Team Name:
Name of College(s)/University(s): Indian Institute of Technology ,Ropar

Team Members Details:

1. M. Sai Yaswanth
2. P. Jayavardhan
3. Y.G.S.V Ashish
4. S.Madhusudhan

PROBLEM STATEMENT 10

Image based Search of Lunar craters from global mosaic.

UNDERSTANDING THE PROBLEM STATEMENT:

This project focuses on lunar imagery processing using OpenCV and QGIS tools to identify and locate lunar craters, aiding scientific research and navigation. It tackles the challenge of matching high-resolution images from Chandrayaan-2's TMC with a global mosaic from LRO's WAC, despite spatial resolution differences (5m vs. 100m). The developed software aims to accurately pinpoint the geographical coordinates of craters and possibly identify them using a lunar crater catalogue.

THE SOLUTION :

To the search the given input image chip from the Chandrayaan-2's TMC (5m resolution) with 100m resolution of LRO WAC global mosaic (100m resolution) the TMC-2's(5m res) input image chip is template matched to match its pixel data composition with the 100m global mosaic for comparison, this template matching is done using OpenCV. Post the template matching the pixel composition data of input image chip is searched over the entire LRO WAC global mosaic (100m resolution).

PROBLEM STATEMENT 10

Image based Search of Lunar craters from global mosaic.

The process of searching for the input image chip's pixel information within the 100m global mosaic involves comparing each pixel's data stored in a 4-byte float32 format of input image with Global Mosaic. The challenge arises because the size of the global mosaic is 5.5GB and contains over 6 billion pixels, estimated based on the lunar circumference (10,921 km) and lunar pole-to-pole distance. Given that each pixel requires 4 bytes, comparing 6 billion pixels would need around 24GB of memory just to prepare the LRO global mosaic data for comparison. Additionally, the pixel information from the Chandrayaan-2 TMC image chip, with a 20 km swath and variable lengths, can use up to 3 GB of memory. To make this model run on less resources and compatible to run on our laptops for proof of concept we processed the LRO's data into chunks and began searching pixel information in input from individual chunks of global mosaic, now only chunks of 1GB of are processed for comparison of pixel information which enables to run this model on less resources. And the time constraint will be optimized in time before submission.

Using OpenCV, we identify the best matches among six object identification methods. The coordinates of these matches are printed and cross-referenced with the lunar crater database, potentially revealing the crater's name. The acceptable margin of distortion in pixel information comparison is determined by training an AI/ML CNN models.

Tools and Technology Used :

- Programming Languages & Libraries:

Python with OpenCV for image processing, NumPy for numerical operations, Matplotlib for visualization, and scikit-image for image enhancement.

- Image Processing Techniques:

Rescaling (using Area, Lanczos, Bicubic, Bilinear, and Nearest methods), illumination normalization (CLAHE), and feature detection (ORB and FLANN-based matching).

- Template Matching & Feature Detection:

Multi-scale template matching for crater identification and feature matching with ORB and FLANN for similarity analysis between crater images and mosaics.

- Geospatial Data Handling: GDAL for converting pixel coordinates to geographic coordinates.

These tools and methods address the resolution differences between Chandrayaan-2 and LRO images, ensuring accurate crater detection and location extraction.

Opportunity should be able to explain the following:

- How different is it from any of the other existing ideas?
- 1. Resolution Alignment: The solution addresses the issue of different image resolutions by rescaling high-resolution Chandrayaan-2 images to match the lower-resolution LRO mosaics. This alignment ensures accurate template matching despite the resolution differences.
- 2. Enhanced Detection: Advanced feature detection techniques (ORB and FLANN) and multi-scale template matching are used to improve crater identification accuracy. These methods are robust to variations in image conditions and scales, leading to more precise crater localization.
- 3. Geospatial Accuracy: GDAL is employed to convert pixel coordinates to geographic coordinates, ensuring accurate extraction and comparison of crater locations with known geospatial data.

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

Illumination Normalisation :

- We used CLAHE Method for illumination normalisation which is the more efficient and we used Use the tile-based processing approach with CLAHE. This allows you to process the large file in manageable chunks.
- It offers a good balance between effectiveness and computational efficiency, especially when implemented with tiling for large images.

Resolution Matching:

- We use Bicubic Interpolation (cv2.INTER_CUBIC) Sharper than bilinear, preserves more detail for resolution matching

Template Matching:

- We used ORB (Oriented FAST and FLANN) for Feature matching methods which is Fast and efficient than SIFT, SURF, AKAZE

Large Image Processing:

- The large image is processed in overlapping tiles to manage memory usage and allow for parallelization.
- Detections from each tile are combined and then clustered to produce the final set of crater detections.
- Choosing Best Methods and function Gives More Efficent and Effective Solution for the problem

SIZE HANDLING OF PIXEL DATA OF LRO MAP

Size of pixes are handelled by :

```
max_pixels = max_memory_usage // bytes_per_pixel
window_size = int(np.sqrt(max_pixels))
window_size = max(window_size, max(height_crater, width_crater))
window_size = (window_size, window_size)
```

This states that,
The code calculates a memory-efficient window size for image processing by dividing available memory by bytes per pixel to determine max_pixels. It then sets the window size to the square root of max_pixels, ensuring it's large enough to handle the crater image dimensions. This approach optimizes memory usage and processing efficiency during template matching.

