

Project Stage II Report on

Automatic Detection of Craters and Boulders from OHRC Images Using Deep Learning

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Engineering

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

BY

Siddhi Kakani (B400350384)

Vedant Dhamale (B400350363)

Shruti Mahapurkar (B400350394)

Prachi Upare (B400350418)

Under the guidance of

Mrs.R.L.Ghule

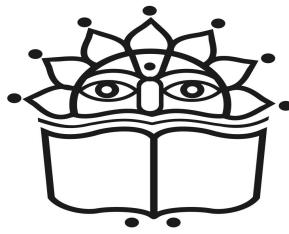
Vidya Pratishthan's

Kamalnayan Bajaj Institute of Engineering and Technology

Baramati-413133, Dist-Pune (M.S.) India

Department of Artificial Intelligence and Data Science

A.Y 2024-25



VPKBIET, Baramati

Certificate

This is to certify that the Project Stage II Report on
**Automatic Detection of Craters and Boulders from
OHRC Images Using Deep Learning**

SUBMITTED BY

Siddhi Kakani (B400350384)
Vedant Dhamale (B400350363)
Shruti Mahapurkar (B400350394)
Prachi Upare (B400350418)

in the partial fulfillment of the requirement for the award of Degree of Bachelor of
Engineering in Artificial Intelligence and Data Science at Vidya Pratishtan's
Kamalnayan Bajaj Institute of Engineering and Technology, Baramati, under the
Savitribai Phule Pune University, Pune. This work is done during year 2024-25
Semester-II, under our guidance.

Mrs. R. L. Ghule
Project Guide

Dr. C. S. Kulkarni
Head of Dept.

Dr. S. B. Lande
Principal

Examiner 1:

Examiner 2:

Acknowledgements

It gives us immense pleasure to submit this project report on **”Automatic detection of Craters and Boulders from OHRC Images using Deep Learning”** The successful completion of this project reflects our hard work and the effort of our guide in providing valuable guidance.

We would like to express our heartfelt thanks to our project guide, **Mrs.R.L.Ghule** for her unwavering support, which enabled us to make this project both insightful and informative. She not only provided us with the necessary literature and guidance but also offered the platform we needed to excel in this project. We are equally indebted to our project coordinator **Mrs.R.S.Naik** who has been a constant source of inspiration and guiding star in helping us achieve our goals. We extend our gratitude to **Dr.C.S.Kulkarni**, Head of Department of Artificial Intelligence and Data Science, and respected Principal **Dr.S.B.Lande** for their support and for providing all the necessary facilities to complete this work. We are also thankful and fortunate to have received continuous encouragement, support, and guidance from the entire teaching staff of the Department of Artificial Intelligence and Data Science, which greatly contributed to the successful completion of our project.

Siddhi Kakani (B400350384)

Vedant Dhamale (B400350363)

Shruti Mahapurkar (B400350394)

Prachi Upare (B400350418)

Abstract

Development of an automated system for detecting craters and boulders in high-resolution images from the Orbiter High Resolution Camera (OHRC) using deep learning techniques is crucial for advancing planetary exploration. By integrating the YOLOv11 object detection algorithm with the Segment Anything Model (SAM) for segmentation, the system ensures both accurate and efficient identification of geological features. Leveraging real-time images from the Chandrayaan-2 mission, this project enables a responsive and scalable approach to planetary surface analysis. Automation minimizes human involvement, reducing errors while enhancing detection speed and accuracy. The ability to process large datasets offers data-driven insights into craters and boulders distribution, contributing valuable information to planetary research. This system not only improves the efficiency of current missions but also supports future exploration by delivering fast, precise data for scientific analysis.

keywords - YOLOv11, SAM Model, Segmentation, Craters and Boulders Detection, Chandrayaan-2, OHRC, Deep Learning, Planetary Surface Analysis, Automated Detection.

List of Figures

4.1	System Architecture	10
7.1	Project Plan 2.0	23
8.1	Data Flow Diagram	24
8.2	Usecase Diagram	25
8.3	Activity Diagram	26
8.4	Class Diagram	27
8.5	Sequence Diagram	28
9.1	OHRC Images	29
9.2	Precision Curve	32
9.3	Recall Curve	33
9.4	Precision-Recall Curve	34
9.5	Confusion Metrix	35
9.6	Detection Result	37
9.7	Segmentation Result	37
9.8	Detection Accuracies	38
9.9	Detection Result Matric	38
9.10	Evaluation Result	39
10.1	GUI Result	44
D.1	Conference Certificate	54
D.2	Conference Certificate	54
D.3	Journal Certificate	55
D.4	Competition Certificate	55

List of Tables

2.1	Literature Survey	4
6.1	Team Structure	22

Notation and Abbreviations

0.1 Notation

- D:- Dataset containing n high resolution images.
- I:- Represent the images from 1-n.
- H:- Height of the image.
- W:- Width of the image.
- C:- Number of channels.
- $P_{i,j}$:- Pixel of the image at i^{th} row and j^{th} column.
- F:- Feature of the image.
- X:- Input Feature map
- K:- Convolution kernel
- b:- Bias term
- l_i :- Predicted class labels for the object within the bounding box b_i
- x_i, y_i :- Coordinates of the center of the boundary box.
- w_i, h_i :-Width and height of the boundary box.
- F_{shallow} :- Represent operations(convolution,activation) in the shallow layer.
- F_{input} :- Feature map output.
- F_{deep} :- Operation in the deep layer.
- $S_{\text{segmentation}}$:-Represent the predicted segmentation mask.
- $\sigma(F_{\text{output}})$:-Activation function applied to produce probability maps.

0.2 Abbreviations

- CNN:-Convolutional Neural Network.
- DA:-Domain Adaption.
- R-CNN:-Region-based Convolutional Neural Network.
- CCD:- Charge Coupled Device.
- DEM:- Digital Elevation Model.
- CSP:- Cross Stage Partial.

Contents

Acknowledgements	i
Abstract	ii
Notation and Abbreviations	v
0.1 Notation	v
0.2 Abbreviations	vi
1 Introduction	1
1.1 Introduction	1
1.2 Motivation	2
2 Literature Survey	3
2.1 Gaps	7
3 Proposed System	8
3.1 Problem Definition	8
3.2 Project Objectives	8
3.3 Scope of Project	8
3.4 Project Constraints	9
4 Proposed System Architecture	10
4.1 Architecture	10
4.2 Mathematical Model	13
4.3 Proposed Algorithm	15
5 Project Requirement Specification	16
5.1 Hardware Requirements	16

5.2 Software Requirements	16
5.3 Performance Requirements	17
5.4 Software Quality Attributes/Requirements	17
5.5 Security Requirements	18
5.6 Other Requirements	18
6 Project Planning	19
6.1 Project Estimates	19
6.1.1 Effort, Duration, and Personnel	20
6.2 Team Structure	22
7 Project Schedule	23
7.1 Project Breakdown Structure	23
8 Project Design	24
8.1 UML Diagrams	24
8.1.1 Data Flow Diagram	24
8.1.2 Usecase Diagram	25
8.1.3 Activity Diagram	26
8.1.4 Class Diagram	27
8.1.5 Sequence Diagram	28
9 Results and Experimentation	29
9.1 Experimental Setup	29
9.2 Test Specifications	30
9.2.1 Assumptions and Dependencies	30
9.3 Performance Measures	31
9.3.1 Accuracy	31
9.3.2 Precision Curve	32
9.3.3 Recall Curve	33
9.3.4 Precision-Recall Curve	34
9.3.5 Confusion Matrix	35
9.3.6 Summary and Interpretation	35
9.4 Experimental Results	36
9.4.1 Exact Method	36

9.4.2 Result Analysis	38
9.5 Discussions	40
9.5.1 Limited Availability of Annotated Lunar Data	40
9.5.2 Performance Limitations in Detecting Small and Overlapping Objects	40
9.5.3 Generalization Issues Across Lunar Terrains	40
9.5.4 Computational and Hardware Constraints.	41
10 Proposed GUI	42
10.1 Proposed GUI	42
10.2 Backend Module	44
11 Conclusion	45
References	46
A Plagiarism Report	48
B Base Paper	49
C Tools Used	51
D Papers Published/Certificates	53
D.1 Copyrights	53
D.2 Certificates	54
D.3 Paper Published	56

Chapter 1

Introduction

1.1 Introduction

Over the past few years, various space missions from organizations like NASA, ISRO, and other international space agencies have gathered vast amounts of data from planetary surfaces, particularly the Moon and Mars. Missions such as Chandrayaan-1 and 2, the Lunar Reconnaissance Orbiter (LRO), and Chang'E have captured high-resolution images that are vital for analyzing the surface features of these celestial bodies. These images provide valuable insights into impact craters and boulders, which are essential for studying the geological history, surface evolution, and for planning future exploration efforts.

Identifying craters and boulders is crucial for understanding a planet's impact history and for determining safe landing sites for upcoming missions. However, the vast amount of data collected from lunar and planetary missions exceeds the capacity of human operators to manually analyze and interpret. By automating this process using deep learning techniques, surface feature detection becomes more efficient and accurate.

Detecting surface features like craters and boulders on planetary bodies is essential for scientific exploration, hazard evaluation, and mission planning. High-resolution imagery, such as that from the Optical High Resolution Camera (OHRC), provides valuable data for identifying these features. However, manually analyzing such large-scale imagery is both time-consuming and susceptible to human error.

To automate this task, advanced computer vision techniques have been developed. In this study, we utilize YOLOv11, a cutting-edge object detection model, alongside the Segment Anything Model (SAM), to detect and segment craters and boulders in OHRC images. YOLOv11 is used for its real-time object detection capabilities, while SAM ensures accurate and flexible segmentation of the detected objects. By integrating these two powerful models, the proposed system aims to offer efficient and precise automatic detection of craters and boulders, enhancing the analysis of planetary surfaces.

1.2 Motivation

Exploring the Moon is an exciting and important goal for scientists and space agencies. Understanding its surface, especially features like craters and boulders, helps us learn more about the Moon's history, structure, and how it has evolved over time. These surface features also play a key role in selecting safe landing sites for future space missions. However, manually identifying craters and boulders in high-resolution satellite images is a difficult, time-consuming task that requires expert knowledge and can still lead to human errors.

To solve this problem, there is a strong need for an automated system that can detect and analyze surface features quickly and accurately. Our motivation comes from the idea of using deep learning technology to reduce human effort while increasing the speed and precision of detection. With the availability of high-resolution OHRC (Orbiter High Resolution Camera) images from Chandrayaan-2 and platforms like Roboflow, it becomes possible to train modern AI models for this task. These images provide detailed views of the lunar surface, which can be used to teach the system how to find craters and boulders.

By using models like YOLOv11 for object detection and SAM for segmentation, we can build a system that works effectively on real satellite data. This will not only improve the efficiency of lunar surface analysis but also help support future space missions by identifying suitable locations for landing, construction, or exploration. The ultimate aim is to make planetary research more data-driven and less dependent on manual interpretation. Our project will contribute to the advancement of planetary science by making surface mapping faster, safer, and more reliable. It can also serve as a foundation for similar applications on other celestial bodies like Mars or asteroids in the future. The motivation behind this project is to bridge the gap between AI and space exploration for a smarter and more accurate future in space missions.

Chapter 2

Literature Survey

Sr No	Papers	Techniques used	Advantages	Research Gaps
1	[1]Z.Zhang, et.al, 2023	<ul style="list-style-type: none">• CNN• Unsupervised Domain Adaption• Causal Inference-based Feature Matching	<ul style="list-style-type: none">• Cross domain Detection• Handle Small Craters	<ul style="list-style-type: none">• Computational expensive• Scale-dependent detection
2	[2]Atheer L.,et.al,2017	<ul style="list-style-type: none">• Crater Detection Algorithm• Crater Size-Frequency Distribution	<ul style="list-style-type: none">• Faster Detection• Scalability	<ul style="list-style-type: none">• Detection Errors
3	[3]Ari Silburt ,et.al,2019	<ul style="list-style-type: none">• CNN• ADAM Optimizer	<ul style="list-style-type: none">• Faster Detection• Transfer Learning	<ul style="list-style-type: none">• Scale-dependent detection• Incomplete augmentation

Sr No	Papers	Techniques used	Advantages	Research Gaps
4	[4]Ebrahim Emami ,et.al,2019	<ul style="list-style-type: none"> • Hypothesis Generation • Hypothesis Verification • CNN 	<ul style="list-style-type: none"> • Multi-scale Detection • Efficient Process 	<ul style="list-style-type: none"> • Scale-dependent detection • Mis-identification
5	[5]Rasha Alshehhi, et.al,2022	<ul style="list-style-type: none"> • Domain Adaption • CNN • Mask R-CNN 	<ul style="list-style-type: none"> • Cross-planet detection • Less manual annotation 	<ul style="list-style-type: none"> • Scale-dependent detection • Imbalanced Data
6	[6] Jionghao Zhu et.al 2023	<ul style="list-style-type: none"> • YOLO V7 • YOLO v5n • YOLO v5s • Charge-Coupled Device & Digital Elevation Model data 	<ul style="list-style-type: none"> • Faster processing • Multi-source data 	<ul style="list-style-type: none"> • Use small dataset • Single-stage model • Lower DEM performance
7	[7]Sourish Chatterjee et.al,2023	<ul style="list-style-type: none"> • YOLO v5 • CSP-Darknet53 • Adaptive Thresholding 	<ul style="list-style-type: none"> • Micro craters detection • High speed 	<ul style="list-style-type: none"> • Overfitting on small dataset • Size sensitive
8	[8]Quan Duan et.al,2024	<ul style="list-style-type: none"> • Digital Elevation Model • Max Curvature Detection Method • Watershed Algorithm 	<ul style="list-style-type: none"> • High Detection • Scalability 	<ul style="list-style-type: none"> • Scale-dependent detection • Mis-identification

Table 2.1: Literature Survey

[1] Planet Craters Detection Based on Unsupervised Domain Adaptation

The detection of craters and boulders in planetary surface images has been a significant topic in remote sensing and planetary science, particularly with the advancement of deep learning technologies. Various researchers have proposed solutions based on CNNs, YOLO architectures, domain adaptation techniques, and image segmentation to improve detection accuracy, generalization, and processing speed across different planetary terrains. Zhang et al. (2023) introduced a novel method that combined Convolutional Neural Networks (CNNs) with Unsupervised Domain Adaptation and Causal Inference-based Feature Matching. This approach addressed the problem of domain shift between source (e.g., simulated or lunar images) and target domains (e.g., real planetary images), enabling robust cross-domain detection. Their method also improved the detection of small craters, which are often difficult to identify. However, the algorithm was computationally expensive, making real-time application challenging. Moreover, the model's performance was scale-dependent, meaning it performed differently across craters of varying sizes.

[2] Automatic crater detection and age estimation for mare regions on the lunar surface

Atheer L. et al. (2017) proposed a Crater Detection Algorithm utilizing Crater Size-Frequency Distribution (CSFD) metrics, which is a traditional method for age estimation and crater cataloging. Their approach provided faster detection and better scalability, especially when applied to large datasets. Despite this, it suffered from detection errors due to reliance on fixed thresholds and image quality.

