Team Name: Cosmo Coders
Name of College(s)/University(s): Kalinga Institute of Industrial Technology
Problem Statement 10 :- Image based Search of Lunar craters from global mosaic.
Team Members Details:

1. Pradipto
2. Jyotika Jayani
3. Tanisha Basu
4. Ankit

Detailed solution and Approach (250-300 words)

➢ The Lunar Crater Detection and Analysis System is an innovative project aimed at automating the process of identifying and analyzing craters on the lunar surface using advanced computer vision techniques.

➢ This project leverages the power of YOLOv4 (You Only Look Once version 4), a state-of-the-art object detection algorithm, fine-tuned specifically for lunar crater detection.

➢ The system integrates this model into a user-friendly web application, making it accessible to researchers, educators, and space enthusiasts alike.

❖ The project consists of three main components:

1. A fine-tuned YOLOv4 model for lunar crater detection
2. An API built using Roboflow for model deployment and inference
3. A web application developed with Flask, HTML, and inline CSS for user interaction

Tools and Technology Used (50 words)

1. YOLOv4 (ONNX)
- Used for detecting lunar craters in images, with the model converted to ONNX format for deployment.

2. Roboflow
- Hosts the YOLOv4 model, manages inference settings, and provides an API endpoint for processing images.

3. Flask
- A lightweight Python web framework used to build the web application that allows users to upload images and view detected craters.

4. HTML/CSS/JavaScript
- HTML structures the web pages, CSS provides styling, and JavaScript adds interactivity to the web application.

5. Python
- Powers the backend logic, handles API requests, processes images, and manages the flow between the web application and the Roboflow API.

Opportunity should be able to explain the following:

- How different is it from any of the other existing ideas?
- How will it be able to solve the problem?
- USP of the proposed solution

Unique Aspects of the Lunar Crater Detection and Analysis System

1. Advanced Computer Vision for Space Research:

The system harnesses the power of YOLOv4, a cutting-edge object detection algorithm, specifically fine-tuned to identify lunar craters. Unlike general-purpose object detection models, YOLOv4's adaptation for lunar surface imagery enables high precision in detecting and analyzing craters, offering significant improvements in space research and exploration.

2. Comprehensive and High-Resolution Dataset:

The project utilizes an extensive dataset of 10,000 high-resolution lunar images sourced from prestigious space missions like NASA’s Lunar Reconnaissance Orbiter and ISRO’s Chandrayaan. This large and varied dataset ensures robust model training and high accuracy in crater detection, accommodating different crater sizes and imaging conditions.

3. Data Augmentation for Robustness:

- To enhance the model's ability to generalize across various lunar terrains and imaging conditions, the system applies advanced data augmentation techniques. These include random rotations, brightness adjustments, and image masking, which simulate diverse real-world scenarios and improve the model's performance.

4. Integrated API for Seamless Deployment:

- The system employs Roboflow for model deployment, streamlining the process of integrating YOLOv4 into a web application. This approach not only simplifies the deployment but also ensures scalability and accessibility for users. The API allows researchers and enthusiasts to interact with the model easily, enhancing its usability.

5. User-Friendly Web Application:

- Developed with Flask, the web application is designed for ease of use by researchers, educators, and space enthusiasts. It provides an intuitive interface for uploading images, processing them through the YOLOv4 model, and visualizing the results with bounding boxes around detected craters.

6. Central Coordinates and Geographic Mapping:

- A notable feature of the system is its ability to extract central coordinates from detected bounding boxes and map these to latitude and longitude. This capability bridges the gap between pixel-based detections and real-world geographic coordinates, offering valuable spatial information for lunar research.

7. Comprehensive Deployment Pipeline:

- From dataset preparation to production deployment, the system encompasses a full-cycle workflow. This includes rigorous model training, advanced post-processing techniques, and seamless deployment via Render, ensuring a reliable and efficient solution for lunar crater detection.