HOW TEMPLATE MATCHING HELP ?

```
methods = [ cv2.TM_CCOEFF, cv2.TM_CCOEFF_NORMED, cv2.TM_CCORR, cv2.TM_CCORR_NORMED,
cv2.TM_SQDIFF, cv2.TM_SQDIFF_NORMED ]
 for method in methods:
     result = cv2.matchTemplate(moon_image_copy_temp, crater_image_resampled, method)
     min_val, max_val, min_loc, max_loc = cv2.minMaxLoc(result)
```

Above pseudo code tells :

Template matching plays a crucial role in identifying and locating a specific feature within a larger image by comparing the template image against various sections of the search image. This process involves calculating similarity scores to find the best match. We are checking all 6 methods in computer vision and we are display the rectangular output matching boxes and finalizing the best box.

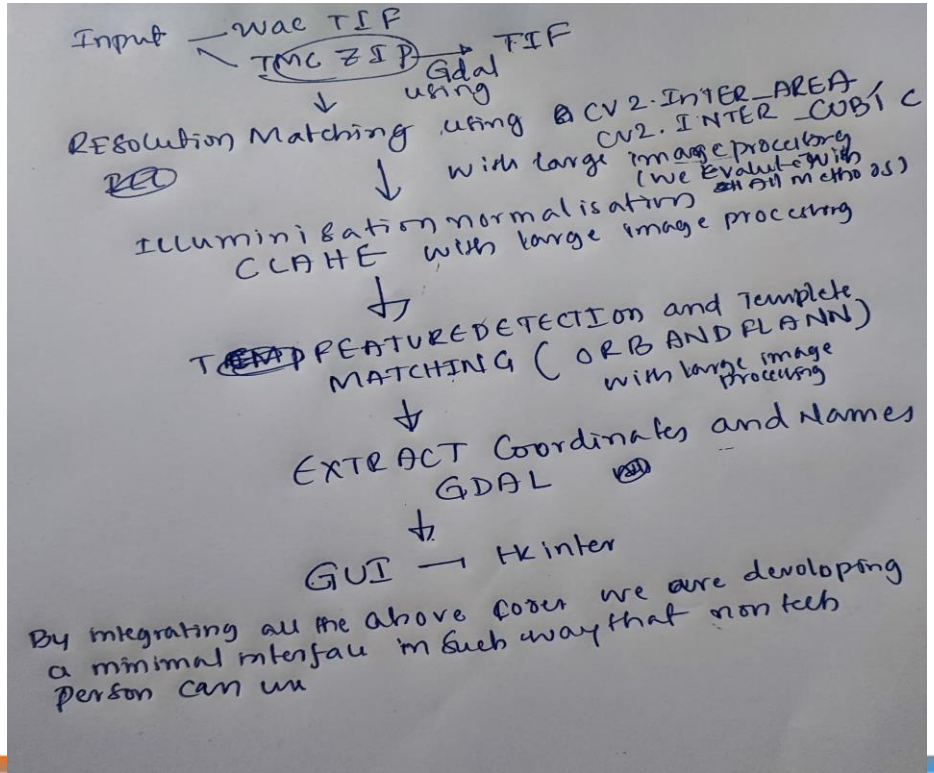
How RESAMPLING works ?

new_width = int(crater_image.shape[1] * scaling_factor_x)
new_height = int(crater_image.shape[0] * scaling_factor_y) crater_image_resampled = cv2.resize(crater_image, (new_width, new_height), interpolation=cv2.INTER_LINEAR)
We have used other three methods too to test