[3]Lunar crater identification via deep learning

Ari Silburt et al. (2019) leveraged a CNN trained with the ADAM optimizer, highlighting the role of deep learning in accelerating crater detection. They also experimented with transfer learning, showing that pre-trained models could generalize across different planetary bodies. Yet, the model's accuracy was limited due to scale sensitivity and incomplete data augmentation, which impacted generalization.

[4] Crater Detection Using Unsupervised Algorithms and Convolutional Neural Networks

Ebrahim Emami et al. (2019) employed a two-stage hypothesis generation and verification framework powered by CNNs. This method enabled multi-scale crater detection and was efficient in processing high-resolution planetary data. Nevertheless, the model occasionally led to misidentification of craters versus other circular formations and still suffered from scale-dependency.

[5] Automated Geological Landmarks Detection on Mars Using Deep Domain Adaptation From Lunar High-Resolution Satellite Images

Rasha Alshehhi et al. (2022) combined domain adaptation techniques, CNNs, and Mask R-CNN for cross-planet crater detection. One of the key strengths of their work was reducing the need for manual annotations using transfer learning and segmentation-based approaches. However, the model's performance decreased in the presence of imbalanced datasets and was still constrained by scale-dependent limitations.

[6] Lunar Impact Crater Detection Based on Yolo V7 Using Muti-source Data

Jionghao Zhu et al. (2023) evaluated the performance of YOLOv7, YOLOv5n, and YOLOv5s models using Charge-Coupled Device (CCD) and Digital Elevation Model (DEM) data. Their approach significantly enhanced processing speed and incorporated multi-source data fusion. However, they relied on a small training dataset, and the DEM-based features underperformed compared to visual inputs. Additionally, the single-stage YOLO architecture could not capture complex features as effectively as two-stage models.

[7] Near-Real-Time Detection of Craters: A YOLO v5 Based Approach

Sourish Chatterjee et al. (2023) used YOLOv5 with CSP-Darknet53 backbone and adaptive thresholding to improve micro-crater detection. Their system achieved high inference speed and was suitable for deployment on limited hardware. But it faced

overfitting issues on small datasets and had difficulty handling varying object sizes, making it less robust in real-world scenarios.

[8] Automatic Lunar Crater Detection Based on DEM Data Using a Max Curvature Detection Method

Lastly, Quan Duan et al. (2024) utilized Digital Elevation Models (DEMs) in conjunction with the Max Curvature Detection Method and the Watershed Algorithm. This technique achieved high detection accuracy and demonstrated strong scalability across terrain types. However, like others, it struggled with scale-dependence and had issues of misidentification due to elevation artifacts or shadows.

2.1 Gaps

One of the key challenges in crater and boulder detection lies in the scale dependency of many models, which struggle to accurately identify objects of varying sizes. This limitation reduces the model's effectiveness when dealing with diverse geological formations. Moreover, while advanced techniques such as deep convolutional neural networks (CNNs) and causal inference frameworks have improved detection accuracy, they also introduce significant computational complexity. This makes them less practical for real-time applications or deployment in environments with limited processing resources. Another common issue in deep learning-based approaches is overfitting, especially when models are trained on small or insufficiently diverse datasets, leading to poor generalization on unseen or varied input data. Additionally, an imbalanced distribution of training data—where crater-like features are much rarer than other classes—can skew the learning process and result in biased or less reliable predictions. These challenges highlight the need for efficient, scalable, and well-regularized models capable of handling real-world variability in planetary surface analysis.

Chapter 3

Proposed System

3.1 Problem Definition

Manual analysis of high-resolution planetary images is time-consuming and error-prone. The goal is to develop a system using Deep Learning to detect and classify craters and boulders in OHRC images.

3.2 Project Objectives

1. To detect craters and boulders using Deep Learning.
2. To segment irregular shapes precisely.
3. To analyze the model's performance.

3.3 Scope of Project

The scope of this project includes important improvements to the system for detecting craters and boulders. Enhancements will be made to YOLOv8 by training it on a larger and more varied dataset that includes different types of planetary surfaces. The project will also incorporate real-time detection features, allowing for quick identification of craters and boulders during missions. Additionally, the detection system will be adapted to work with images from other planetary bodies, such as Mars and the Moon. Detection results will be combined with other planetary data to provide a more comprehensive analysis. Lastly, the system will enable real-time analysis to deliver immediate feedback and support decision-making during orbital missions.

3.4 Project Constraints

1. Access to sufficient high-resolution OHRC images may be restricted.
2. Development and experimentation must fit within a limited timeline and budget.
3. Large image datasets demand significant storage capacity.
4. Limited availability of labeled data for training affects model performance and generalization.

Chapter 4

Proposed System Architecture

4.1 Architecture

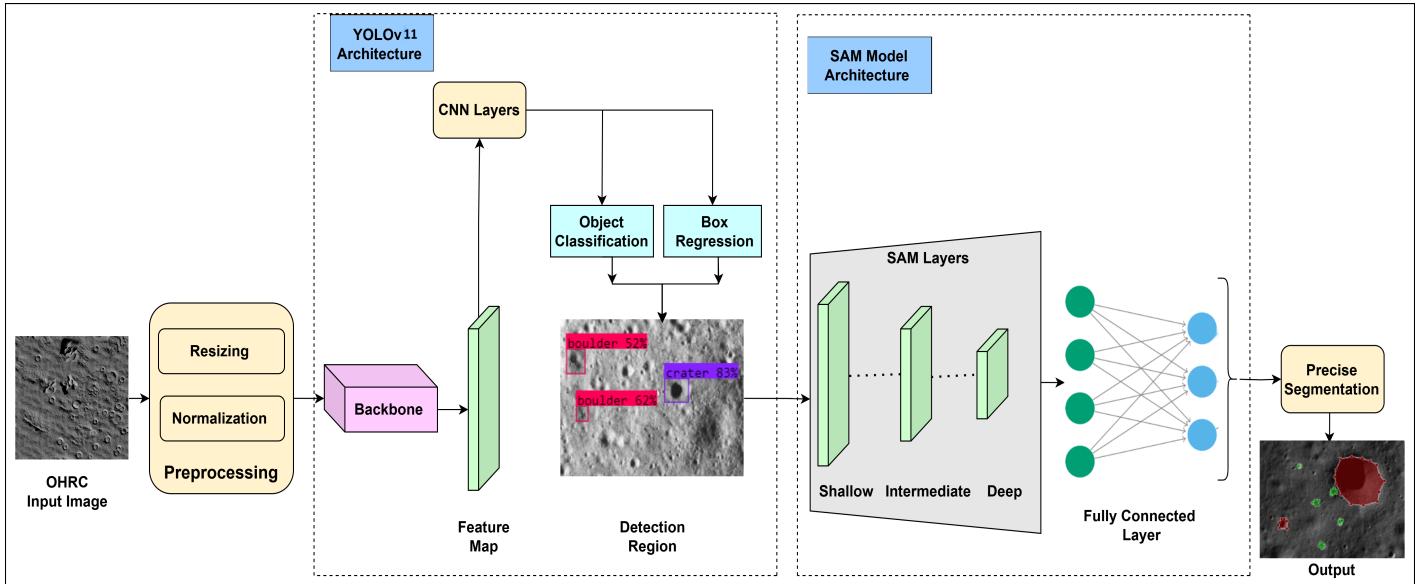


Figure 4.1: System Architecture

Explanation of the architecture:

The proposed system architecture focuses on the analysis of high-resolution lunar surface imagery, specifically for the detection and segmentation of craters and boulders. This process leverages advanced computer vision techniques and deep learning models to achieve precise object identification and segmentation. The key stages of the system are detailed below:

1. Input Data Acquisition

The process begins with acquiring images from the Orbiter High Resolution Camera (OHRC). These high-fidelity images serve as the primary input for subsequent processing.

2. Preprocessing

To optimize the raw images for model inference, several preprocessing techniques are applied:

- **Resizing:** The input images are resized to align with the dimensions required by the model.
- **Normalization:** Pixel values are normalized to enhance model performance by ensuring consistent data scaling.

These preprocessing steps ensure that the images are in the correct format and scale, ready for feature extraction.

3. Feature Extraction Using Backbone Network

The preprocessed images are passed through a backbone network, which is responsible for extracting high-level features from the input. This process results in the creation of a *Feature Map*, which captures essential information for object detection.

4. Object Detection with YOLOv11

The extracted feature map is fed into the **YOLOv11 (You Only Look Once Version 11)** architecture, which utilizes several Convolutional Neural Network (CNN) layers for efficient object detection. The YOLOv11 model performs the following tasks:

- **Object Classification:** Identifies and classifies specific objects (such as craters and boulders) present in the image.
- **Box Regression:** Determines the precise bounding boxes around the detected objects, localizing them within the image.

The result of this stage is a *Detection Region*, where objects are accurately labeled and enclosed within bounding boxes.

5. Precise Segmentation with SAM Model

The detected regions are subsequently processed using the **SAM (Segmentation Attention Model)** architecture. The SAM model is structured into multiple layers to enhance segmentation accuracy:

- **Shallow, Intermediate, and Deep Layers:** These layers progressively refine the detected regions, ensuring detailed segmentation.
- **Fully Connected Layer:** This final layer consolidates outputs from different depths to make a comprehensive segmentation decision.

The SAM model generates a segmented output, providing precise delineation of the detected objects.

6. Output Generation

The final output of the system is an annotated image where craters and boulders are clearly segmented and labeled, facilitating further scientific analysis and research. This detailed segmentation aids in understanding the lunar surface's topography and geological features.

4.2 Mathematical Model

Input Dataset Representation

1. Dataset Representation

Let \mathcal{D} be the dataset containing n high-resolution images. Each image I_i belongs to $\mathbb{R}^{H \times W \times C}$.

$$\mathcal{D} = \{I_1, I_2, \dots, I_n\}$$

2. Input

$$I \in \mathbb{R}^{H \times W \times C}$$

3. Data Preprocessing and Normalization

Normalization

Normalize each pixel value P_{ij} for $j=1,2,\dots,m$ to scale the values between 0 and 1 using min-max scaling:

$$p'_{ij} = \frac{p_{ij} - \min(p_i)}{\max(p_i) - \min(p_i)}$$

YOLOv11 Model

1. Backbone

It extracts features from input images using convolutional layers and downsampling techniques.

- Convolutional Layer

$$F = \sigma(X * K + b)$$

- DownSampling

$$F_{down} = \text{Pool}(F)$$

2. Combining Feature Map

$$F_{out} = F_l + \text{Upsample}(F_{l+1})$$

YOLOv11 Output Representation

1. Output

- Bounding Box

$$B = \{b_1, b_2, \dots, b_n\}$$

- Class labels

$$L = \{l_1, l_2, \dots, l_n\}$$

- Bounding Box Represented

$$b_i = \{x_i, y_i, w_i, h_i\}$$

SAM Layer

1. Shallow Layers

$$F_{\text{shallow}} = f_{\text{shallow}}(F_{\text{input}})$$

2. Intermediate Layer

$$F_{\text{intermediate}} = f_{\text{intermediate}}(F_{\text{shallow}})$$

3. Deep Layer

$$F_{\text{deep}} = f_{\text{deep}}(F_{\text{intermediate}})$$

Fully Connected Layer

1. Flattening

$$F_{\text{flattened}} = \text{Flatten}(F_{\text{deep}})$$

2. Fully Connected Layer

$$F_{\text{output}} = W_{\text{fc}} \cdot F_{\text{flattened}} + b_{\text{fc}}$$

Segmentation Output

1. Segmentation Prediction

$$S_{\text{segmentation}} = \sigma(F_{\text{output}})$$

4.3 Proposed Algorithm

Algorithm 1: Proposed Algorithm

Input: OHRC Images

Output: Detected Craters and Boulders

1 Training Phase: YOLOv11 Model(dataset)

2 for $epoch = 0$ to $epochs = 100$ **do**

3 model \leftarrow YOLOv11()

4 **for** $i = 0$ to n **do**

5 (image, label) \leftarrow dataset[i]

6 model.train(image, label)

7 **end**

8 **end**

9 Testing Phase: Object Detection(image)

10 output \leftarrow model.predict(image)

11 bboxes \leftarrow extract bounding boxes from output

12 Segmentation with SAM (Optional)

13 for $i = 0$ to $len(bboxes)$ **do**

14 mask[i] \leftarrow SAM(image, prompt = bboxes[i])

15 **end**

16 System Implementation Phase

17 trained_model \leftarrow YOLOv11(dataset)

18 input_image \leftarrow load image

19 output \leftarrow Object Detection(input_image)

20 masks \leftarrow SAM(input_image, output)

21 display(output, masks)

Chapter 5

Project Requirement Specification

5.1 Hardware Requirements

1. **Disk Space:** Minimum 512 GB SSD.
2. **Processor:** 12th Gen Intel Core i9
3. **GPU:** NVIDIA RTX A4000 or higher
4. **RAM:** Minimum 16 GB

5.2 Software Requirements

1. **OS:** Windows 10
2. **OS Type:** 64-bit
3. **Python Version:** 3.11.4
4. **Tools:** Google Colaboratory/ Jupyter Notebook /Kaggle

5.3 Performance Requirements

1. Accuracy:

- The YOLOv11 model must achieve higher accuracy in crater and boulder detection compared to traditional object detection models (e.g., YOLOv4 or other classical ML models).
- The SAM (Segment Anything Model) should achieve a segmentation accuracy with an IoU (Intersection over Union) score of at least 85% for precise boundary extraction of craters and boulders.

2. Real-Time Detection:

- The system should detect and segment craters and boulders in real-time, with a processing latency of less than 1 second per image. This includes both the YOLOv11 detection and SAM segmentation process when running on a suitable GPU

5.4 Software Quality Attributes/Requirements

1. **Accuracy:** The system should accurately detect and segment craters and boulders in high-resolution imagery. The combination of YOLOv11 for object detection and SAM for segmentation ensures precise identification of surface features.
2. **Efficiency:** The system must process large volumes of planetary surface data quickly and efficiently. By automating the detection of craters and boulders, the system significantly reduces the time required for analysis compared to manual methods.
3. **Scalability:** The system should be scalable to handle the vast and growing amount of data collected from future space missions, enabling it to process increasingly larger datasets from planetary surfaces.
4. **Reliability:** The system needs to consistently provide accurate results across various datasets and conditions, ensuring that the detection and segmentation processes are dependable for scientific analysis and mission planning.
5. **Maintainability:** The system should be designed with modular architecture and well-documented code, making it easy to update and modify as AI techniques and data collection methods evolve over time.

6. **Adaptability:** The system should be adaptable to different planetary environments (e.g., the Moon, Mars), varying resolutions, and future imaging technologies. This allows it to stay relevant as new missions collect more data from other celestial bodies.
7. **Testability:** The system should allow comprehensive testing of its object detection and segmentation components. Rigorous validation processes ensure that the system performs accurately under different conditions, minimizing the risk of errors.
8. **Usability:** The system should be user-friendly for scientists and engineers, providing intuitive interfaces or outputs that allow easy interpretation of detected surface features for further analysis or mission planning.

5.5 Security Requirements

1. **Data Protection:** Ensure that the images and any input data used for detection are stored securely and are protected from unauthorized access.
2. **Model Integrity:** Make sure that the YOLOv11 and SAM models are protected from tampering or unauthorized modifications. Store model files in a secure environment, with access controls in place.
3. **Input Validation:** Ensuring that the inputs (images) are from trusted sources and in the right format helps avoid system failures or potential security risks from maliciously crafted input data.