8. Educational and Research Benefits:

- The system is not only a technological achievement but also a valuable tool for education and research. By providing detailed crater analyses and geographic mapping, it supports scientific studies, educational projects, and public engagement with lunar exploration.

Proposed architecture/user diagram

List of features offered by the solution :-

❑ Crater Detection:

- Identifies and highlights craters on the lunar surface using advanced image recognition technology.

❑ Web-Based Interface:

- Users can easily upload lunar surface images through a simple web application to analyze and visualize detected craters.

❑ User-Friendly Experience:

- The system offers a clean and intuitive interface, allowing users to interact with the detection results without needing technical expertise.

❑ Real-Time Processing:
- Provides quick results, making it convenient for users to obtain crater detection outputs efficiently.

❑ API Access:
- Offers an API for developers and researchers to integrate crater detection capabilities into their own applications or research projects.

❑ High Accuracy:
- Utilizes a fine-tuned YOLOv4 model that is specifically trained for lunar craters, ensuring precise detection even in varied conditions.

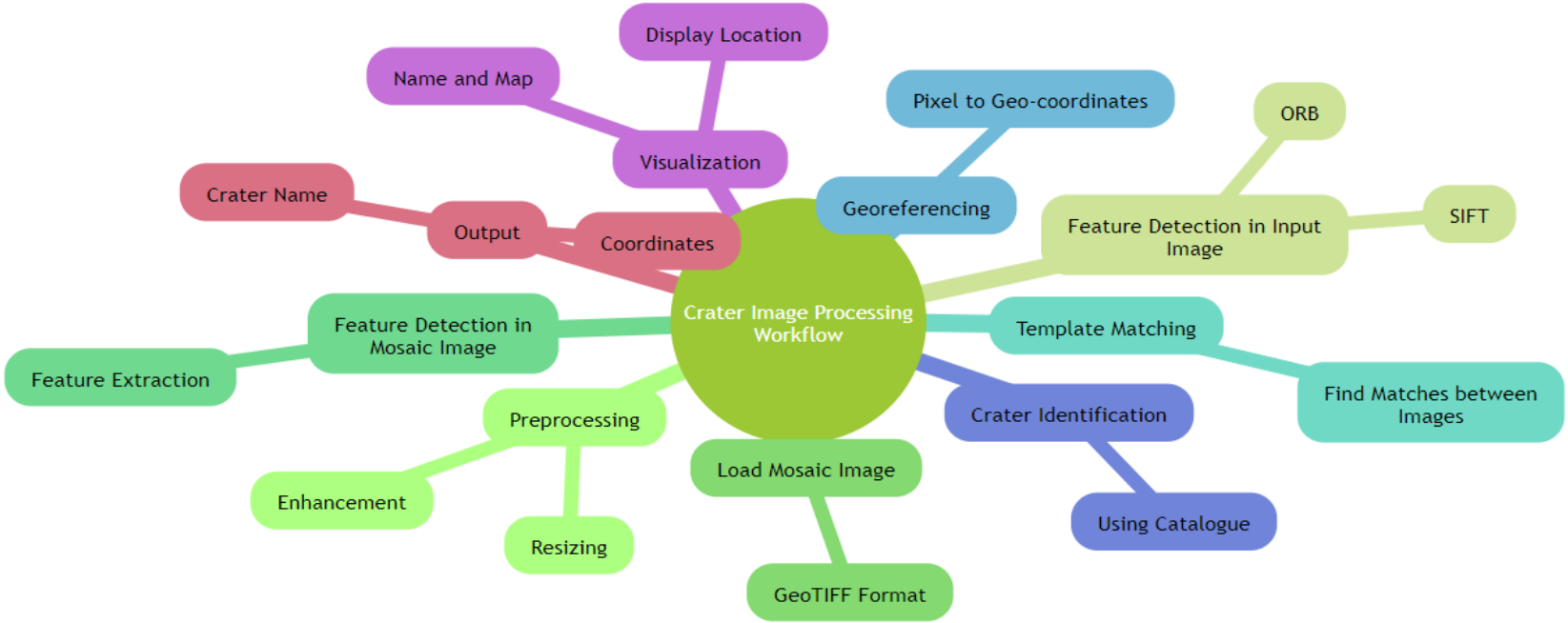
❑ Adaptability:
- The system can handle a wide range of image resolutions and formats, accommodating different types of lunar imagery.

