were calculated to find potential boulders. DBSCAN was used for clustering the points to boulders. Gemini 2.5 pro was accessible in Google AI Studio. It employed all datasets except Lidar DSM and Lidar hillshade. First it calculated the curvatures from the Lidar DEM. Then it produced a smoothed DEM from the original Lidar DEM and estimated the height differences between them. The candidates points were extracted according to the multiple features: slope, curvature, roughness, and local height differences etc.. It also mentioned that roughness is optional because some boulder surfaces might have relative low roughness. Connected components analysis was used to group the candidates as boulder objects. The method considered the rejection of cliffs by using the aspect ratio to define the shape of object. OpenAI o1 model proposed a simple solution which only used the Lidar DEM. It ignored other datasets and suggestions given by the prompts. The idea from the algorithm was that a smoothed DEM was produced from the original DEM. The relative height differences were estimated. By analyzing the object shape using aspect ratio, the potential boulders were detected. By running the LLMs suggested python codes in PyCharm, these methods were tested. Table 2 showed the analysis of the solution given by the LLMs.

Gemini 2.5 + Prompt

Proposed Algorithm Strategy
The core idea is to use the high-resolution DEM and its derivatives (Slope, Roughness, Curvature) to find candidate boulder locations, then filter these candidates based on size, height, and shape characteristics. We will primarily work with the raster data for efficiency.
Algorithm Steps

1. **Setup:** Import libraries, define constants (paths, thresholds, CRS).
2. **Calculate Curvature:** Compute a curvature raster from the DEM. A simple approach is using the Laplacian, or more sophisticated methods involving Gaussian derivatives (Hessian matrix). A positive curvature indicates convex features.
3. **Calculate Relative Height (Detrended DEM):** Create a raster representing how much higher each pixel is compared to its local surroundings. This can be done by subtracting a smoothed version of the DEM (e.g., Gaussian blur or mean filter) from the original DEM. Boulders will appear as positive "bumps" in this relative height raster.
4. **Identify Candidate Pixels:** Threshold the derived rasters to find pixels likely belonging to boulders:
    - Relative Height > min_relative_height (e.g., 0.4m - slightly less than 0.5m to account for smoothing)
    - Curvature > min_curvature (positive value indicating convexity)
    - Slope > min_slope_on_boulder (e.g., 15-20 degrees - boulders aren't flat)
    - Roughness > min_roughness (optional, can help but might filter out smoother boulders) Combine these conditions using logical AND.
5. **Segment Candidates:** Group the contiguous candidate pixels into individual objects using connected components analysis (e.g., scipy.ndimage.label).
6. **Filter Objects:** Analyze each segmented object (potential boulder) based on geometric criteria:
    - **Minimum Height:** Calculate the actual height difference within the object's footprint on the *original* DEM (max(DEM) - min(DEM) within the object). This must be > 0.5m.
    - **Minimum Size/Dimensions:** Calculate the bounding box or area of the object. Ensure its dimensions are plausible for a boulder > 0.5m (e.g., area > ~0.5 * 0.5 m², considering pixel size; check bounding box dimensions). A minimum pixel count per object is also essential (e.g., at least 3-4 pixels for a 0.5m object at 25cm resolution).
    - **Maximum Size/Shape (Cliff Rejection):** Filter out very large or elongated objects that are more likely cliffs or other terrain features. Calculate the object's aspect ratio or major/minor axis lengths. Reject objects longer than a certain threshold (e.g., 10-20 meters).
    - **Context Check (Optional):** Check the average slope *around* the object's boundary. If the surrounding slope remains high, it might be part of a larger cliff face.
7. **Extract Boulder Locations:** For each valid boulder object, determine a representative point (e.g., the centroid or the pixel with the maximum relative height within the object).
8. **Create Shapefile:** Generate a GeoDataFrame (using GeoPandas) containing the points representing the detected boulders. Include attributes like estimated height or size if desired. Save as a shapefile with EPSG:3067.
9. **Visualization:**
    - Load a *sample* of the ground points (Class 2) from the LAZ file to avoid memory issues.
    - Load the detected boulder locations from the shapefile.
    - Create a 2D plot showing the ground points and overlaying the detected boulders as red circles.