5.6 Other Requirements

The craters and boulders detection system should comply with all relevant data privacy regulations and industry standards, as outlined in the Security Requirements section. The system must respect data ownership and usage restrictions, ensuring that high-resolution imagery and detection results are collected, processed, and utilized only for authorized scientific purposes and with proper consent from relevant space agencies. Additionally, the system should implement suitable data masking and de-identification techniques to protect individual privacy while allowing for effective analysis and feature detection. This is particularly important when sharing data with research teams or publishing findings, ensuring that sensitive information is not disclosed.

Chapter 6

Project Planning

This plan is the basis for the execution for the tracking of all the project activities. It shall be used throughout the life of the project and shall be kept up to date to reflect the actual accomplishments and plans of the project

6.1 Project Estimates

Project estimation is a crucial aspect of software engineering, aimed at predicting the effort, time, and cost needed to complete a project. It involves approximating key project parameters, especially when input data is uncertain or incomplete. Accurate estimates are essential for effective planning, budgeting, and resource management. The COCOMO model helps estimate effort (in person-months), development time, and costs. Since our project includes machine learning, data processing, and image detection, it falls into the "Semi-Detached" mode, characterized by moderate software complexity. Effort, duration, and staffing estimates for the project are determined based on its size, measured in KLOC (Kilo Lines of Code). With an estimated size of 1.1 KLOC, the COCOMO-2 model is applied for the calculations.

6.1.1 Effort, Duration, and Personnel

The effort (person-months), development duration (months), and personnel (team size) estimates can be determined using the following formulas:

$$\text{Effort (E)} = A_b \times (KLOC)^{B_b}$$

$$\text{Duration (D)} = C_b \times (E)^{D_b}$$

$$\text{Personnel (P)} = \text{Effort} / \text{Duration}$$

Where:

KLOC = 1.1 (estimated project size in KLOC)

A_b , B_b , C_b , D_b are COCOMO-2 coefficients for a Semi-Detached project.

Effort Applied (Person-Months)

Effort is calculated using the formula:

$$\text{Effort (E)} = A_b \times (KLOC)^{B_b}$$

For $A_b = 2.4$ and $B_b = 1.05$ (semi-detached mode):

$$E = 2.4 \times (1.1)^{1.05}$$

$$E \approx 2.6 \text{ Person - Months}$$

Development Time (Months)

Development time is calculated using:

$$\text{Duration(D)} = C_b \times (E)^{D_b}$$

For $C_b = 2.5$ and $D_b = 0.38$:

$$D = 2.5 \times (2.6)^{0.38}$$

$$D \approx 3.59 \text{ Months}$$

Number of Team Members (Personnel) : 4

Productivity

Productivity is calculated by :

$$\text{Productivity} = \frac{\text{Effort}}{KLOC}$$

$$\text{Productivity} = \frac{2.6}{1.1}$$

$$\text{Productivity} \approx 0.423 KLOC/\text{Person - Month}$$

$$\text{Productivity} \approx 423 KLOC/\text{Person - Month}$$

6.2 Team Structure

Name	Role	Responsibility
Siddhi S. Kakani	Project Manager	Project Planning, Coordination and implementation of YOLOv11 and SAM models.
Vedant S. Dhamale	SRS Designer and Developer	SRS creation, UML modeling, functional requirement analysis, and co-development of YOLOv11 and SAM models.
Shruti S. Mahapurkar	Documentation and UI Developer	Technical report writing, literature review, requirement analysis, and front-end development using Gradio, HTML, and CSS.
Prachi B.Upare	Documentation and UI Developer	Structured documentation, diary maintenance, literature survey, and front-end development using Gradio, HTML, and CSS.

