Team Name: Voyager Visionaries
Name of College(s)/University(s): Veer Surendra Sai University of Technology, Burla

Team Members Details:

1. Debadutta Patra
2. Sanghamitra Pradhan
3. Trishna Das
4. Samuel Priyatam Dash

Detailed solution and Approach (250-300 words)

When we talk about crater & boulder, we just don't look at the surface of a planet but we talk about its history and also about how it was formed. Craters speak about geomorphology of the planetary bodies. Since generations the methodologies that we've been using have always resulted in crater detection but with less accuracy rates. But with the integration of newer technologies like resnet which is based on convolutional neural network CNN we can automate the process further while maintaining the same accuracy or even better accuracy . Replacing the current manual annotations which is a very time consuming process with a enhanced model which works on CNN, deep learning model can help us classify the craters based on their efficacy and give us detailed information about various features like colours, textures and shape descriptors.
Or model focuses on implementing the image processing techniques and machine algorithms to locate craters and estimate the ages based on physical features.

Tools and Technology Used (50 words)

We've used Python to train our model, utilizing libraries like OpenCV, NumPy, TensorFlow, Keras, MediaPipe, ResNet, Mask R-CNN, and Matplotlib. Real-time processing uses ARM Cortex processors, while ground-based processing employs high-performance servers with NVIDIA GPUs and high-capacity SSDs/HDDs for storage and dataset management.

Opportunity should be able to explain the following:

- How different is it from any of the other existing ideas?

Currently our way of understanding craters and boulders is by manually analyzing 2000 images which is a time consuming process. Our solution targets exactly this problem.

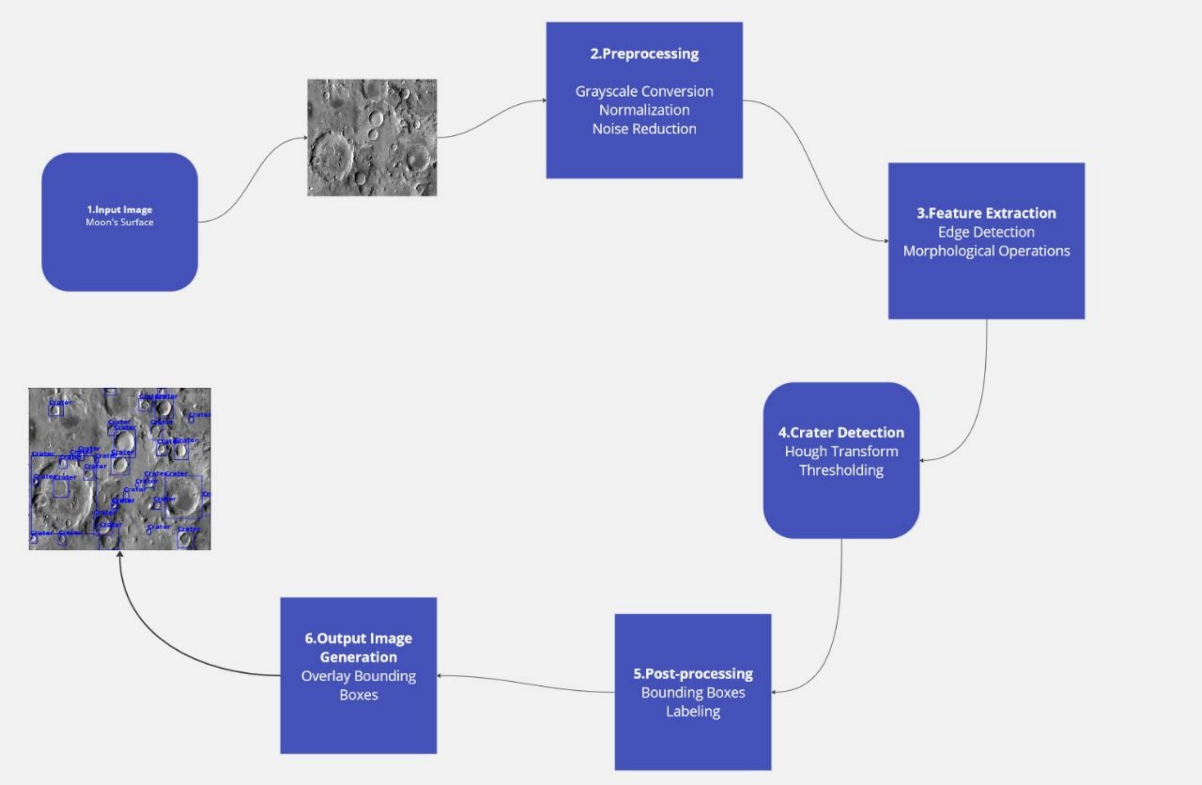
- How will it be able to solve the problem?

Our model captures video frames by using opencv and the frames are preprocessed and labels are drawn bounding the craters and boulders. .It has depth estimation techniques and is based on stereo vision algorithms .