OpenAI o1 + Prompt
Explanation:

1. **Loading the DEM:** The DEM (with a 25 cm resolution) is loaded using rasterio. In our case, it represents the ground surface as derived by TerraScan.
2. **Baseline and Height Difference:** A morphological opening (using a window of 9×9 pixels by default) is applied to the DEM. This “opens” the surface—smoothing out small structures. Subtracting this smoothed surface from the original DEM results in a map of local height differences (which acts as a proxy for curvature and protrusions, as suggested by Hansen et al.).
3. **Thresholding and Labeling:** A simple threshold (0.2 m) isolates the “bumps” that could be boulders. We then apply connected-component labeling to group neighboring candidate pixels. Each group (or region) is later filtered to ensure that both its width and height (converted from pixels to meters) are at least 0.5 m.
4. **Filtering by Shape:** To avoid selecting elongated features (like sections of a cliff), we compute an aspect ratio for each candidate; regions with a high aspect ratio (beyond a chosen threshold) are discarded.
5. **Saving and Plotting:** The centroids of the remaining candidate regions are saved into a shapefile (with EPSG:3067) using GeoPandas. Finally, a plot is produced: the DEM is shown as a “terrain” background and the detected boulders are overlaid as red circles.

Claude 3.7 Sonnet + prompt

Algorithm Explanation
The boulder detection algorithm I've developed uses a multi-feature approach that integrates several data sources to improve detection accuracy while maintaining computational efficiency:

1. **Data Loading & Preprocessing**
   - Processes LAZ point cloud data to extract ground points (class 2)
   - Loads complementary raster data (DEM, DSM, slope, roughness)
   - Computes derived features like height differences and curvature
2. **Candidate Point Selection**
   - Identifies initial boulder candidates using local height differences
   - Uses spatial indexing (KDTree) for efficient neighborhood searches
   - Processes data in chunks to maintain memory efficiency with large files
3. **Feature-Based Classification**
   - Calculates local normals to identify surface orientation
   - Measures local height variations, slopes, and curvatures
   - Differentiates boulders from cliffs using multiple criteria:
   - Height differences > 0.5m but < 5.0m (to avoid cliffs)
   - High local curvature (characteristic of boulder surfaces)
   - Moderate slopes < 65° (to avoid cliff faces)
4. **Boulder Delineation**
   - Uses DBSCAN clustering to group nearby points into individual boulders
   - Calculates boulder properties (height, width, length)
   - Filters by minimum size requirement (>0.5m in each dimension)
5. **Output & Visualization**
   - Saves results as shapefile with boulder properties in EPSG:3067 projection
   - Creates visualization with ground points and detected boulders as red circles

Table 1. The algorithms produced by the LLMs

LLMs / Items of analysis	Claude 3.7 Sonnet	Gemini 2.5 Pro	OpenAI o1
Bugs or error messages	No	First try: Out of memory Second try: run time 10h+	No
Use of provided data	All data except Lidar-hillshade	All data except Lidar-DSM and Lidar-hillshade	Only Lidar DEM
Used Features	local height differences, local normal, high variation, slopes, curvatures, DBSCAN	curvature, DEM smoothing, relative height, slope, roughness, segment candidates, local height differences, connected components analysis, aspect ratio	DEM smoothing, local height differences, aspect ratio
Understanding of datasets	Very good	Good	Fine
Efficiency	Relatively good	Not considered	Very fast