Table 6.1: Team Structure

Chapter 7

Project Schedule

7.1 Project Breakdown Structure

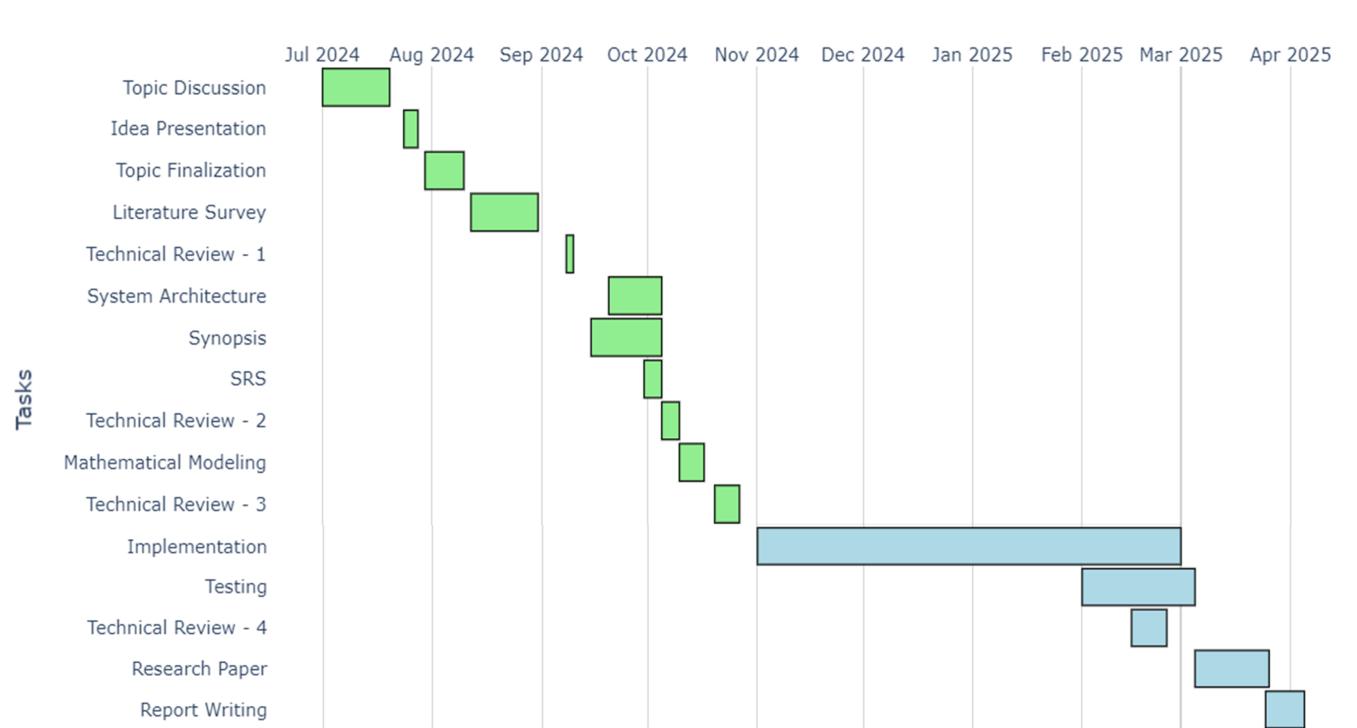


Figure 7.1: Project Plan 2.0

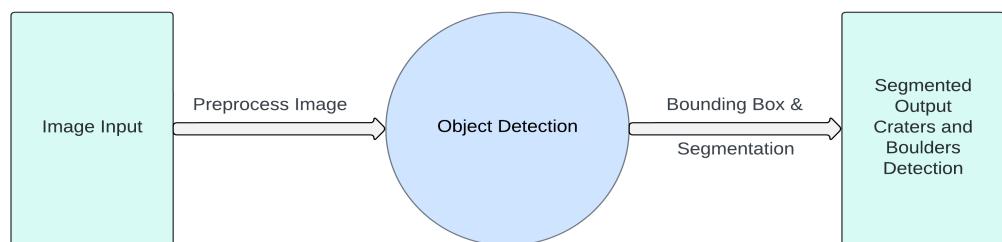
Chapter 8

Project Design

8.1 UML Diagrams

8.1.1 Data Flow Diagram

Level-0



Level-1

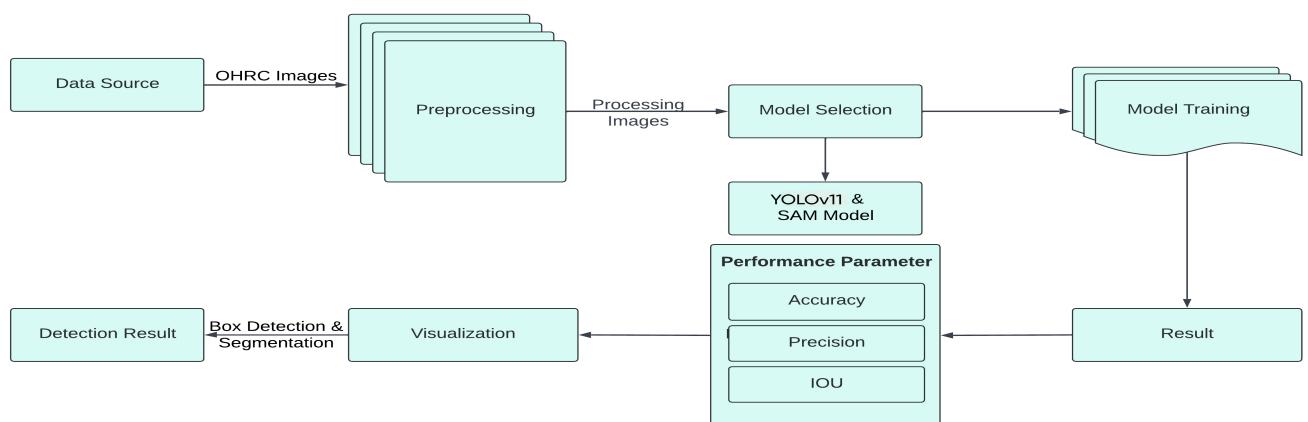


Figure 8.1: Data Flow Diagram

8.1.2 Usecase Diagram

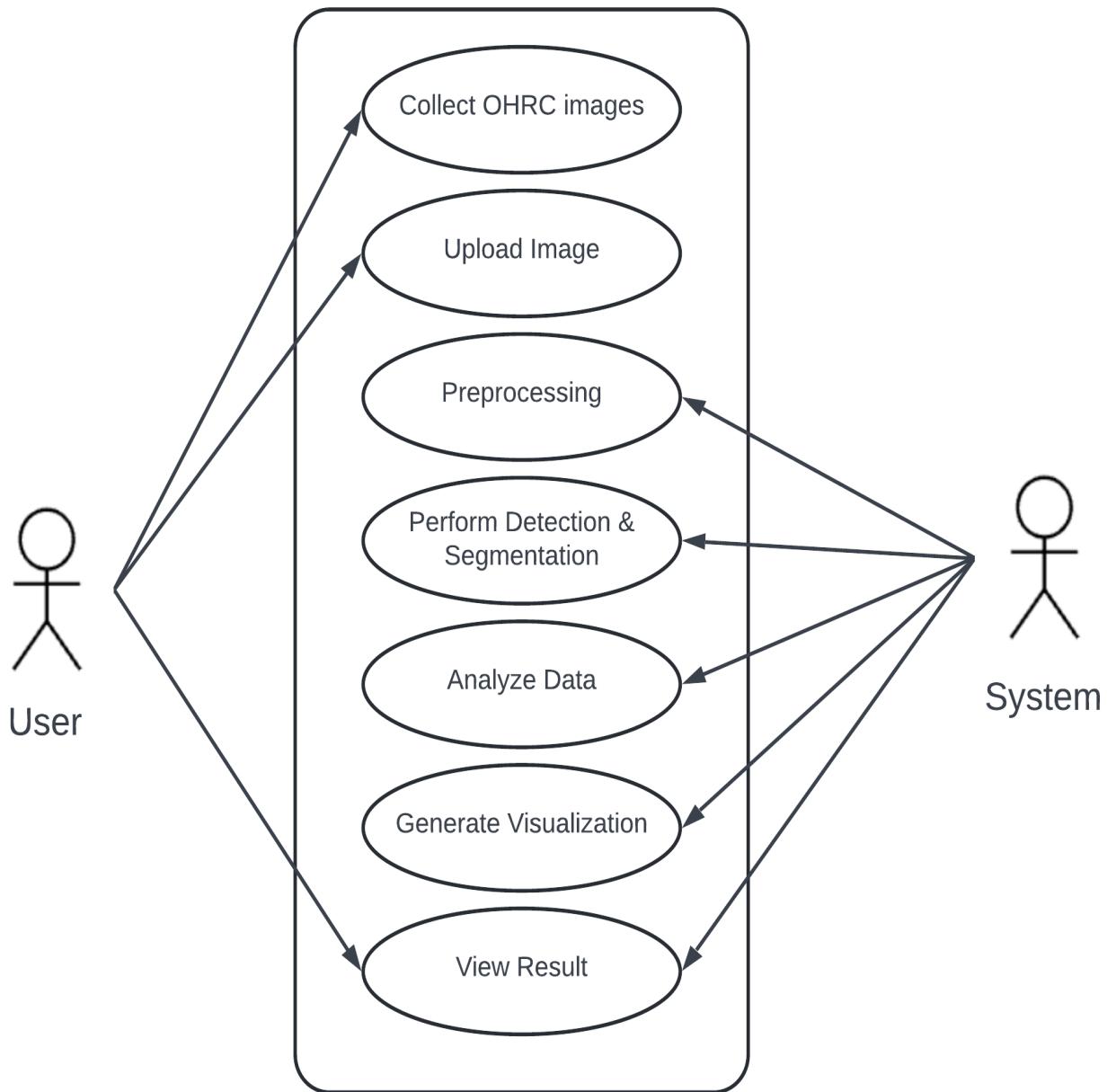


Figure 8.2: Usecase Diagram

8.1.3 Activity Diagram

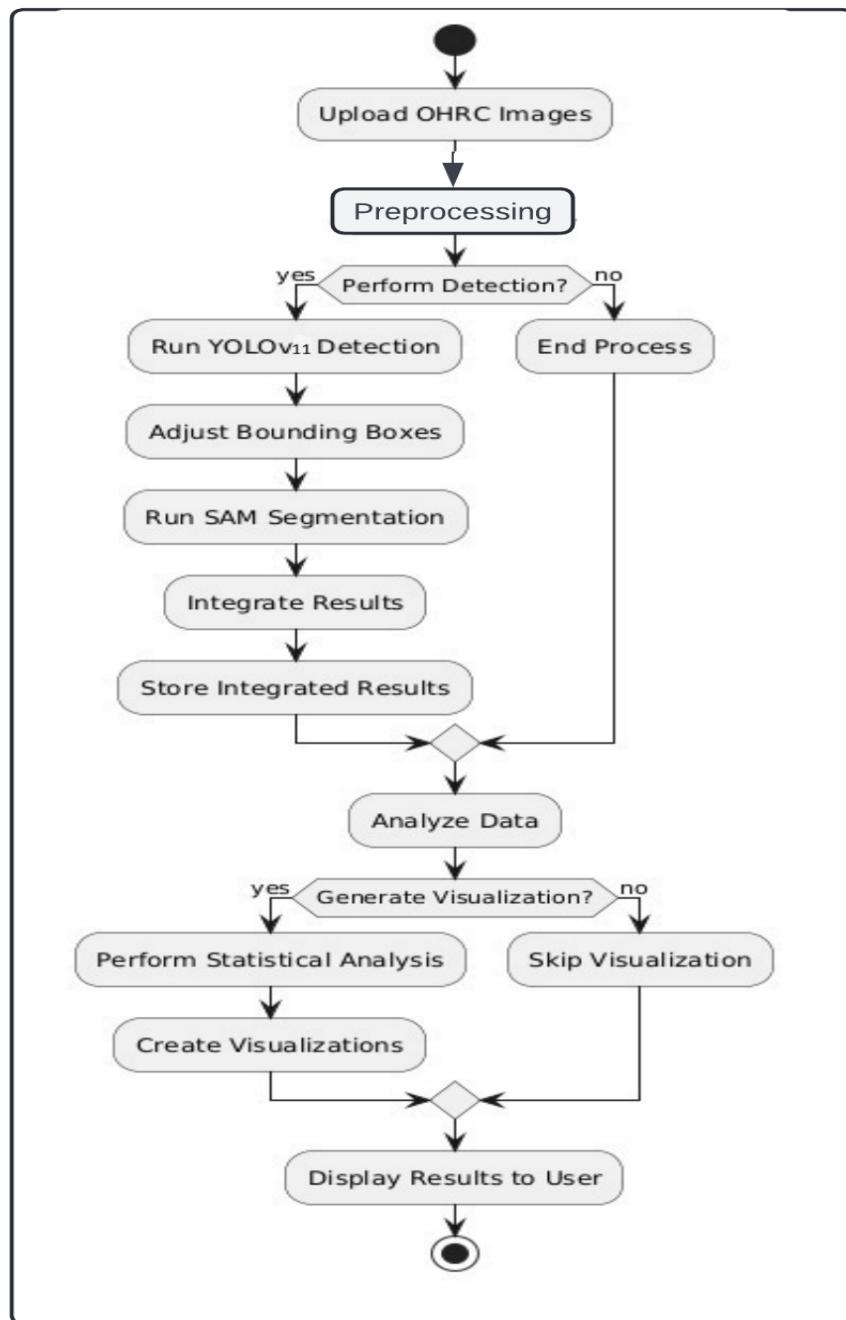


Figure 8.3: Activity Diagram

8.1.4 Class Diagram

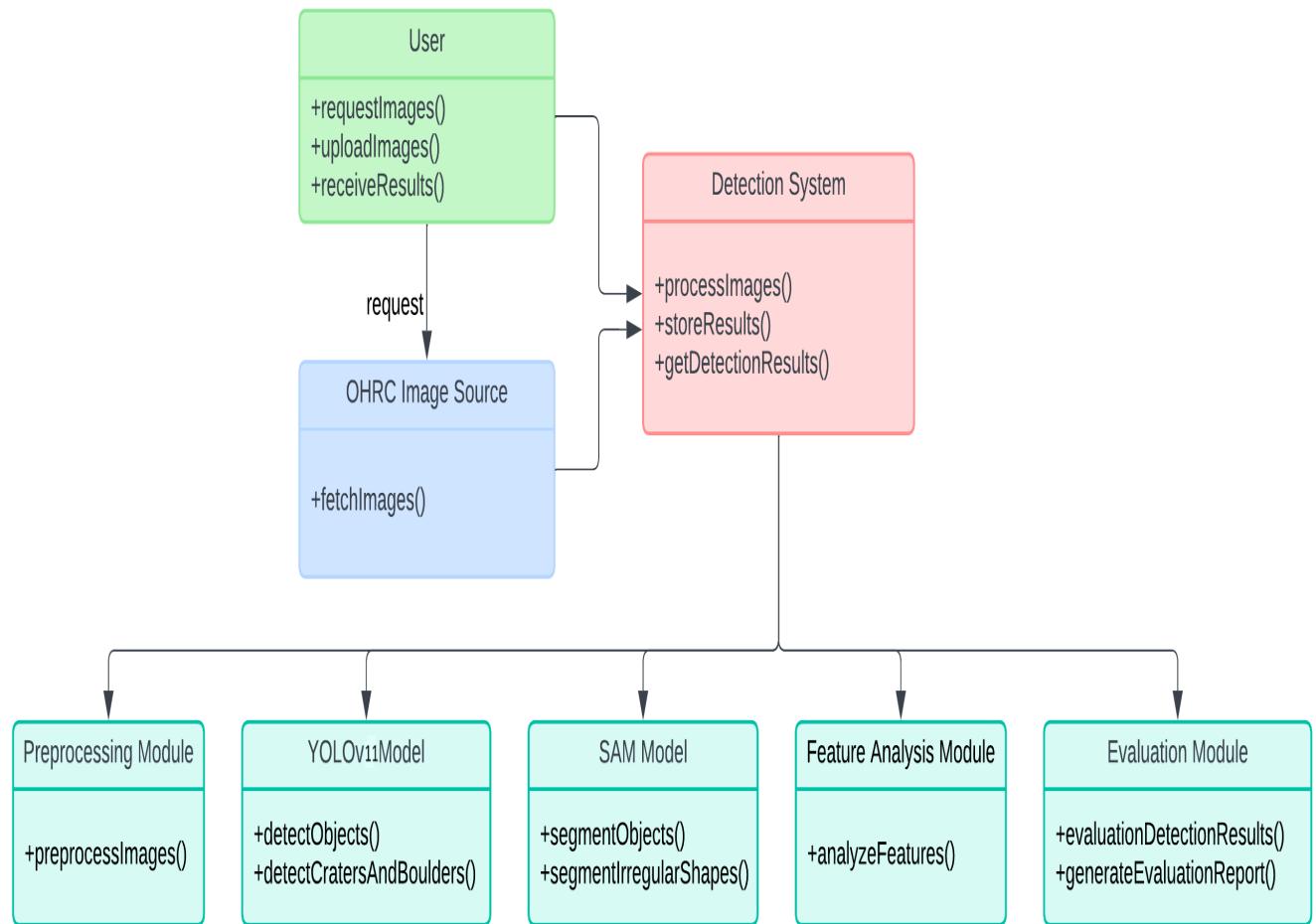


Figure 8.4: Class Diagram

8.1.5 Sequence Diagram

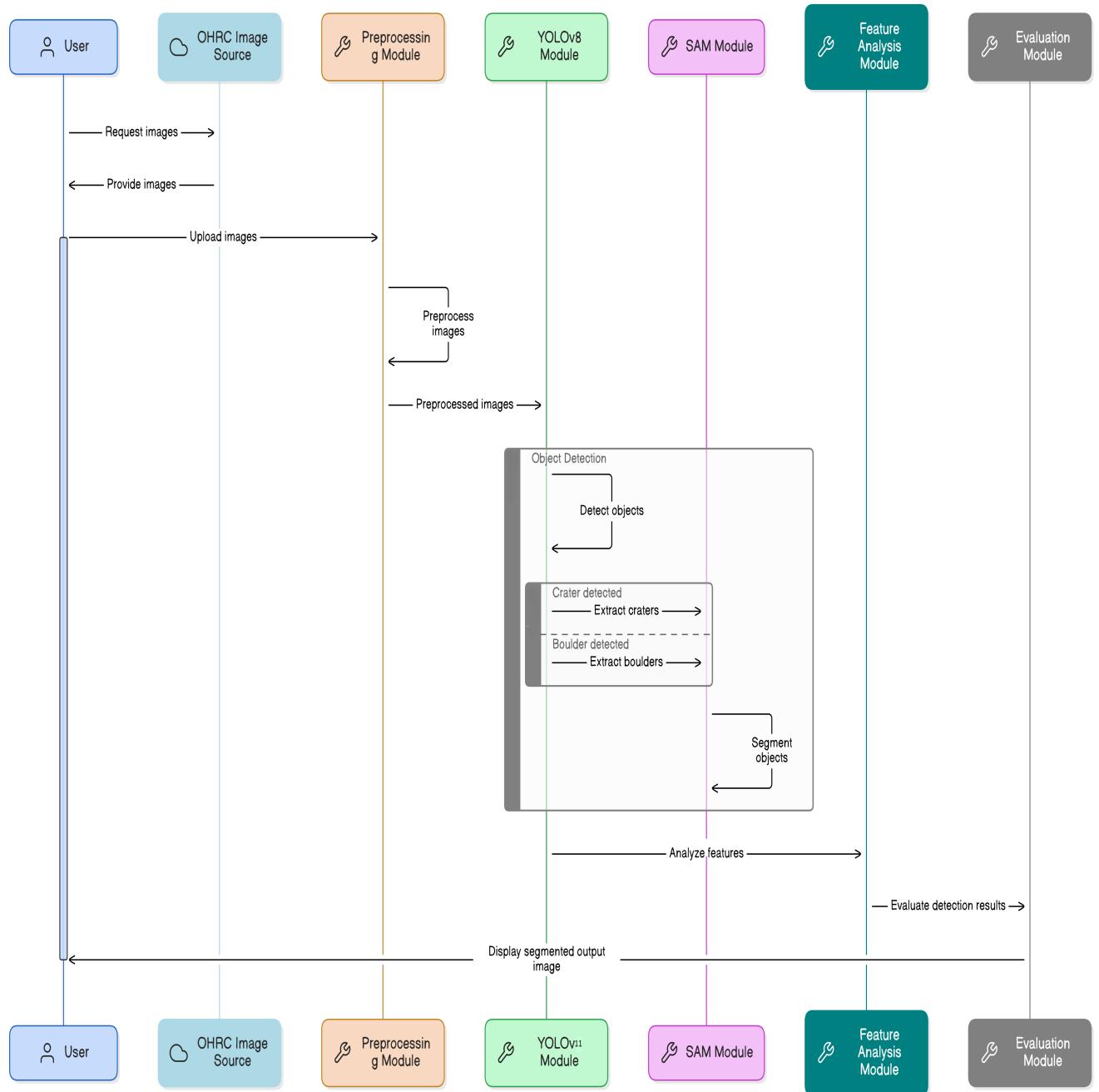


Figure 8.5: Sequence Diagram

Chapter 9

Results and Experimentation

9.1 Experimental Setup

For the purpose of experimentation, we have collected high-resolution OHRC (Orbiter High Resolution Camera) images from the Chandrayaan-2 mission provided by ISRO and other publicly available datasets from platforms like Roboflow. These datasets contain detailed imagery of the lunar surface with visible crater and boulder formations, which are essential for training and evaluating the deep learning models used in this project.

This collection process helped us understand the nature of lunar topography and the variation in crater and boulder shapes, sizes, and textures. The images were curated and annotated before being fed into the detection and segmentation pipeline. These curated images are used for testing and validating the performance of our system.

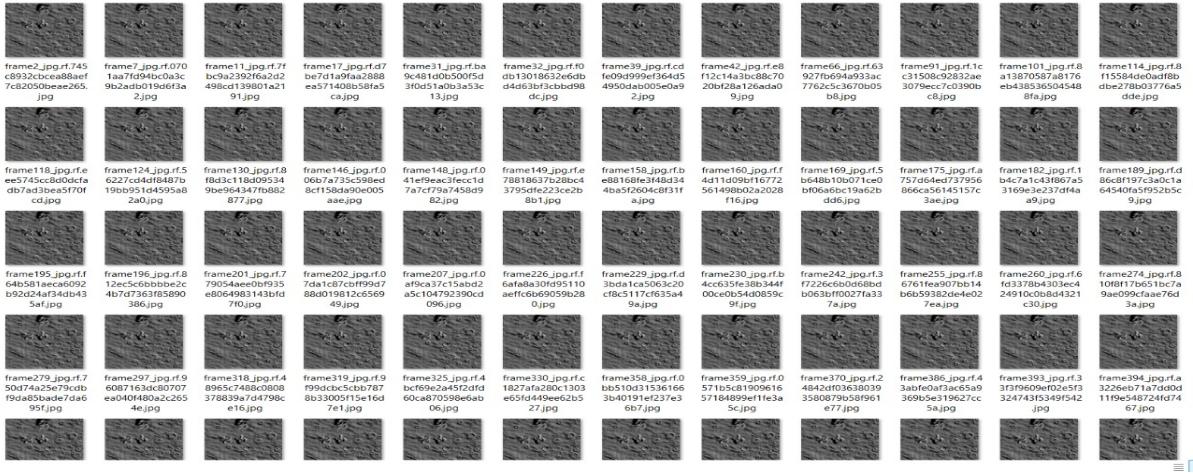


Figure 9.1: OHRC Images

In our system architecture, YOLOv11 is used for detecting craters and boulders, while

the Segment Anything Model (SAM) is employed for precise segmentation of the detected regions. This dual-model setup ensures efficient identification and accurate shape outlining of geological features on the Moon's surface.

This dataset is essential for automating the feature recognition process and reducing manual analysis errors, thereby accelerating scientific research in planetary surface exploration.

9.2 Test Specifications

- The YOLOv11 and SAM models were evaluated using a separate subset of the OHRC dataset from Chandrayaan-2. This test set included high-resolution lunar images with manually annotated craters and boulders to ensure unbiased performance.
- Model performance was assessed using:
 - Precision
 - Recall
 - F1-Score
 - Mean Average Precision (mAP)
 - Intersection over Union (IoU) for segmentation accuracy
- **Test Dataset for System:** The complete system was tested on unseen OHRC images using the trained models. Results were manually verified against ground truth annotations to validate accuracy.

9.2.1 Assumptions and Dependencies

Assumptions

1. **Availability of High-Resolution OHRC Data:** It is assumed that high-resolution images from Chandrayaan-2's OHRC payload are accessible and preprocessed for training, validation, and testing.
2. **Image Quality and Annotation Accuracy:** The performance of the system is dependent on the quality of image annotations. It is assumed that the labeled data used for training and testing are accurate and representative of lunar terrain features.

3. Computational Resources: It is assumed that sufficient GPU-enabled hardware is available to train and deploy YOLOv11 and SAM models efficiently, as both models are computationally intensive.

Dependencies

- 1. Model Framework and Libraries:** The system depends on the successful integration of deep learning libraries such as PyTorch or TensorFlow for model training, evaluation, and deployment.
- 2. Ground Truth Annotations:** The model's accuracy depends heavily on the availability of reliable ground truth data for craters and boulders. Manually annotated datasets or previously validated labels are essential for accurate model supervision.
- 3. Server and Storage Infrastructure:** The system requires reliable infrastructure for storing large volumes of OHRC images, model checkpoints, and result outputs. Cloud or local servers must support fast I/O operations for real-time inference.
- 4. Support for Future Datasets:** Future extensions may rely on datasets from other lunar missions or terrain types. Hence, the system should be adaptable to accept and process newer datasets with minimal changes.

9.3 Performance Measures

9.3.1 Accuracy

The accuracy metrics provides insights into the overall performance of the YOLOv11 and SAM model. The test accuracy indicates the percentage of correctly classified samples.

- Test Accuracy:- The test accuracy is 94%

9.3.2 Precision Curve

The Precision-Confidence Curve shows how precision improves as the confidence threshold increases. It is used to determine how reliably the model makes correct predictions at higher confidence levels. This curve is important for minimizing false positives and improving prediction accuracy.