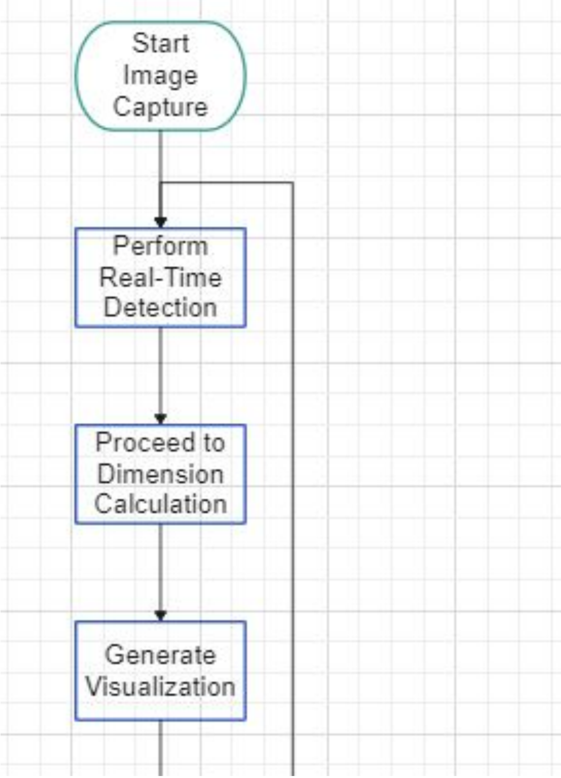
- USP of the proposed solution

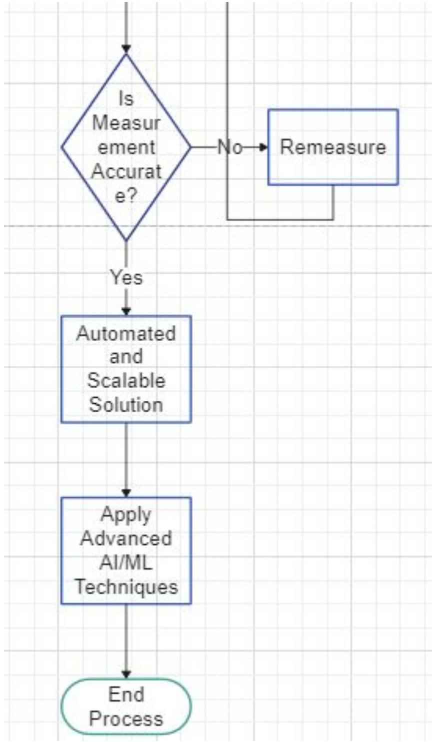
We aim at leveraging leveraging opencv, ResNet and deep learning pipelines for handling large volumes of OHRC data by employing AI models.

Proposed architecture/user diagram

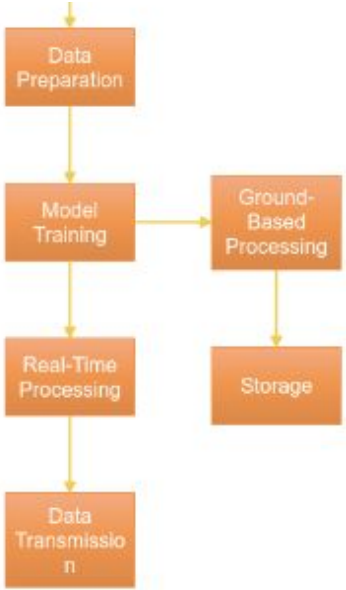


List of features offered by the solution





Process flow diagram or Use-case diagram



Solution Brief (Overall)

The proposed solution employs cutting-edge AI and ML techniques to develop an advanced system for automatic detection, classification, and quantification of lunar craters and boulders from OHRC images. Key components include:

1. **Deep Learning Models**: Utilizes **convolutional neural networks (CNNs)** and **other advanced neural architectures for high-accuracy object detection and segmentation**, trained on extensive lunar image datasets.
2. **Image Processing Techniques**: Applies **noise reduction, contrast enhancement, and edge detection** to improve image quality before analysis, ensuring clear and well-defined features.
3. **Depth Estimation Algorithms**: Incorporates **depth estimation from stereo image pairs to generate 3D topographic maps**, enabling precise measurement of crater depth and boulder size.
4. **Real-Time Processing**: **Integrates these components for real-time analysis**, providing timely and relevant data critical for lunar mission planning and execution.

This system enhances lunar exploration by delivering accurate, automated surface feature detection and measurement, supporting efficient mission planning, hazard assessment, and scientific research.

CODE LOGIC

This code sets up the environment for the Mask R-CNN project, managing dependencies and configuration details. It begins by importing necessary libraries such as `pip`, `logging`, and `pkg_resources`. The script tries to import the `setup` function from `setuptools`, and if that fails, it falls back on `distutils.core`.

A custom function `_parse_requirements` is defined to read the `requirements.txt` file and gather all dependencies listed in it. This function handles different versions of `pip` to ensure compatibility. It returns a list of the required packages as strings.

The variable `install_reqs` holds the parsed list of requirements. If there is an error in parsing, a warning is logged, and the script uses a default empty list to ensure the setup process continues without interruption.

The `setup()` function configures various aspects of the package. It includes the package name ('mask-rcnn'), version (2.1), and the project's URL. Author details and license type are specified for identification and legal purposes. The description briefly explains that this package is for Mask R-CNN, which is used for object detection and instance segmentation.

The `packages` parameter specifies that the "mrcnn" package should be included during installation. The `install_requires` parameter lists dependencies from the `requirements.txt` file. The `include_package_data` flag ensures any additional files mentioned in `MANIFEST.in` are included. The `python_requires` parameter ensures the package is only installed on compatible versions of Python (3.4 and above).

A detailed description provides more information about the package, including its implementation using Python 3, Keras, and TensorFlow. It explains that the model generates bounding boxes and segmentation masks for each object instance in an image, using a Feature Pyramid Network and a ResNet101 backbone.

Classifiers are used to categorize the package, indicating its development status, intended audience, license, programming language, and relevant topics like AI, image recognition, and image segmentation. Keywords like "image instance segmentation" and "object detection" help users find the package more easily.

In essence, this script automates the setup process for the Mask R-CNN project, managing dependencies and ensuring the package is correctly configured for use.

BHARATIYA ANTARIKSH HACKATHON 2024

Innovation partner H2S

इसरो isro

THANK YOU