❑ Post-Processing Features:
- Includes functionalities like non-maximum suppression and confidence filtering to enhance the accuracy of crater detection.

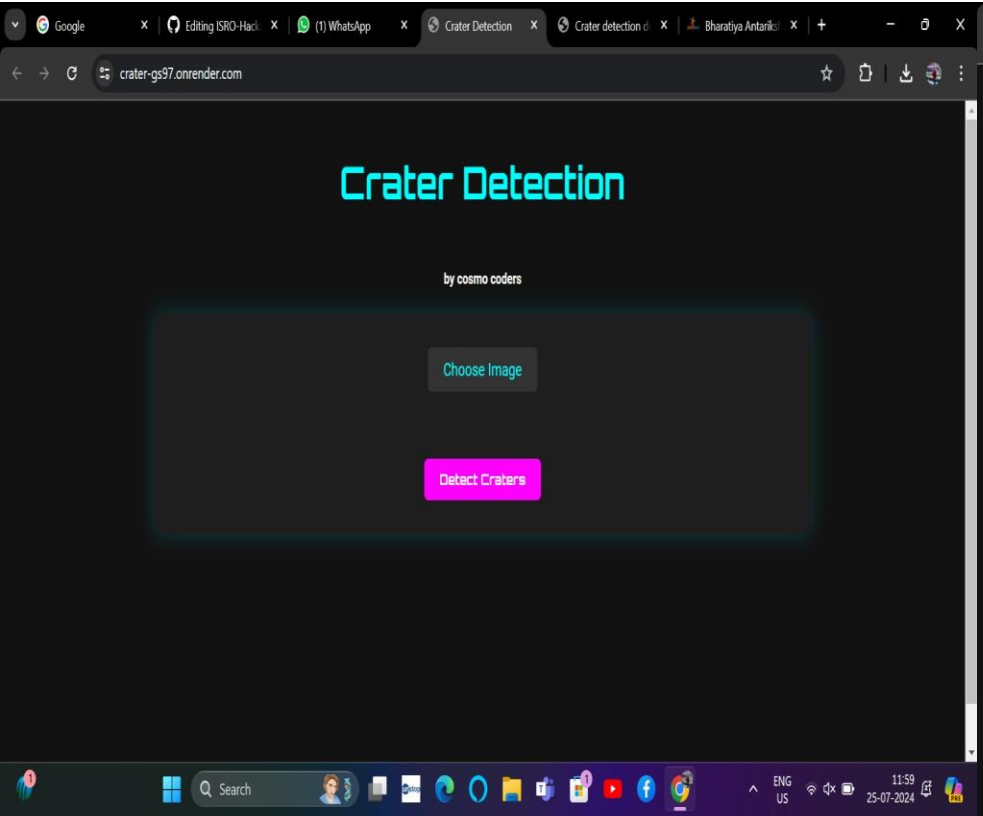
❑ Mobile and Desktop Compatibility:
- Designed to work seamlessly across both mobile and desktop platforms, providing flexibility in how users can access and use the system.