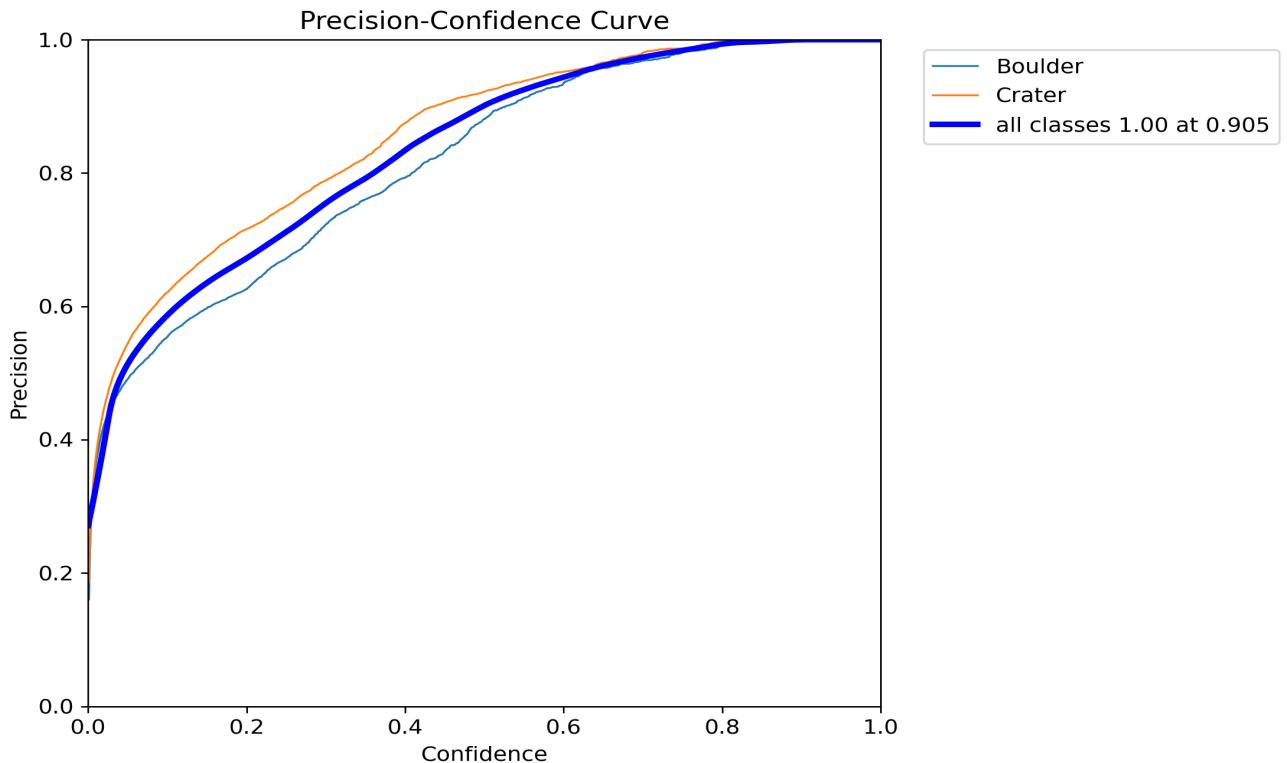


Figure 9.2: Precision Curve

9.3.3 Recall Curve

The Recall-Confidence curve shows how well the model detects craters and boulders at different confidence levels. As the confidence increases, recall usually decreases, meaning fewer objects are detected. This helps us understand how reliable the model is when it is more certain about its predictions.

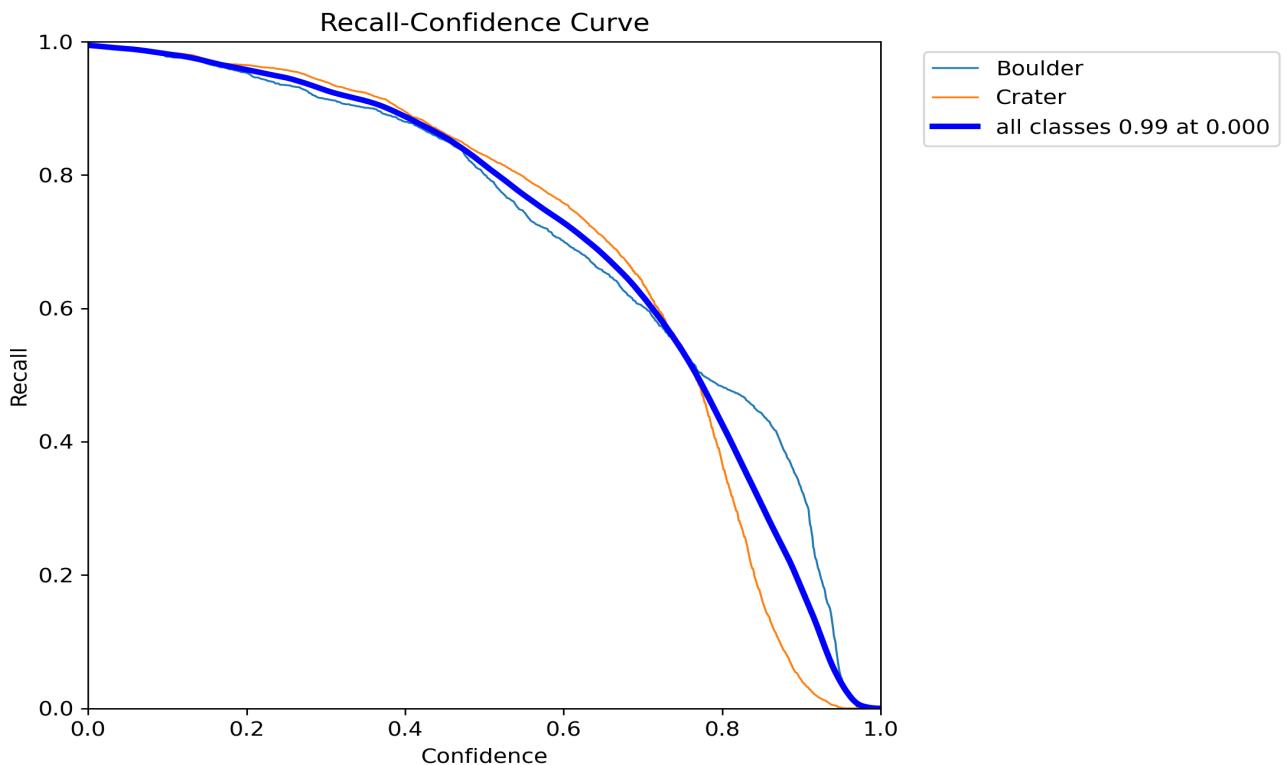


Figure 9.3: Recall Curve

9.3.4 Precision-Recall Curve

The precision-recall curve illustrates the trade-off between precision and recall for different classification thresholds as Figure 9.2. It helps evaluate the model's performance in scenarios where class imbalance exists.

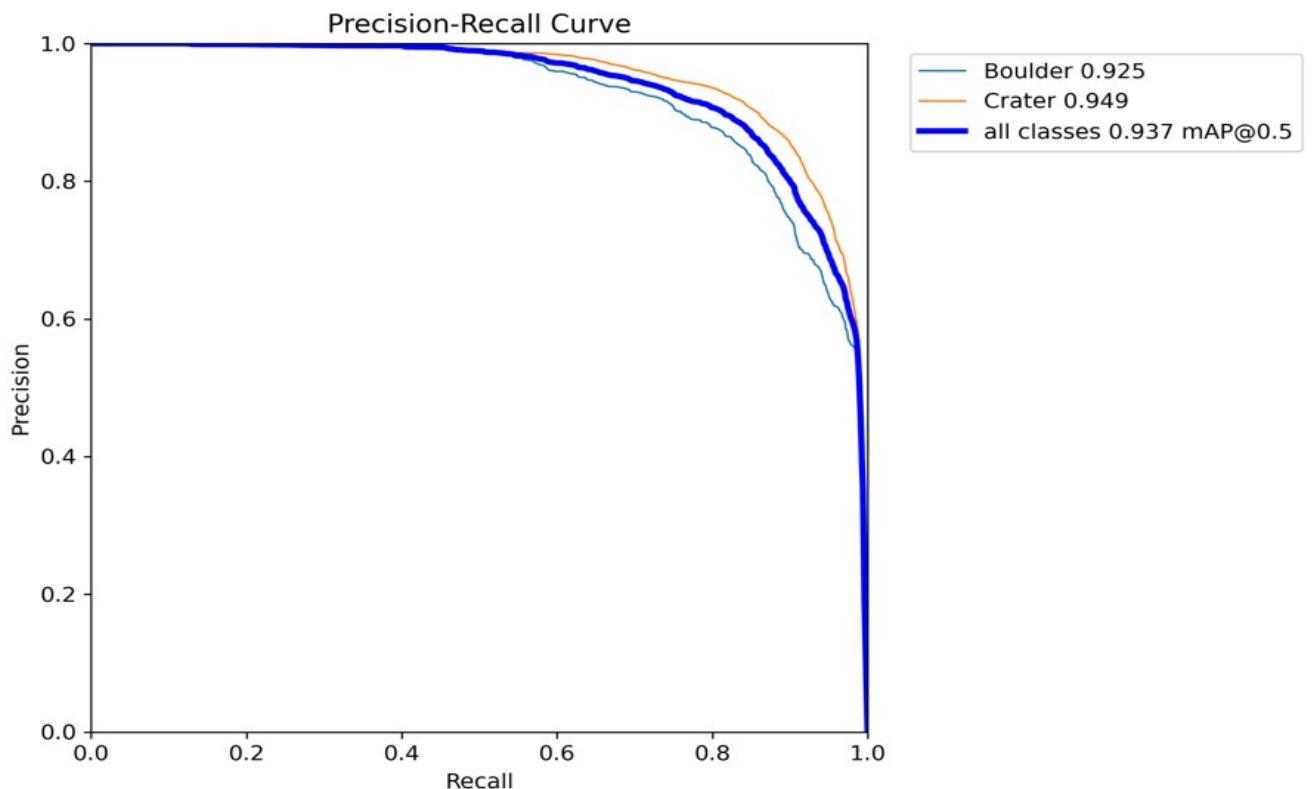


Figure 9.4: Precision-Recall Curve

9.3.5 Confusion Matrix

The confusion matrix provides a tabular representation of actual versus predicted class labels. It allows us to visualize the model's performance in terms of correctly and incorrectly classified instances for each class.

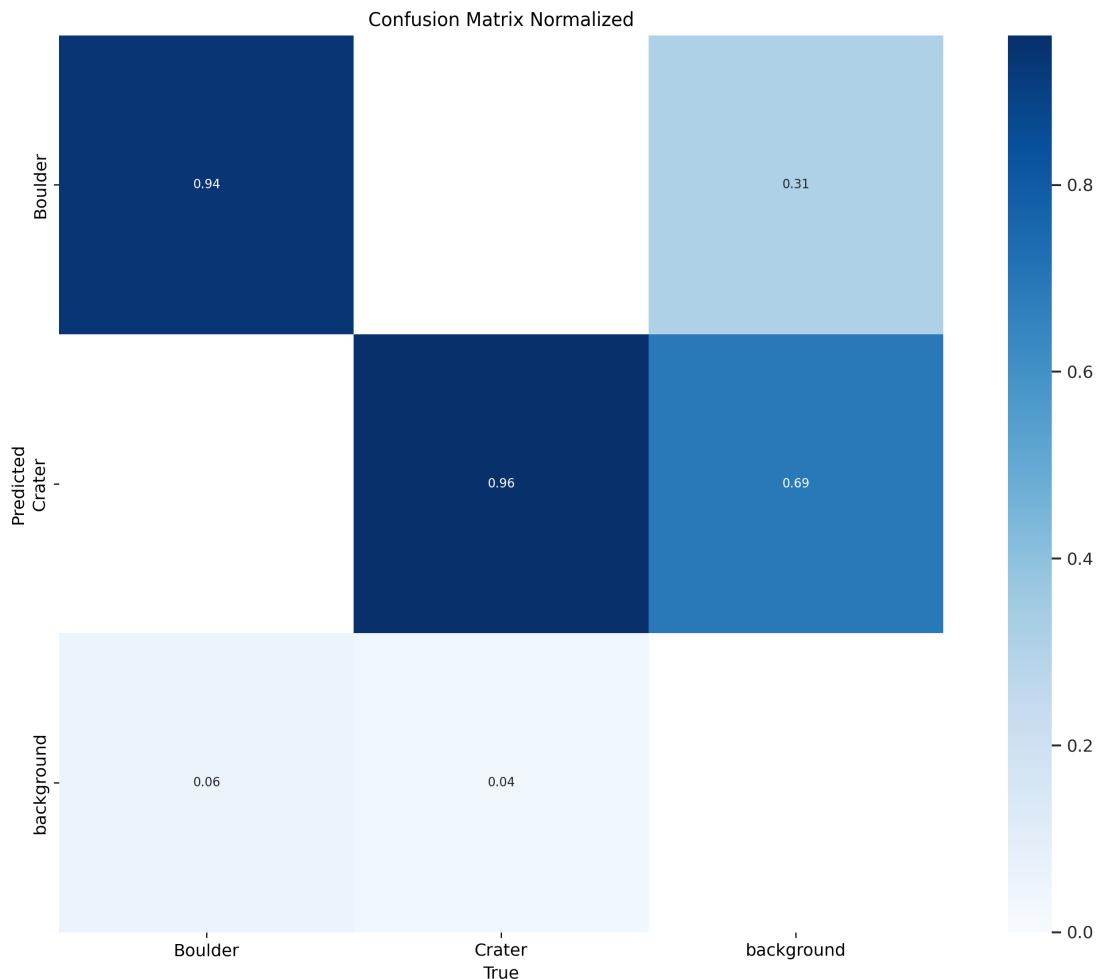


Figure 9.5: Confusion Metrix

9.3.6 Summary and Interpretation

The model demonstrates strong performance in classifying both "Boulder" and "Crater" objects, achieving a mean Average Precision (mAP@0.5) of 0.937. Specifically, the average precision for the "Crater" class is 0.949, slightly outperforming the "Boulder" class, which scores 0.925. The precision-recall curves for both classes are consistently high, indicating reliable detection with minimal trade-off between precision and recall. The confusion matrix supports this, showing a high number of correct predictions for Craters (9266) and Boulders (2834). However, there is noticeable misclassification of background objects, with 3043 and 1354 background instances being incorrectly predicted as Crater

and Boulder, respectively. This suggests the model occasionally confuses background with actual objects, and improving background discrimination could further enhance overall accuracy.

9.4 Experimental Results

9.4.1 Exact Method

Step 1: Dataset Collection High-resolution lunar surface images were obtained from:

- Chandrayaan-2 OHRC mission dataset.
- Open-source lunar terrain datasets.

Images include various geological formations such as craters and boulders under different illumination conditions.

Step 2: Data Preprocessing

- Images were resized to standard dimensions (e.g., 640x640).
- Annotation of craters and boulders was done using tools like Roboflow.
- For SAM input, masks were generated based on bounding box prompts.
- Dataset was split into training (80%), validation (10%), and testing (10%) sets.