Table 2. Analysis of LLMs' suggested algorithms

From the results, Claude 3.7 Sonnet performed the best. There were no bugs in the code. It went smoothly. The runtime was a little long, about five hours for data processing, including 78M Lidar points. The code from Gemini 2.5 Pro produced bugs. For the first run, it went 'out of memory'. Then it suggested a solution to handle the bugs. Other problem came out. After a few round improvements by Gemini, finally it worked, but the runtime was rather long, over ten hours. From the OpenAI o1 model, the runtime was less than one minute. But the result was poor. Only one boulder was detected in one dataset.
The advantages of utilizing the LLMs lie in the possibilities of continuous improvement by iterating chat with them. The LLMs work like team members. Various solutions can be tested, and problems can be solved with their help. Fig. 4 shows the result from Claude 3.7 Sonnet model. It can be seen that the boulders (red dots) were well detected when overlapping with the LidarDEM. The reference data from the national topographic database were symbols evenly distributed in the area when the boulders presented. It doesn't represent the real location of the boulders. However, from the visual inspection, the potential of LLMs is evidenced.

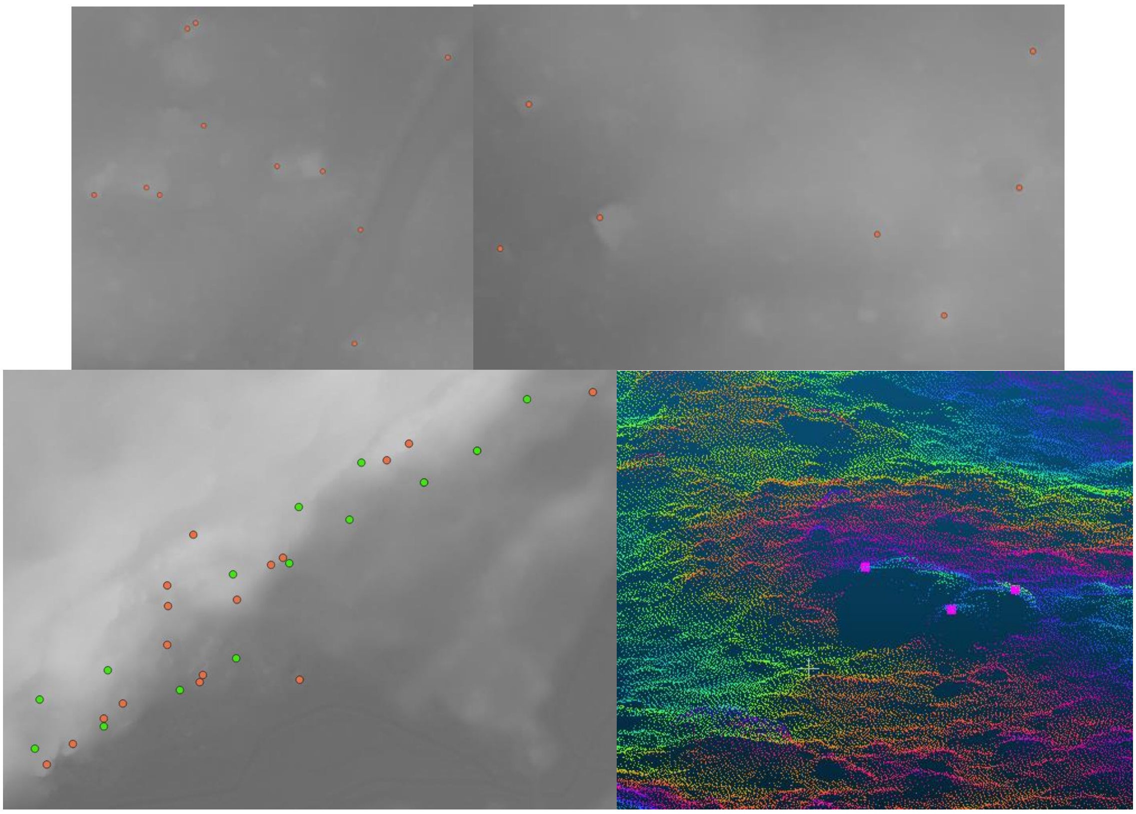

Figure 4. The results from Claude 3.7 Sonnet.
The red and magenta dots are the boulders detected by the LLMs. The green dots were the reference data with evenly distributed boulder symbols. The gray images are the DEMs. The color image shows the Lidar ground points. The color was determined by the heights of the terrain.