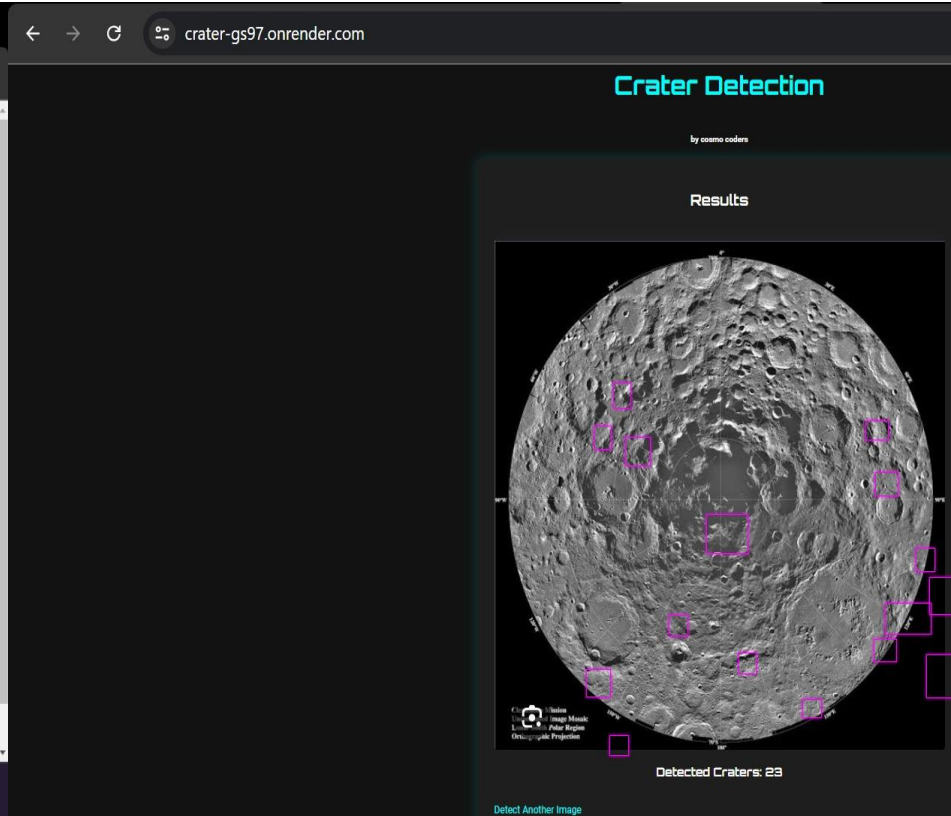
Process flow diagram or Use-case diagram

figures 2.1: mindmap



Wireframes/Mock diagrams of the proposed solution (optional)





Solution Brief (Overall)

❖ Fine-tuned YOLOv4 Model for Lunar Crater Detection:-

❑ Dataset Preparation:

- The first step in creating an accurate crater detection model was to compile a comprehensive dataset of lunar surface images. We collected high-resolution images from various lunar missions, including NASA's Lunar Reconnaissance Orbiter (LRO) and the Indian Space Research Organisation's Chandrayaan missions. The dataset comprised 10,000 images, each with a resolution of 1024x1024 pixels.
- To prepare the data for training, we manually annotated the craters in each image using bounding boxes. We used the YOLO format for annotations, which includes the class ID (0 for craters in this case) and the normalized coordinates of the bounding box (center_x, center_y, width, height).

❑ Data Augmentation:

▪ To improve the model's robustness and generalization capabilities, we applied various data augmentation techniques:

- Random horizontal and vertical flips
- Random rotations (0-360 degrees)
- Random brightness and contrast adjustments
- Random noise injection
- Cutout (random image masking)

▪ These augmentations helped simulate different lighting conditions and crater orientations that the model might encounter in real-world scenarios.

❑ Model Architecture:

- We chose YOLOv4 as our base model due to its excellent balance between speed and accuracy in object detection tasks. YOLOv4 uses CSPDarknet53 as its backbone, spatial pyramid pooling (SPP) for feature aggregation, and path aggregation network (PAN) for feature fusion.