Step 3: Crater and Boulder Detection using YOLOv11

- YOLOv11 model was trained on the labeled dataset.
- Configuration:
 - Epochs: 100
 - Batch Size: 32
 - Image Size: 640x640
- Detects objects by predicting:
 - Bounding box coordinates
 - Class probabilities

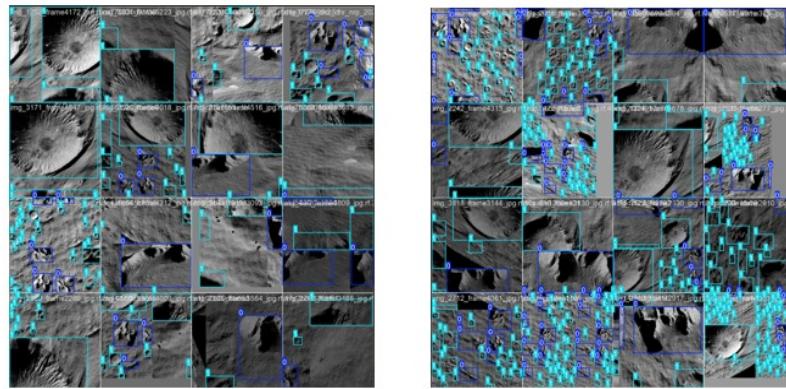


Figure 9.6: Detection Result

Step 4: Segmentation using SAM

- The bounding boxes from YOLOv11 are directly used as prompts for SAM
- SAM generates segmentation masks outlining the exact shapes of detected features.
- This enhances spatial accuracy and allows for detailed surface analysis
- Masks accurately follow crater rims and boulder edges, which is not possible with bounding boxes alone.

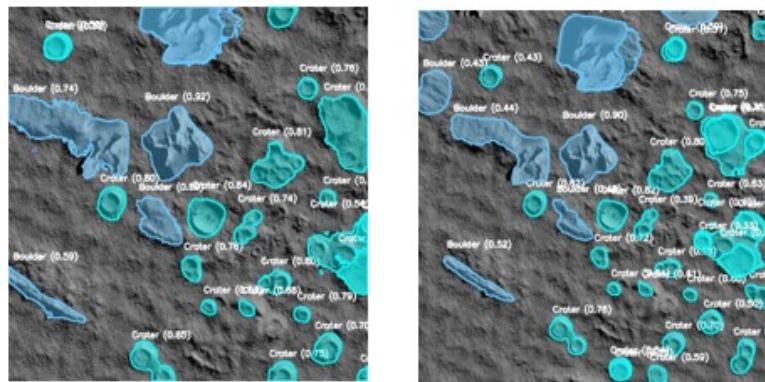


Figure 9.7: Segmentation Result

Step 5: Training and Evaluation

- Evaluation Metrics:
 - Precision, Recall, F1-score.
 - mAP@0.5 and mAP@0.5:0.95 for detection.
 - IoU for segmentation accuracy.

- Tools and Libraries Used:
 - Python, PyTorch, OpenCV.
 - Roboflow.
 - Jupyter Notebook / Google Colab.

epoch	time	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)	val/box_loss	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1	lr/pg2
1	230.111	2.51486	2.16339	2.40433	0.50954	0.05154	0.00614	0.00226	4.0477	20.2827	35.4353	0.07023	0.003307	0.00330729
2	481.963	1.79277	1.45904	1.7092	0.60665	0.56863	0.61995	0.34448	1.63695	1.53291	1.71877	0.04017	0.006581	0.00658086
3	726.784	1.62884	1.29859	1.58011	0.37235	0.59196	0.33407	0.15287	1.84413	2.09013	1.86661	0.01005	0.009794	0.00979443
4	941.37	1.56278	1.22588	1.53696	0.60416	0.52458	0.52019	0.26614	1.75289	1.91742	1.81929	0.00973	0.00973	
5	1162.35	1.50061	1.16442	1.49211	0.63627	0.64037	0.6273	0.33918	1.58427	1.48678	1.642	0.00964	0.00964	0.00964
6	1395.28	1.47344	1.1226	1.46747	0.38166	0.4344	0.35606	0.15109	1.99371	1.76162	1.98759	0.00955	0.00955	
7	1616.53	1.43154	1.07866	1.44373	0.60884	0.6156	0.6668	0.38354	1.56372	1.46015	1.61581	0.00946	0.00946	
8	1837.55	1.40807	1.05261	1.42187	0.64477	0.64671	0.70552	0.41531	1.42573	1.15905	1.52568	0.00937	0.00937	
9	2063.04	1.39889	1.04293	1.41336	0.65234	0.55267	0.60966	0.31411	1.71365	1.35926	1.71153	0.00928	0.00928	
10	2303.47	1.38595	1.04733	1.40572	0.64034	0.5872	0.63846	0.35327	1.59271	1.32999	1.6215	0.00919	0.00919	
11	2513.82	1.35626	0.99656	1.38722	0.77063	0.76539	0.82881	0.50646	1.40471	0.95608	1.51567	0.0091	0.0091	
12	2724.83	1.3608	0.99465	1.38861	0.64258	0.71275	0.68622	0.41262	1.43046	1.14986	1.52456	0.00901	0.00901	
13	2976.55	1.34891	0.99505	1.37459	0.64557	0.59857	0.64195	0.33971	1.59997	1.44877	1.63395	0.00892	0.00892	
14	3209.35	1.34059	0.97324	1.36985	0.6288	0.63707	0.65488	0.3624	1.47729	1.23671	1.53792	0.00883	0.00883	
15	3466.25	1.32074	0.95386	1.3549	0.74021	0.73864	0.78493	0.46704	1.46471	1.03605	1.53813	0.00874	0.00874	
16	3686.31	1.30624	0.94312	1.34758	0.75213	0.73503	0.80519	0.49659	1.31891	1.00678	1.40926	0.00865	0.00865	
17	3911.75	1.30359	0.93776	1.34162	0.77337	0.71744	0.81418	0.49043	1.3789	1.00209	1.4725	0.00856	0.00856	
18	4121.75	1.30543	0.93429	1.34295	0.80182	0.62577	0.76628	0.46797	1.39584	1.07857	1.45476	0.00847	0.00847	
19	4365.67	1.28446	0.9201	1.32497	0.82502	0.71518	0.83314	0.51018	1.34346	0.97299	1.42987	0.00838	0.00838	
20	4577.32	1.28311	0.91633	1.33202	0.79348	0.76816	0.84736	0.53186	1.31478	0.95582	1.42822	0.00829	0.00829	
21	4789.04	1.27636	0.8997	1.32173	0.71728	0.64237	0.73417	0.45524	1.36377	0.99781	1.4337	0.0082	0.0082	
22	5005.84	1.26191	0.89771	1.31742	0.73857	0.70097	0.7748	0.47577	1.38798	1.06381	1.46601	0.00811	0.00811	
23	5224.14	1.27939	0.9029	1.32449	0.73035	0.7169	0.77839	0.47727	1.34002	0.99175	1.42338	0.00802	0.00802	
24	5451.27	1.25172	0.88476	1.30419	0.77689	0.76475	0.8248	0.54048	1.26764	0.88743	1.36053	0.00793	0.00793	
25	5684.46	1.27066	0.89625	1.31112	0.7624	0.79737	0.85791	0.57424	1.23346	0.88781	1.3451	0.00784	0.00784	
26	5916.56	1.24749	0.87966	1.29519	0.78903	0.82532	0.88185	0.5951	1.21232	0.81085	1.32284	0.00775	0.00775	
27	6177.3	1.24904	0.87493	1.29492	0.8245	0.76408	0.87176	0.57686	1.24126	0.84105	1.3521	0.00766	0.00766	
28	6416.43	1.24316	0.86645	1.29296	0.75678	0.78442	0.83456	0.55113	1.23959	0.85903	1.34402	0.00757	0.00757	
29	6638.03	1.24067	0.86514	1.2928	0.79488	0.8186	0.88404	0.60897	1.16152	0.78005	1.29842	0.00748	0.00748	
30	6850.96	1.22001	0.85403	1.28073	0.76789	0.67223	0.7424	0.4775	1.29137	0.99511	1.38634	0.00739	0.00739	

Figure 9.8: Detection Accuracies

9.4.2 Result Analysis

Class	Precision	Recall	mAP@0.5	mAP@0.5:0.95
Boulder	0.859	0.859	0.928	0.776
Crater	0.902	0.871	0.952	0.664

Figure 9.9: Detection Result Matric

The table presents detailed evaluation metrics for the object detection model across two classes: Boulder and Crater. The model achieves a high level of accuracy, as reflected in the precision, recall, and mean Average Precision (mAP) scores.

For the Boulder class, the model attains a precision of 0.859 and a recall of 0.859, indicating a balanced and reliable detection rate. The mAP@0.5 score is 0.928, signifying

strong performance when using a 0.5 Intersection over Union (IoU) threshold. Additionally, the mAP@0.5:0.95 which averages mAP over IoU thresholds from 0.5 to 0.95 in steps of 0.05 is 0.776, demonstrating robust performance across varying localization strictness.

For the Crater class, the model achieves a higher precision of 0.902 and a recall of 0.871, reflecting its slightly better capability in correctly identifying craters. The mAP@0.5 is notably high at 0.952, while the mAP@0.5:0.95 is 0.664, indicating good but slightly less consistent performance across varying IoU thresholds compared to the Boulder class.

Overall, these metrics confirm that the model performs well in detecting both object types, with particularly strong results at the standard IoU threshold of 0.5. However, the drop in mAP when evaluated over a stricter range (0.5 to 0.95) suggests there is room for improvement in precise localization, especially for the Crater class.

Metric	Val
Precision	0.862
Recall	0.865
mAP@0.5	0.940
mAP@0.5:0.95	0.720

Figure 9.10: Evaluation Result

The table presents the overall detection performance of the model. The model achieved a precision of 0.862 and a recall of 0.865, indicating a good balance between correctly identifying objects and minimizing false positives. The mean Average Precision (mAP) at 0.5 IoU is 0.940, demonstrating high accuracy in detecting and classifying objects at the standard threshold. The mAP@0.5:0.95, which evaluates performance over a range of IoU thresholds, is 0.720, reflecting consistent detection capability even under stricter localization requirements. These results confirm that the model performs robustly in object detection tasks.

9.5 Discussions

9.5.1 Limited Availability of Annotated Lunar Data

One of the key limitations of the system is the dependency on a limited number of annotated OHRC (Orbiter High Resolution Camera) images from the Chandrayaan-2 mission. The model's ability to accurately detect and classify lunar craters and boulders heavily depends on the quantity and quality of training data. However, the dataset lacks extensive and diverse annotations, which can lead to overfitting and reduced generalization on unseen lunar terrains. As a result, the system may fail to identify smaller or irregular surface features accurately. To overcome this issue, future enhancements can focus on building larger annotated datasets with varied terrain types, improving the model's adaptability and detection accuracy.

9.5.2 Performance Limitations in Detecting Small and Overlapping Objects

YOLOv11, while effective for real-time detection, faces difficulty in accurately detecting small or partially overlapping objects such as tiny craters or clustered boulders. The bounding box-based detection method may lead to incorrect or merged predictions for closely located surface features. This limitation reduces the precision of crater/boulder localization in highly dense or shadowed regions. Future work may include tuning anchor boxes, adopting multi-scale detection approaches, or integrating attention mechanisms to better capture fine details in high-resolution OHRC images.

9.5.3 Generalization Issues Across Lunar Terrains

The model is primarily trained and tested on specific areas of the moon covered by the Chandrayaan-2 dataset. This regional limitation may affect the model's generalization capability when applied to lunar regions with different geological characteristics or lighting conditions. Changes in illumination due to solar angles and surface albedo can alter the visual appearance of craters and boulders. Therefore, the system may underperform when deployed on unexplored or varied lunar terrains. Incorporating data from a broader range of regions and applying domain adaptation techniques could significantly enhance the model's robustness and reliability.

9.5.4 Computational and Hardware Constraints.

Another limitation of the proposed system lies in its high computational requirements. The combined use of YOLOv11 for detection and SAM (Segment Anything Model) for segmentation demands significant processing power, particularly when applied to large-sized OHRC images. This poses challenges for real-time inference and restricts its deployment on resource-constrained platforms, such as onboard systems in spacecraft or rovers. To address this issue, future implementations can focus on model optimization techniques like pruning, quantization, or the use of lightweight architectures for embedded applications.

Chapter 10

Proposed GUI

The system integrates with high-resolution OHRC lunar images to identify and classify surface features such as craters and boulders using deep learning techniques. The proposed system enhances lunar exploration analysis by automating the detection process using YOLOv11 for object detection and the Segment Anything Model (SAM) for precise segmentation. The user interface serves as the bridge between raw satellite data and detailed scientific insights, allowing for efficient visualization and interpretation of lunar terrain.

The Graphical User Interface (GUI) not only facilitates easy image upload and feature detection but also displays annotated images and summaries. Through seamless design and interactive features, the system significantly enhances user experience, promoting accuracy in navigation, landing site assessment, and geological studies of the Moon. This platform simplifies image analysis while maintaining scientific rigor, making it valuable for both researchers and mission planners.

By incorporating such functionalities, the system presents a robust platform for real-time and post-mission analysis. The GUI offers modularity, high responsiveness, and cross-compatibility through Gradio, allowing future enhancements and integration with more advanced planetary datasets.

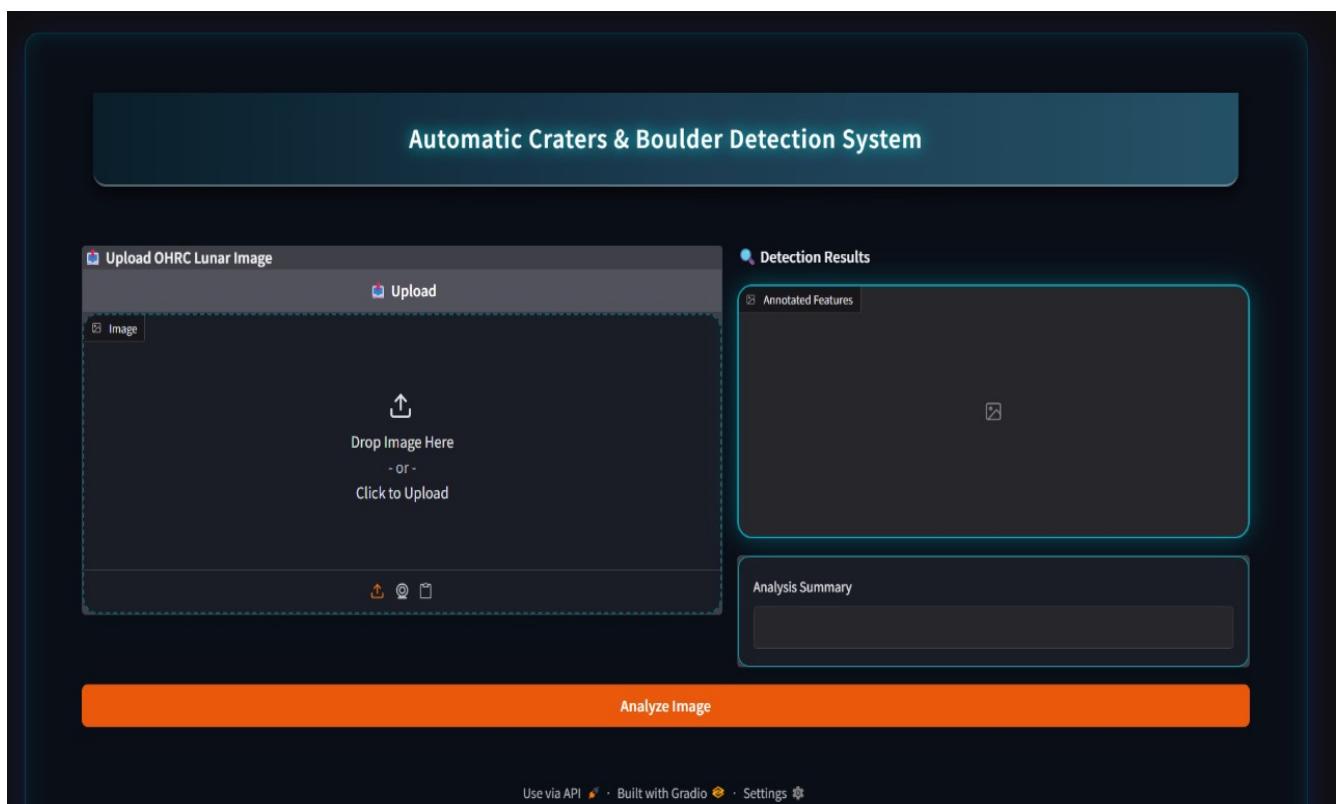
10.1 Proposed GUI

The proposed graphical user interface (GUI) for crater and boulder detection is developed using Gradio, an efficient Python-based library used to build ML-powered web apps. The interface is intuitive and structured to support image upload, analysis, and results interpretation in a stepwise format.