6. Discussion

In this experiment, we explored three LLMs with their capacities of the utilization of knowledge, scenario understanding, and reasoning regarding boulder detection. There was limited information from literature about boulder detection from the earth surfaces, especially regarding the use of Lidar data due to the demanding in high point density. In those three LLMs, each of them produced algorithm from different perspectives and had ‘own thinking’. Claude 3.7 Sonnet model tried to utilize all given information in Lidar data and their products (slope, roughness, and hillshade). Table 3 addressed the insights of the Claude 3.7 Sonnet model: what have been considered and what can be improved further. Similarly, Gemini 2.5 Pro model exploited almost all data except LidarDSM. Table 4 presented the explanation and key consideration from Gemini 2.5 Pro model. OpenAI o1 has a strong personality that it only picked Lidar DEM as data for further analysis. The notes from OpenAI o1 model can be found from Table 5. In this study, the reference data was imperfect. In the future, a field survey is needed to collect reliable reference data.

Scientific Considerations: Claude 3.7 Sonnet
The algorithm addresses specific challenges mentioned in your requirements:
1. Vegetation cover: Leverages last echo information to detect boulders under vegetation
2. Boulder vs. cliff differentiation: Uses multiple features (height, slope, curvature) to distinguish between these features
3. Size filtering: Ensures detected boulders meet the >0.5m dimension requirement
4. Computational efficiency: Employs spatial indexing and chunk processing for handling large datasets
To improve results further, we could:
• Adjust the parameters (curvature_threshold, boulder_cluster_eps, etc.) based on your specific terrain
• Add validation using a sample of known boulder locations

Table 3. Consideration of Claude 3.7 Sonnet

Explanation and Key Considerations: Gemini 2.5 Pro

1. **Dependencies:** You'll need numpy, rasterio, scipy, scikit-image, geopandas, shapely, matplotlib, and laspy (with lazrs support: pip install laspy[lazrs]).
2. **Parameter Tuning:** The Parameters section is **critical**. MIN_RELATIVE_HEIGHT, SMOOTHING_SIGMA, MIN_SLOPE, MIN_CURVATURE, MIN_PIXEL_COUNT, and MAX_BOULDER_AXIS will significantly impact results. You *must* experiment with these values based on visual inspection of the results and the characteristics of boulders in your specific area. Start with the suggested values and adjust.
3. **Curvature Calculation:** The provided calculate_laplacian_curvature is simple. More robust methods exist (e.g., using skimage.feature.hessian_matrix_eigvals or specific geomorphometry libraries), which might give better results but add complexity.
4. **Relative Height:** This is a powerful feature. The SMOOTHING_SIGMA determines the scale of the local neighborhood. A larger sigma finds larger bumps relative to a broader area.
5. **NoData Handling:** The code includes basic NoData handling using np.nan. Ensure your input rasters use a consistent NoData value or NaN. More sophisticated interpolation might be needed if NoData gaps are problematic.
6. **Object Filtering:** The filtering steps (height, size, shape) are crucial for removing false positives like noise, small vegetation artifacts misclassified as ground, and cliffs. The MAX_BOULDER_AXIS helps reject elongated cliff-like features.
7. **Efficiency:** The algorithm prioritizes raster operations, which are generally efficient. Loading the full point cloud is only done for visualization sampling. Processing a 1km x 1km 25cm DEM (4000x4000 pixels) should be feasible on a modern machine.
8. **Under-Canopy Boulders:** This algorithm primarily relies on the quality of the DEM generated from ground points. If TerraScan effectively used last returns to model the ground under vegetation, these boulders *should* be represented in the DEM and detectable. If the DEM is poor under dense canopy, a more complex points (especially last returns) within vegetated areas would be needed as a refinement step, significantly increasing complexity and processing time.
9. **Shapefile Output:** GeoPandas handles the creation of the shapefile with the correct CRS.
10. **Visualization:** Sampling the ground points is essential for performance. Adjust GROUND_POINT_SAMPLE_FRACTION based on your system's memory and desired plot density. The plot provides immediate visual feedback.
11. **Coordinate System:** The code assumes input rasters have CRS information and ensures the output shapefile uses the target CRS (EPSG:3067).