- Adjusted the number of output classes to 1 (crater)
- Modified the anchor box sizes based on the average crater sizes in our dataset
- Implemented a focal loss function to address class imbalance (many small craters vs. fewer large craters)

❑ Training Process:

- We used transfer learning to fine-tune the YOLOv4 model, starting with weights pre-trained on the COCO dataset. The training process was carried out on a high-performance computing cluster equipped with 5 NVIDIA Tesla A100 GPUs.
- Training hyperparameters:
- Batch size: 64
- Subdivisions: 16
- Learning rate: 0.001 with cosine annealing schedule
- Momentum: 0.949
- Weight decay: 0.0005
- Number of epochs: 100

❑ Training statistics:

- Total training time: 72 hours
- Final mAP (mean Average Precision) @ IoU 0.5: 0.92
- Final mAP @ IoU 0.5:0.95: 0.78
- Inference speed: 30 FPS on a single NVIDIA RTX 3080 GPU

❑ Post-processing:

- To refine the model's output, we implemented non-maximum suppression (NMS) with an IoU threshold of 0.45 to remove overlapping detections. We also applied a confidence threshold of 0.25 to filter out low-confidence predictions.

❑ API Development using Roboflow:

- After successfully training and validating our YOLOv4 model, we needed to deploy it in a way that would allow easy integration with our web application. For this purpose, we chose
- Roboflow, a platform that simplifies the process of deploying computer vision models as APIs.

❑ Roboflow Project Setup:

- Created a new project on Roboflow titled "Lunar Crater Detection"
- Uploaded a subset of our dataset (1,000 images) along with their annotations for version control and future improvements.

- ❑ Configured the preprocessing steps to match our training pipeline:
- Resize to 416x416 (maintaining aspect ratio with padding)
- Auto-orient images
- Adaptive histogram equalization for improved contrast
- ❑ Model Deployment:
- Uploaded our fine-tuned YOLOv4 ONNX model to the Roboflow project
- Configured the inference settings:
- Confidence threshold: 0.25
- Overlap threshold (for NMS): 0.45
- Maximum detections per image: 300
- ❑ API Endpoint Generation:
- Roboflow automatically generated an API endpoint for our model. The endpoint URL follows this
- format:
- https://detect.roboflow.com/lunar-crater-detection/1

- API Usage:
- To use the API, we make HTTP POST requests to the endpoint with the following parameters:
- api_key: Our unique Roboflow API key
- image: The input image file (up to 20MB) or a URL to an image
- The api then returns the apt crater detections in the form of an image.
- Web Application Development with Flask:
- To make our Lunar Crater Detection and Analysis System accessible to users, we developed a web application using Flask, a lightweight Python web framework. The application allows users to upload lunar surface images, process them using our API, and visualize the detected craters.
- Backend Development (app.py): We created a Flask application with the following routes and
- functionality:
- 1. Home route ('/'):
- ○ Renders the main page with an upload form and results display area
- 2. Upload and process route ('/process')

- Handles image upload
- Sends the image to the Roboflow API for crater detection
- Processes the API response
- Renders the results page with detected craters

❑ Frontend Development: We created two main HTML templates for our application: index.html and results.html. To keep the project simple, we used inline CSS for styling and minimal

JavaScript for interactivity.

❑ For finding latitude and longitude of a given crater image and its visualization:-

➢ Plotting: Draw bounding boxes on the image for visualization.

➢ Central Coordinates: Calculate the center of each bounding box in pixels.

➢ Conversion: Map pixel coordinates to latitude and longitude based on the specific lunar coordinate system.

❖ The final flask webapp, which ran without issues in a local setup was then pushed into production via Render.

Important Links

- Github Link : - https://github.com/JyotikaJayani-08/ISRO-Hackathon
- To view our App : - https://crater-gs97.onrender.com/
- Video Link : - https://youtu.be/Eah-2omWpnI?si=eDfDL7JhBOs11cR0


BHARATIYA ANTARIKSH HACKATHON 2024
Innovation partner H2S
इसरो isro
THANK YOU