The GUI allows users to upload OHRC images, process them through integrated YOLOv11 and SAM models, and visually interpret detection results. It streamlines the workflow from data ingestion to final output with clarity and ease of use.

The layout and features of the GUI:

- Upload functionality for OHRC lunar images (.jpg, .png, .tif).
- Display of the uploaded OHRC image
- Automatic detection of craters and boulders using YOLOv11
- Segmentation of detected features using SAM
- Visualization of annotated features with bounding boxes or masks.
- Analysis summary with count and types of detected objects
- Modular selection of detection model (YOLOv11 / SAM / Combined).
- Confidence and NMS threshold sliders for fine-tuned detection
- Option to reset or re-upload image for new analysis
- Support for API integration and batch analysis in future enhancements



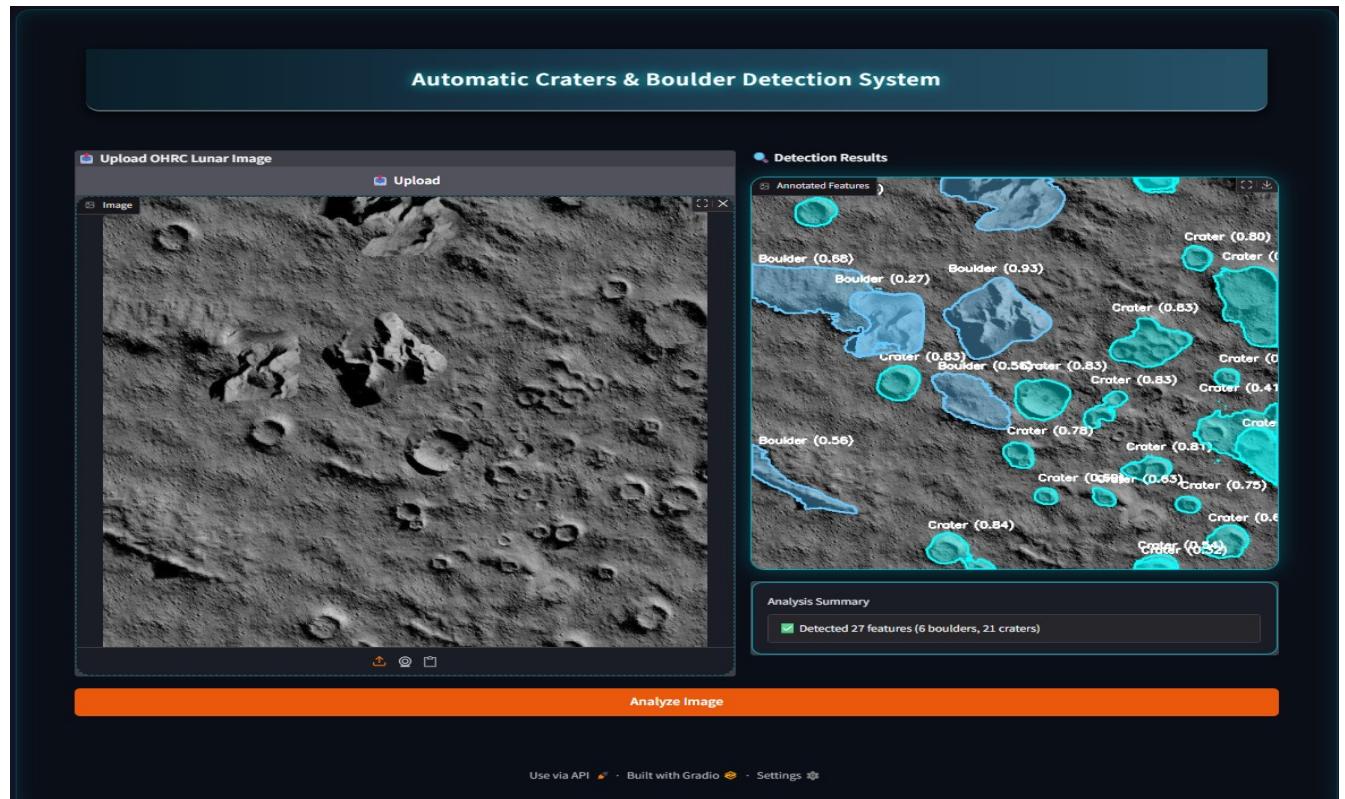


Figure 10.1: GUI Result

10.2 Backend Module

The backend module of the system handles the core functionalities required for accurate crater and boulder detection from OHRC images. This module integrates advanced computer vision and deep learning models—YOLOv11 for object detection and Segment Anything Model (SAM) for precise boundary segmentation of lunar surface features. Upon receiving high-resolution images from the GUI, the backend preprocesses the input using grayscale conversion, resizing, and normalization techniques optimized for lunar imagery. The YOLOv11 model first identifies regions of interest (ROIs) by detecting craters and boulders using bounding boxes with confidence scores. These ROIs are then passed to the SAM module, which performs pixel-level segmentation to delineate the exact boundaries of the detected features. The backend also manages result annotation, such as labeling detected objects with class (crater/boulder), confidence values, and segmentation masks. Additionally, it stores the output in a structured format for further scientific analysis or mission planning. The modular design allows seamless integration of future models or datasets, ensuring adaptability for broader planetary applications.

Chapter 11

Conclusion

The Automatic Detection of Craters and Boulders from OHRC Images using Deep Learning represents a significant advancement in the field of planetary surface analysis and lunar research. By implementing a two-stage deep learning approach using YOLOv11 for object detection and the Segment Anything Model (SAM) for segmentation, the system effectively identifies, localizes, and outlines craters and boulders with high precision. This automation not only reduces manual effort and associated errors but also accelerates the data analysis process, making it suitable for large-scale image datasets from lunar missions.

The experimental results validate the accuracy and reliability of the proposed system, demonstrating its capability to handle high-resolution OHRC images, such as those collected from the Chandrayaan-2 mission and public repositories like Roboflow. By enhancing geological feature recognition, the system contributes to safer navigation, better landing site selection, and deeper scientific understanding of the lunar surface. Moreover, it lays the groundwork for future applications in space research, where intelligent automation is increasingly essential.

Looking ahead, there are several opportunities for future enhancement and expansion of the system. This includes integrating 3D image data for terrain modeling, improving model accuracy with more annotated datasets, enabling real-time edge deployment on satellites or rovers, and extending feature detection to other planetary surfaces. These improvements would further reinforce the system's role in supporting advanced space missions and exploration efforts.

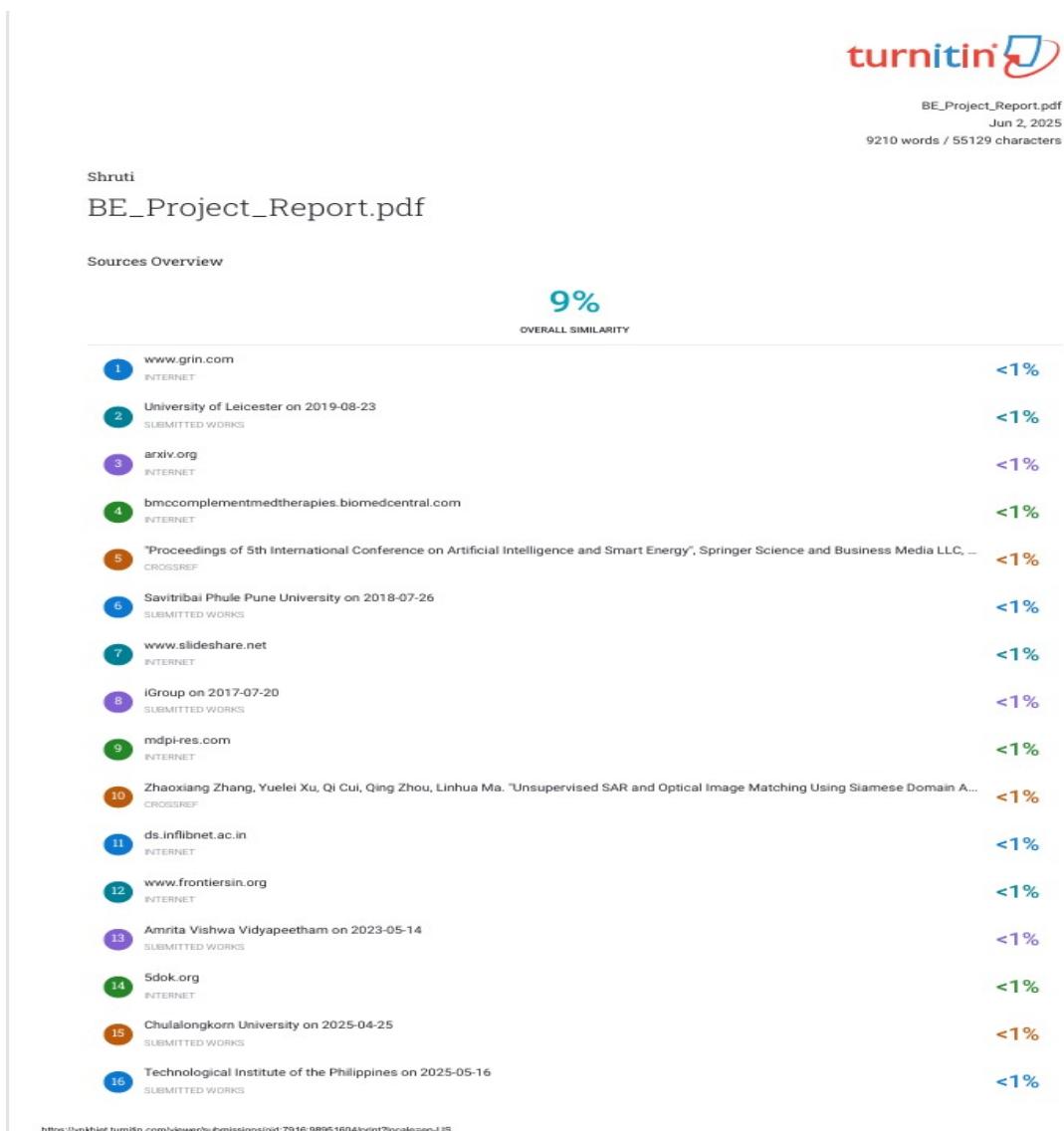
References

- [1] Z. Zhang, Y. Xu, J. Song, Q. Zhou, J. Rasol and L. Ma, "Planet Craters Detection Based on Unsupervised Domain Adaptation," in IEEE Transactions on Aerospace and Electronic Systems, vol. 59, no. 5, pp. 7140-7152, Oct. 2023. [Online].
- [2] A. L. Salih, P. Schulte, A. Grumpe, C. Wöhler and H. Hiesinger, "Automatic crater detection and age estimation for mare regions on the lunar surface," 2017 25th European Signal Processing Conference (EUSIPCO), Kos, Greece, 2017, pp. 518-522. [Online].
- [3] A. Silburt, M. Ali-Dib, C. Zhu, A. Jackson, D. Valencia, Y. Kissin, D. Tamayo, K. Menou, "Lunar crater identification via deep learning," Icarus, Volume 317, 2019. [Online].
- [4] E. Emami, T. Ahmad, G. Bebis, A. Nefian and T. Fong, "Crater Detection Using Unsupervised Algorithms and Convolutional Neural Networks," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 8, pp. 5373-5383, Aug. 2019. [Online].
- [5] R. Alshehhi and C. Gebhardt, "Automated Geological Landmarks Detection on Mars Using Deep Domain Adaptation From Lunar High-Resolution Satellite Images," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 2274-2283, 2022. [Online].
- [6] J. Zhu, J. Liang and X. Tian, "Lunar Impact Crater Detection Based on Yolov7 Using Multi-source Data," 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), Jilin, China, 2023. [Online].
- [7] S. Chatterjee, S. Chakraborty, A. Nath, P. R. Chowdhury and B. Deshmukh, "Near-Real-Time Detection of Craters: A YOLO v5 Based Approach," 2023 International

- Conference on Machine Intelligence for GeoAnalytics and Remote Sensing (MIGARS), Hyderabad, India, 2023. [Online].
- [8] Q. Duan, R. Liu and Y. Liu, "Automatic Lunar Crater Detection Based on DEM Data Using a Max Curvature Detection Method," in IEEE Geoscience and Remote Sensing Letters, vol. 21, pp. 1-5, 2024, Art no. 8001205. [Online].
- [9] C. Ouyang et al., "Causality-inspired single-source domain generalization for medical image segmentation," IEEE Transactions on Medical Imaging, vol. 42, no. 4, pp. 1095–1106, Apr. 2023. [Online].
- [10] P. Li, D. Li, W. Li, S. Gong, Y. Fu, and T. M. Hospedales, "Simple feature augmentation for domain generalization," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 8886–8895. [Online].
- [11] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "VarifocalNet: An IoU-aware dense object detector," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 8510–8519. [Online].
- [12] X. Wang et al., "Robust object detection via instance-level temporal cycle confusion," in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 9123–9132. [Online].
- [13] J. Huang, D. Guan, A. Xiao, and S. Lu, "FSDR: Frequency space domain randomization for domain generalization," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 6891–6902. [Online].
- [14] A. RoyChowdhury et al., "Automatic adaptation of object detectors to new domains using self-training," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 780–790. [Online].
- [15] A. Gao and S. Zhou, "Feature Density Based Terrain Hazard Detection for Planetary Landing," in IEEE Transactions on Aerospace and Electronic Systems, vol. 54, no. 5, pp. 2411-2420, Oct. 2018 [Online].
- [16] Y. Jung, S. Lee and H. Bang, "Digital Terrain Map Based Safe Landing Site Selection for Planetary Landing," in IEEE Transactions on Aerospace and Electronic Systems, vol. 56, no. 1, pp. 368-380, Feb. 2020 [Online].

