HW3: Image Analysis and Segmentation for Environmental Mapping

Addis Ababa University
Addis Ababa Institute of Technology
School of Information Technology and Engineering

Course Title: Probabilistic Graphical Models (ITSC-1051)

Reported To: Beakal Gizachew (PhD)
Reported By: Mahlet Nigussie

JUNE 27, 2024
Addis Ababa , Ethiopi

Image Analysis and Segmentation for Environmental Mapping

1. Introduction

In this project, we aim to utilize satellite imagery and undirected graphical models for mapping environmental features. This approach holds significant promise in enhancing environmental monitoring and analysis by providing detailed spatial information about land cover types such as forests, grassy areas, water bodies, and urban regions. This information plays a crucial role in environmental monitoring, resource management, and understanding the impact of climate change.

2. Methodology

2.1 Data Acquisition

For this project, satellite imagery was obtained from Sentinel Hub, leveraging its comprehensive archive and APIs

Satellite Image Retrieval and Elevation Data:

- Utilizes Earth Engine (ee) library to retrieve Sentinel-2 satellite imagery for a specified location and date range.
- Filters the collection for low cloud cover to ensure image quality.
- Calculates elevation at the specified location using Earth Engine's terrain data.
- Exports the image to Google Drive for further processing (optional).

Source and Dataset Details: The satellite imagery used in this project was sourced from Google Earth Engine (GEE), providing high-resolution images suitable for environmental mapping. The dataset includes images with a resolution of approximately 10 meters per pixel and accompanying metadata such as acquisition date, sensor type (Sentinel-2), and geographic coordinates.

Data Retrieval Process: Images were retrieved programmatically using Python scripts interfacing with the Google Earth Engine API. The scripts were designed to download images based on specified criteria, including the geographic coordinates centered at latitude 11.57628 and longitude 37.42476, and a time range from May 1, 2024, to May 10, 2024. This approach ensured that only relevant images for the environmental mapping study were retrieved.

Location and Image Selection: The script focuses on retrieving Sentinel-2 satellite imagery and elevation data for a specific geographical location centered at latitude 11.57628 and longitude 37.42476, which is located in Bahirdar, Ethiopia. A circular buffer with a radius of approximately 3048 meters (10000ft) is created around this location to encompass the area of interest. This ensures that the subsequent analysis and visualization are centered on this targeted geographic point.

Elevation Data Retrieval: To complement the satellite imagery, elevation data is retrieved using Google Earth Engine's terrain dataset (USGS/SRTMGL1_003). The script calculates the mean elevation within a scale of 30 meters per pixel at the specified location.

Image Retrieval and Visualization: Upon confirming the presence of suitable images in the collection, the script retrieves the first image that meets the filtering criteria. The unique identifier (ID) of the selected image is printed for reference. Using the geemap library, an interactive map centered on the specified location is generated. The retrieved Sentinel-2 image is visualized on this map with specific visualization parameters, emphasizing the Red, Green, and Blue (RGB) bands. This approach enhances the clarity and interpretability of the satellite data by adjusting the color intensity and contrast through parameters like min, max, and gamma.

Map Overlay and User Interaction: To provide context, the script overlays the defined circular buffer (location geometry) onto the map with a distinctive pink color. This overlay helps in visualizing the spatial extent of the area of interest relative to the retrieved Sentinel-2 image. The interactive map interface includes a layer control panel, allowing users to toggle between different layers (e.g., image and location overlay) for better exploration and analysis of the satellite data.

Image Retrieval and Export Process: After confirming the presence of suitable Sentinel-2 satellite images in the collection, the script proceeds with exporting the selected image to a google drive for further analysis and archival purposes. The exported image is formatted as GeoTIFF, ensuring compatibility with Geographic Information System (GIS) software for subsequent processing.