Table 4. Explanation from Gemini 2.5 Pro model

Keep in mind that one may also use the point cloud directly—for example by reading the ".laz" file with laspy and then applying spatial neighbourhood techniques—but because you already have high-resolution DEM, DSM, slope, and other products, using the DEM (representative of the ground surface) is an efficient approach.

Further Notes: OpenAI o1
Vegetated Areas: In areas where vegetation might hide boulders (e.g., if the boulder is visible only in the last echoes), one could incorporate the DSM or the point cloud's last echoes (after filtering by classification) to "peek" beneath vegetation.

Curvature and Slope Products: You could also load and integrate the Lidar slope or curvature rasters to help distinguish boulders from cliffs. For example, candidate regions with a very high local slope (from the slope raster) could be discarded as potential cliffs.

Efficiency: This algorithm is efficient because most of the heavy lifting is done by vectorized (raster) operations on the DEM. If working with extremely large datasets (e.g., multiple km²), consider tiling the data.

Table 5. Notes from OpenAI o1 model

7. Conclusion

This study examined the comprehension and reasoning abilities of large language models (LLMs) for boulder detection from 20p/m² Lidar data and its derivatives. Three LLMs—Claude 3.7 Sonnet, Gemini 2.5 Pro, and OpenAI o1—were selected for the experiment. A series of prompts, including detailed data descriptions, insights from human expertise, and clear task objectives, were designed to guide the models. Each LLM generated a set of algorithms and Python code for the task. Among them, Claude 3.7 Sonnet outperformed the others, demonstrating superior dataset understanding, strong reasoning capabilities, and careful attention to the prompt's content. Gemini 2.5 Pro also exhibited strong reasoning skills, though its algorithm failed to account for computing efficiency, resulting in an "out of memory" error. After adjustments, the runtime for

Gemini's algorithm on 1 km² of Lidar data was over 10 hours, compared to just 5 hours for Claude's solution. The OpenAI o1 model provided a simpler solution, but it overlooked critical data and ignored human-provided hints, relying solely on Lidar DEM for boulder detection. Consequently, its algorithm produced suboptimal results with a runtime of only one minute. Overall, Claude 3.7 Sonnet showed considerable potential for future research applications, proving capable of functioning as an effective tool to augment scientific research efforts.

Acknowledgements

The authors wish to thank CSC - IT Center for Science, Finland (urn:nbn:fi:research-infras-2016072531) and the Open Geospatial Information Infrastructure for Research (Geoportti, urn:nbn:fi:research-infras-2016072513) for computational resources and support.

References

Bickel, V., et al. (2019). "Impacts drive lunar rockfalls over billions of years." Nature Communications, 10(1), 1-7.

Birk, A., Stoyanov, T., Nevatia, Y., Ambrus, R., Poppinga, J., & Pathak, K. (2008). Terrain classification for autonomous robot mobility: from safety, security rescue robotics to planetary exploration. In IEEE International Conference on Robotics and Automation (ICRA), Planetary Rovers Workshop (pp. 1-5).

Feldens, P., Darr, A., Feldens, A., & Tauber, F. (2019). Detection of boulders in side scan sonar mosaics by a neural network. Geosciences, 9(4), 159.

Golombek, M., et al. (2008). "Size-frequency distributions of rocks on the northern plains of Mars with special reference to Phoenix landing surfaces." Journal of Geophysical Research: Planets, 113(E3).