Appendix A

Plagiarism Report



Appendix B

Base Paper

Z. Zhang, Y. Xu, J. Song, Q. Zhou, J. Rasol and L. Ma, "Planet Craters Detection Based on Unsupervised Domain Adaptation," in IEEE Transactions on Aerospace and Electronic Systems, vol. 59, no. 5, pp. 7140-7152, Oct. 2023

Planet Craters Detection Based on Unsupervised Domain Adaptation

ZHAOXIANG ZHANG 

YUELEI XU

Northwestern Polytechnical University, Xi'an, China

JIANING SONG 

City, University of London, London, U.K.

QING ZHOU 

JARHINBEK RASOL

LINHUA MA

Northwestern Polytechnical University, Xi'an, China

The detection and localization of craters on the Moon and other planets play an essential role in planet landing, spacecraft navigation, and geologic study. Historically, craters detection involves manually measuring the size and placement of craters in surface images. In this article, we propose an automated pipeline named CraterNet to detect craters on the Moon and Mercury from digital elevation map (DEM) images. Firstly, the convolutional neural network (CNN) based object detection model is trained and tested to detect the craters in a supervised manner. To address the domain discrepancy between the source and target data, an unsupervised domain adaptation (UDA) approach combined with domain randomization is suggested. A causal inference-based feature matching (CIFM) approach integrated with histogram matching is then developed to improve the effectiveness of the unsupervised crater detection. The DeepMoon crater dataset and the unsupervised Mercury crater DEMs are introduced in this article to illustrate the applicability and efficacy of the designed method. Results indicate that: 1) The developed approach demonstrates a high

Manuscript received 4 July 2022; revised 6 December 2022 and 13 March 2023; accepted 8 June 2023. Date of publication 16 June 2023; date of current version 11 October 2023.

DOI. No. 10.1109/TAES.2023.3285512

Refereeing of this contribution was handled by J. Gross.

This work was supported in part by The Nature Science Foundation of Shaanxi under Grant 2022JQ-653 and in part by The China University Industry-University-Research Innovation Fund under Grant 2021ITA10022.

Authors' addresses: Zhaoxiang Zhang, Yuelei Xu, Qing Zhou, Jarhinbek Bieke, and Linhua Ma are with the Unmanned System Research Center, Northwestern Polytechnical University, Xi'an 710072, China, E-mail: (zhangzhaoxiang@nwpu.edu.cn; xuyuelie@nwpu.edu.cn; lslczhq@126.com; jarhinbek.r@mail.nwpu.edu.cn; malinhua@nwpu.edu.cn); Jianing Song is with the Department of Electrical and Electronic Engineering, City, University of London, WC1E 7HU London, U.K., E-mail: (jianing.song@city.ac.uk). (Corresponding author: Jianing Song.)

0018-9251 © 2023 IEEE

performance of supervised crater detection on the DeepMoon dataset, with the F1 and AP scores of 0.786 and 0.804, respectively. 2) The detection model is transferred and performs well on the Mercury dataset in which craters are of different sizes and shapes, as the Precision, Recall, and F1 score for ellipse-shape craters are 0.734, 0.773, and 0.753, respectively. 3) The proposed CraterNet outperforms other deep learning-based segmentation and detection models in terms of crater detection and localization scores.

I. INTRODUCTION

Autonomous landing on an extraterrestrial body has become one of the most important technologies for future space missions [1]. A planetary exploration mission can acquire extensive scientific research benefits in a safe manner by using autonomous pin-point landing technology [2]. For a safe touchdown, the position of a lander relative to the landing site should be updated in real-time. Technically, inertial navigation systems are ready for planet landing missions but cannot satisfy the need for relative status updates [3]. Many researchers have introduced terrain-relative navigation technology to compensate for inertial navigation. By providing supplemental data to adjust for drift in the inertial navigation system, terrain-relative navigation can enhance the performance of a spacecraft's location estimate.

A typical terrain relative navigation method relies on sensing several pre-mapped terrain landmarks to estimate the relative pose [4]. One reliable terrain relative navigation landmark for such a mission is a crater. A crater generally has a stable parabolic or truncated cone form and appears as a round or oval outline. As the crater has more distinct and stable characteristics than other landscape markers, various studies have recommended applying craters to conduct relative navigation in planet landing missions.

Thanks to the development of computer vision techniques, visual sensing systems show a great potential for promptly processing and accurately analyzing captured images and videos with computer vision methods for crater detection and spacecraft landing [5]. Previous research has introduced machine learning methods in crater classification and detection. By analyzing raw crater figures with convolutional neural network (CNN) completely, robust crater recognition algorithms have been built, revealing promising findings that have piqued attention. Thus, this article focuses on techniques of CNN to detect craters in planet digital elevation map (DEM) images.

Despite the significant benefit of extracting features accurately, one of the major disadvantages of CNN models is their lack of task generalization capacity [6]. It is often the case that CNN-based models trained via one dataset will fail to meet expectations on another, especially when the datasets are separated by a domain gap. For instance, the performance of a crater detector network trained by the DeepMoon dataset [7] with circle-shape craters degrades when testing it on the Mercury dataset [8] with ellipse-shape craters. While enlarging the training dataset can alleviate this gap in other computer vision tasks, this is not possible on crater detection tasks because collecting crater data is expensive, and there is no publicly available dataset. The

Appendix C

Tools Used

1. Google Colab

- **Purpose:** Used Google Colab to design and deploy the front-end interface using Gradio, enabling a simple and interactive way to visualize model outputs.
- **Benefits:** Colab provides a cloud-based environment that supports seamless development and testing of Gradio interfaces without local installation. It enables quick prototyping of user-friendly UIs to interact with machine learning models. Additionally, it allows real-time sharing and collaboration, making it easier for team members to view and test the interface instantly.

2. Kaggle Notebook

- **Purpose:** Used Kaggle Notebook to develop and train the object detection and segmentation models using YOLOv11 for crater and boulder detection, and SAM (Segment Anything Model) for segmentation tasks.
- **Benefits:** Kaggle provides a powerful cloud-based platform with free GPU and TPU access, which is ideal for training computationally intensive deep learning models like YOLOv11 and SAM. It supports a wide range of pre-installed libraries, saving setup time and making it easier to focus on experimentation and model tuning. Additionally, Kaggle's notebook sharing and version control features enable smooth collaboration and reproducibility of experiments.

3. Gantt Chart

- **Purpose:** Developed a Gantt chart to visualize and manage project tasks, milestones, and Time requirements.
- **Benefits:** Provide a clear overview of the timeline, helping with scheduling tasks, allocating resources, and managing the project efficiently.

4. LaTeX

- **Purpose:** Utilized LaTeX for creating documents and writing reports.
- **Benefits:** Made sure the documents had a professional and consistent format, which is especially important for technical reports. LaTeX is helpful for creating high-quality and well-organized documents.

5. Star UML

- **Purpose:** Used Star UML to design and visualize the system architecture, including use case diagrams and other types of UML diagrams.
- **Benefits:** Helped create clear and detailed visuals of the system structure, making it easier for team members and stakeholders to communicate.

Appendix D

Papers Published/Certificates

D.1 Copyrights



Home | Edit Profile | Change Password |

Welcome Siddhi Kakani (Applicant)

User Home | Options | Logout

Online Services

- e-Filing of Application
- Pending Application for Form XIV
- Status of the Application
- Details of Processing Fee
- Check List
- Workflow
- Upload Discrepancy Reply
- Pre-Hearing Documents
- Hearing Documents
- Correspondence
- Upload Work & Documents
- Re-Upload Work By Applicant
- Make Repayment
- Pending Payment
- Payment History
- Objection Petition
- Download Certificate 

Upload Work & Documents

Work Awaited Applications

No Work Awaited Records

Uploaded Documents

S.No.	Diary No	Work Title	Date of Filing	Type of Work	Uploaded Date	Status	View Document
1	15559/2025-CO/SW	Automated Detection of Craters and Boulders from OHRC Images using Deep Learning	22/04/2025	Computer Software	27/05/2025	Waiting	Software Work to be copyrighted
2	15559/2025-CO/SW	Automated Detection of Craters and Boulders from OHRC Images using Deep Learning	22/04/2025	Computer Software	27/05/2025	Waiting	Others

Home | Terms & Condition | Hyperlink Policy | Privacy Policy | Disclaimer | Feedback | Contact Us | Web Information Manager

D.2 Certificates



Figure D.1: Conference Certificate



Figure D.2: Conference Certificate

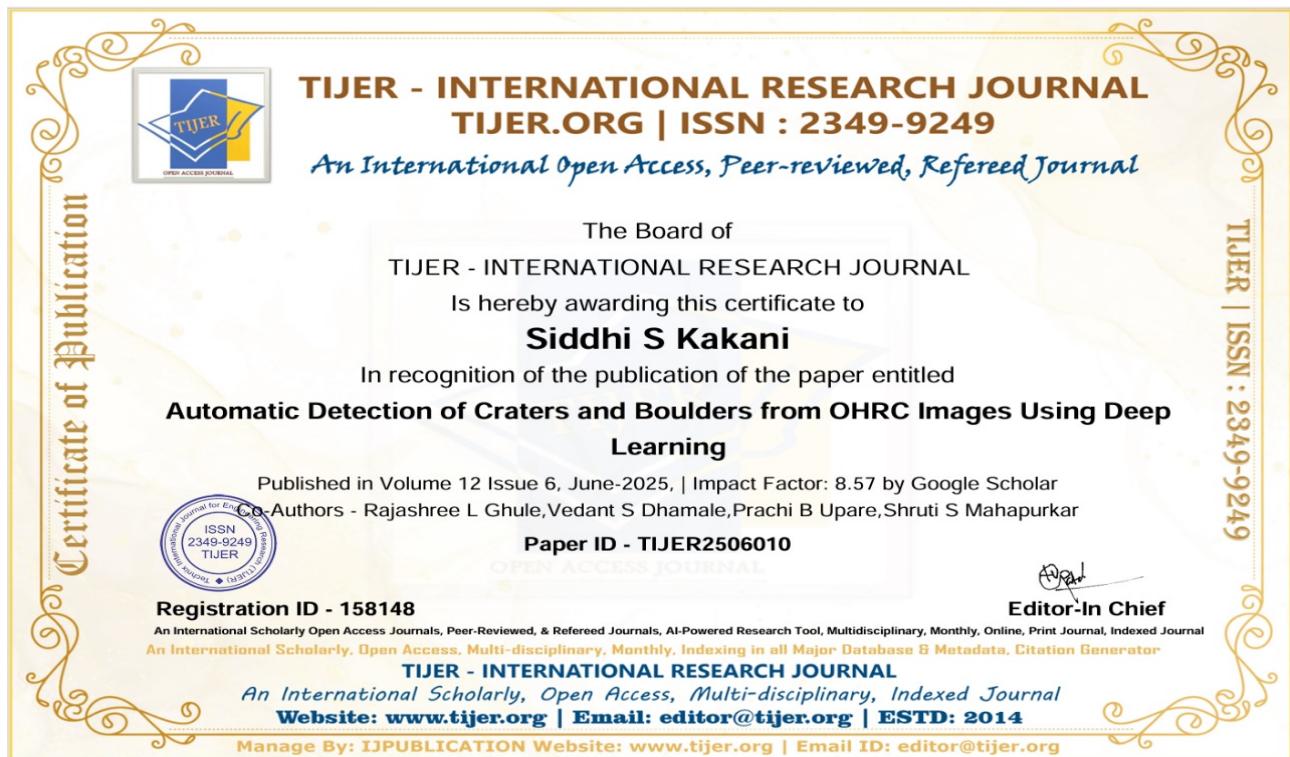


Figure D.3: Journal Certificate



Figure D.4: Competition Certificate

D.3 Paper Published

TIJER || ISSN 2349-9249 || © June 2025, Volume 12, Issue 6 || www.tijer.org

Automatic Detection of Craters and Boulders from OHRC Images Using Deep Learning

1st Siddhi S Kakani, 2nd Rajashree L Ghule, 3rd Vedant S Dhamale, 4th Prachi B Upare,
5th Shruti S Mahapurkar

¹Undergraduate Student, ² Professor at VPKBIET Artificial Intelligence and Data Science, ³Undergraduate Student, ⁴Undergraduate Student, ⁵ Undergraduate Student Artificial Intelligence and Data Science

Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology Baramati, India

Abstract –

The development of an automated system for detecting craters and boulders in high-resolution images from the Orbiter High Resolution Camera (OHRC) using deep learning techniques is crucial for advancing planetary exploration. This study introduces a novel integration of the YOLOv11 object detection algorithm with the Segment Anything Model (SAM) for segmentation, ensuring accurate and efficient identification of geological features. Unlike previous approaches that rely on either object detection or segmentation alone, our method combines real-time object detection with precise segmentation, achieving superior boundary accuracy and detection speed. Leveraging images from the Chandrayaan-2 mission, the proposed system is optimized for scalability and responsiveness in planetary surface analysis. This automation minimizes human involvement, reduces errors, and significantly enhances detection reliability. Furthermore, the ability to process large datasets with improved precision provides new insights into the distribution and morphology of craters and boulders. This contribution supports planetary research by addressing limitations in detection accuracy and efficiency found in prior methodologies, thereby delivering fast and reliable data for scientific analysis and future exploration missions.

Index Terms - YOLOv11, SAM Model, Segmentation, Craters and Boulders Detection, Chandrayaan-2, OHRC, Deep Learning, Planetary Surface Analysis, Automated Detection

INTRODUCTION

Over the past few years, various space missions from organizations like NASA, ISRO, and other international space agencies have gathered vast amounts of data from planetary surfaces, particularly the Moon and Mars. Missions such as Chandrayaan-1 and 2, the Lunar Reconnaissance Orbiter (LRO), and Chang'E have captured high-resolution images that are vital for analyzing the surface features of these celestial bodies. These images provide valuable insights into impact craters and boulders, which are essential for studying geological history, surface evolution, and planning future exploration efforts. Identifying craters and boulders is crucial for understanding a planet's impact history and determining safe landing sites for upcoming missions.

However, the vast amount of data collected from lunar and planetary missions exceeds the capacity of human operators to manually analyze and interpret. Traditional methods and standalone algorithms often fall short in handling the complexity and scale of modern datasets. Existing approaches are limited by their reliance on either object detection or segmentation alone, resulting in inaccuracies, particularly when dealing with overlapping or irregular geological features. Additionally, many methods struggle with maintaining high detection speeds and scalability across diverse planetary terrains. This creates a pressing need for an automated system that combines real-time processing, precision, and adaptability to varied data inputs.

To address this research gap, our study leverages YOLOv11 and the Segment Anything Model (SAM). YOLOv11, a state-of-the-art object detection model, offers significant advantages in real-time detection, providing rapid identification of craters and boulders with high confidence. Its lightweight architecture ensures scalability and efficiency, making it suitable for processing large datasets. Conversely, SAM excels in segmentation, delivering highly accurate boundary delineation even for complex and overlapping objects. By integrating these two models, our approach overcomes the limitations of previous methods, enabling simultaneous detection and segmentation with unprecedented accuracy and speed.

This study demonstrates the integration of YOLOv11 and SAM for processing high-resolution imagery from the Optical High-Resolution Camera (OHRC), a critical instrument onboard Chandrayaan-2. Our system not only enhances the efficiency of analyzing planetary surfaces but also minimizes human intervention, reducing errors and enabling robust scientific insights. The proposed solution holds significant promise for advancing planetary exploration by addressing critical challenges in data analysis and interpretation.