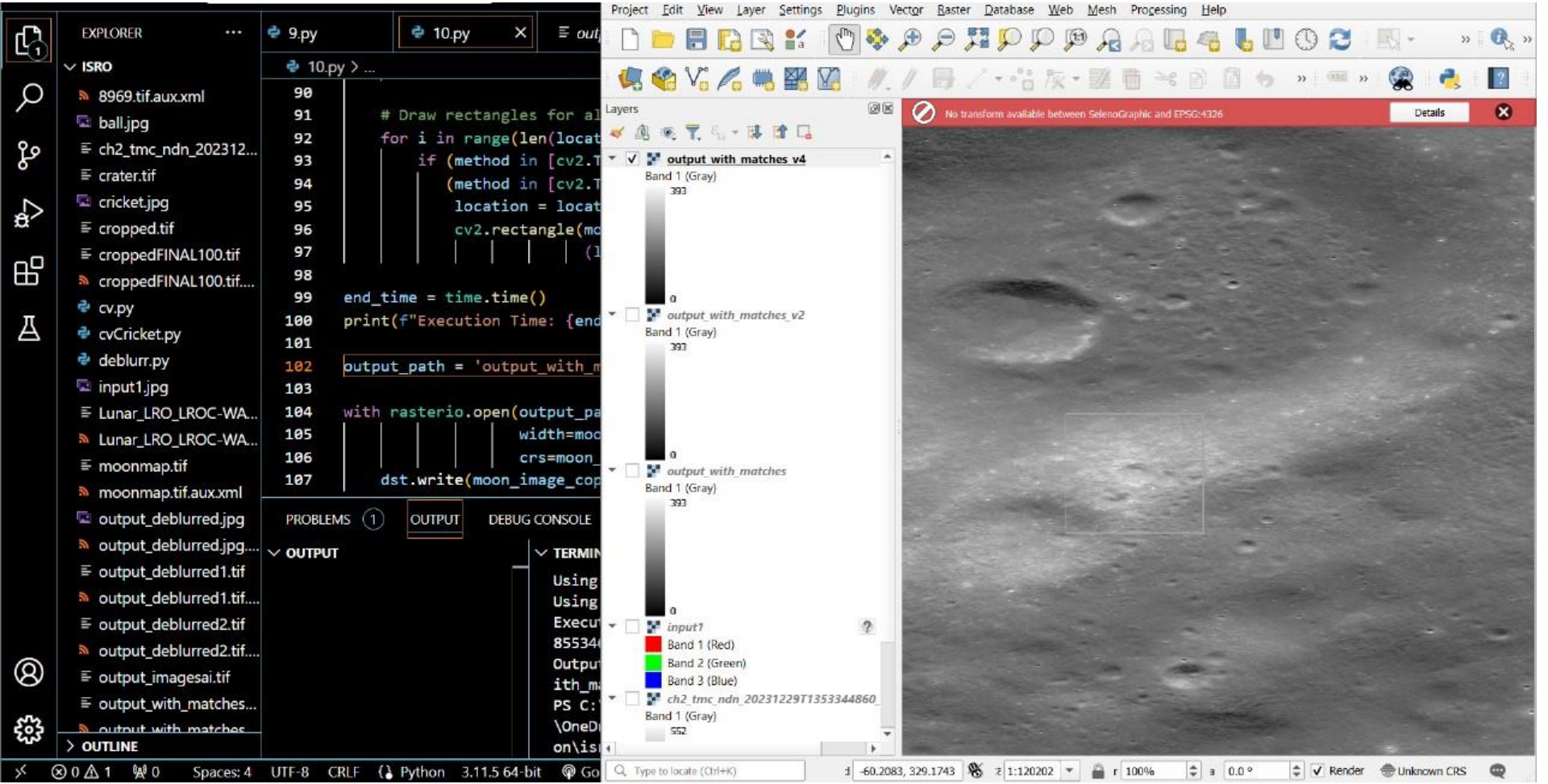
This pseudo code tells:

Resampling the template to match the resolution of the search image ensures that both images are on the same scale, enhancing the accuracy of the match. By resizing the template to appropriate dimensions, you avoid distortions and improve the robustness of the template matching process. Because of both are of different spatial resolution. One is of 100m and another is of 5m.

Proposed architecture/user diagram



Proof of Work : Implementation of Idea



List of features offered by the solution:

- Resolution Alignment: Advanced rescaling techniques to align high-resolution Chandrayaan-2 images with lower-resolution LRO mosaics for accurate template matching.
- Multi-Scale Template Matching: Capability to detect craters at various scales using sophisticated template matching methods.
- Feature Detection: Utilizes ORB (Oriented FAST and Rotated BRIEF) and FLANN (Fast Library for Approximate Nearest Neighbors) for robust feature detection and matching.
- Illumination Normalization: Applies techniques like CLAHE (Contrast Limited Adaptive Histogram Equalization) and Multi-Scale Retinex for consistent image quality and improved detection performance.
- Geospatial Data Handling: Uses GDAL (Geospatial Data Abstraction Library) for converting pixel coordinates to geographic coordinates, facilitating precise location extraction.
- Automated Processing: Streamlines image preprocessing, feature detection, and geospatial analysis in an automated workflow to reduce manual intervention and increase efficiency.
- Visualization: Provides tools for visualizing matches between crater images and mosaics, aiding in verification and analysis.
- Crater Identification: Cross-references detected coordinates with a lunar crater catalogue to identify and name craters.
- High Accuracy: Employs state-of-the-art methods for improved accuracy in crater detection and location extraction.


BHARATIYA ANTARIKSH HACKATHON 2024
Innovation partner H2S
इसरो isro
THANK YOU