Gomo, S., Rapetsoa, M., Manzi, M., Onyebueke, E., Dildar, J., Sihoyiya, M., Mutshafa, N., Harrison, W., Bourdeau, J., Brovko, O., James, I., Cooper, G., Scheiber, S., & Durrheim, R. (2023). Integrated geophysical methods for boulder delineation to improve mining. Geophysical Prospecting, 71, 1226 - 1246. https://doi.org/10.1111/1365-2478.13322.

Hansen, S. S., Ernstsen, V. B., Andersen, M. S., Al-Hamdani, Z., Baran, R., Niederwieser, M., ... & Kroon, A. (2021). Classification of boulders in coastal environments using random forest machine learning on topo-bathymetric Lidar data. Remote Sensing, 13(20), 4101.

Hansen, S. S., Ernstsen, V. B., Andersen, M. S., Al-Hamdani, Z., Baran, R., Niederwieser, M., ... & Kroon, A. (2022). Evaluation of Boulder Characteristics for Improved Boulder Detection Based on Machine Learning Techniques. Geosciences, 12(11), 421.

Hood, D., Sholes, S., Karunatillake, S., Fassett, C., Ewing, R., & Levy, J. (2022). The Martian Boulder Automatic Recognition System, MBARS. Earth and Space Science, 9. https://doi.org/10.1029/2022EA002410.

Jiang, S., Tung Kwok, C. Y., & Wong, M. S. (2019). Extraction of rocks and boulders on natural terrain using semantic segmentation. In 40th Asian Conference on Remote Sensing, ACRS (pp. 1-10).

Maharaja, D., Kumar, M., & Tiwari, A. Boulders and Craters Detection Using Transfer Learning.

Nagle-McNaughton, T., Williams, J., Gallegos, Z., Wilkie, H., Martinez, D., & Scuderi, L. (2019). Identification and Quantification of Boulder Distributions Near the Jezero Crater Fan System Using HiRISE Imagery. , 2089, 6136.

Palafox, L.F., et al. (2017). "Automated detection of geological landforms on Mars using CNNs." Computers & Geosciences, 101, 48-56.

Prieur, N., Amaro, B., Gonzalez, E., Kerner, H., Medvedev, S., Rubanenko, L., Werner, S., Xiao, Z., Zastrozhnov, D., & Lapôtre, M. (2023). Automatic Characterization of Boulders on Planetary Surfaces From High-Resolution Satellite Images. Journal of Geophysical Research: Planets, 128. https://doi.org/10.1029/2023JE008013.

Rothrock, B., et al. (2016). "SPOC: Deep learning-based terrain classification for Mars rover missions." AIAA SPACE Forum, 5539.

Su, M., Liu, Y., Xue, Y., Cheng, K., Kong, F., & Guan, L. (2021). Detection method for boulders in subway shield zones based on data fusion multi-resistivity three-dimensional tomography. Bulletin of Engineering Geology and the Environment, 80, 8171 - 8187. https://doi.org/10.1007/s10064-021-02406-z.

Yang, W., Su, D., Lin, X., Lv, Z., Yao, Z., Li, H., Zhang, Y., & Chen, X. (2024). Super-large diameter slurry shield tunnel encountering boulder formation: Detection, treatment and data analysis. Transportation Geotechnics. https://doi.org/10.1016/j.trgeo.2024.101408.

Zhu, L., Geng, X., Li, Z., & Liu, C. (2021). Improving YOLOv5 with attention mechanism for detecting boulders from planetary images. Remote Sensing, 13(18), 3776.

Google DeepMind (2025). https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/#gemini-2-5-thinking Access in April 8, 2025

Metz, Cade (September 12, 2024). "OpenAI Unveils New ChatGPT That Can Reason Through Math and Science". The New York Times. Retrieved September 12, 2024.

OpenAI (2025). "Introducing OpenAI o1". Retrieved April 5, 2025.

Anthropic (2025). https://www.anthropic.com/news/claude-3-7-sonnet Retrieved April 3, 2025.