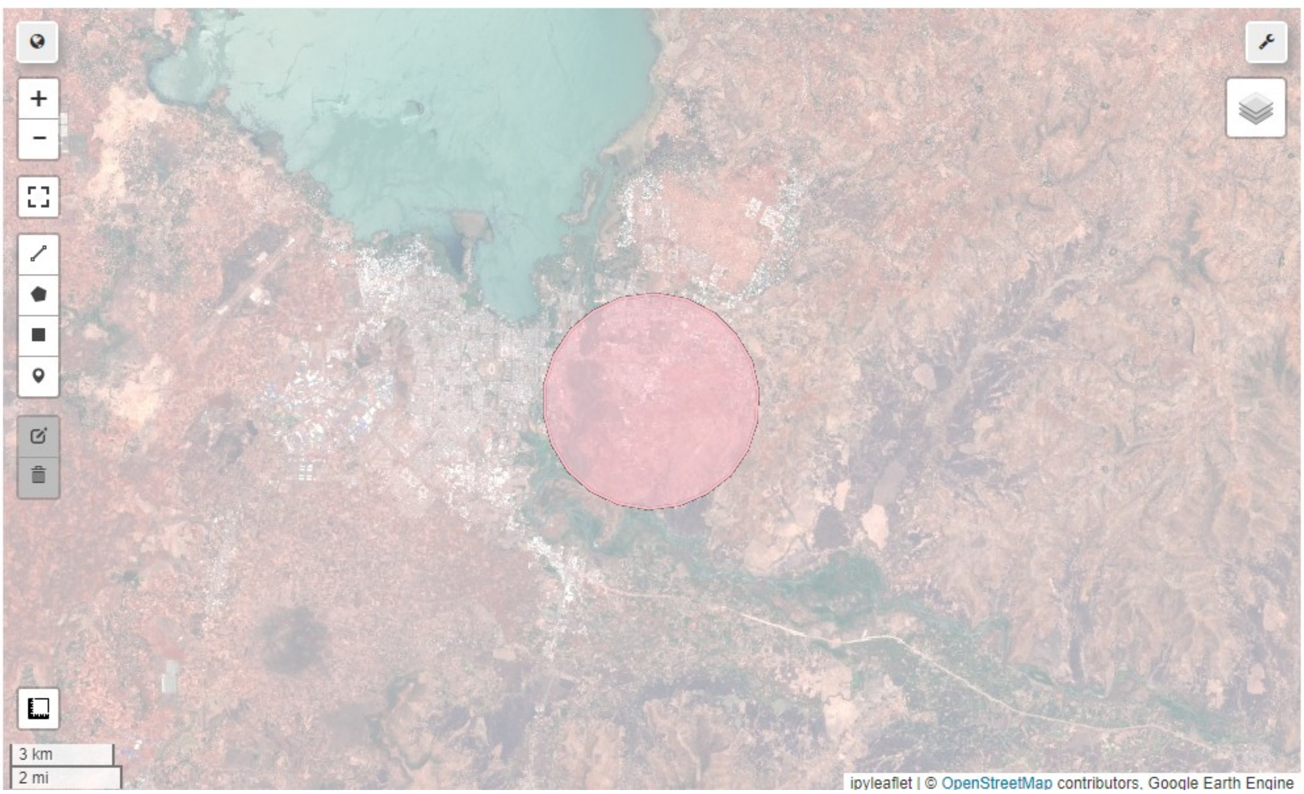

Figure 1: Image Retrieval

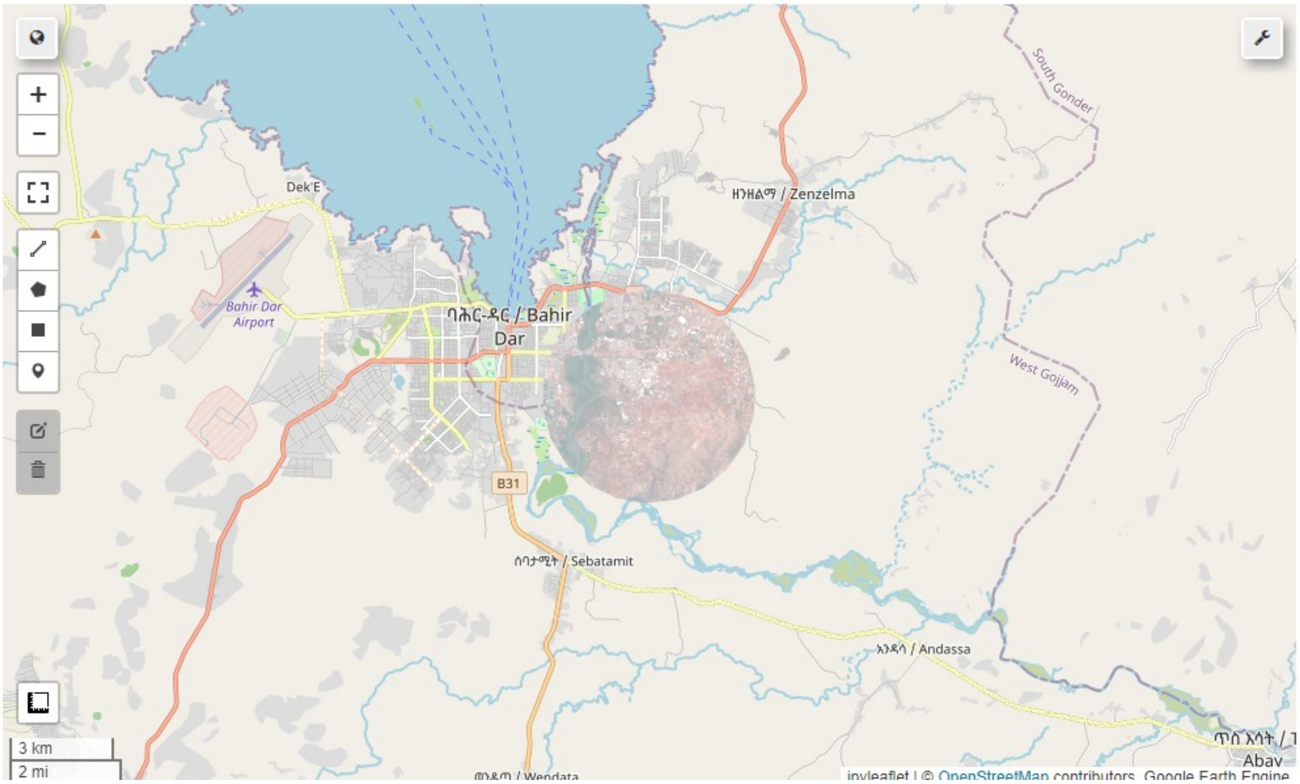

Figure 2: Image Retrieval hiding the rest

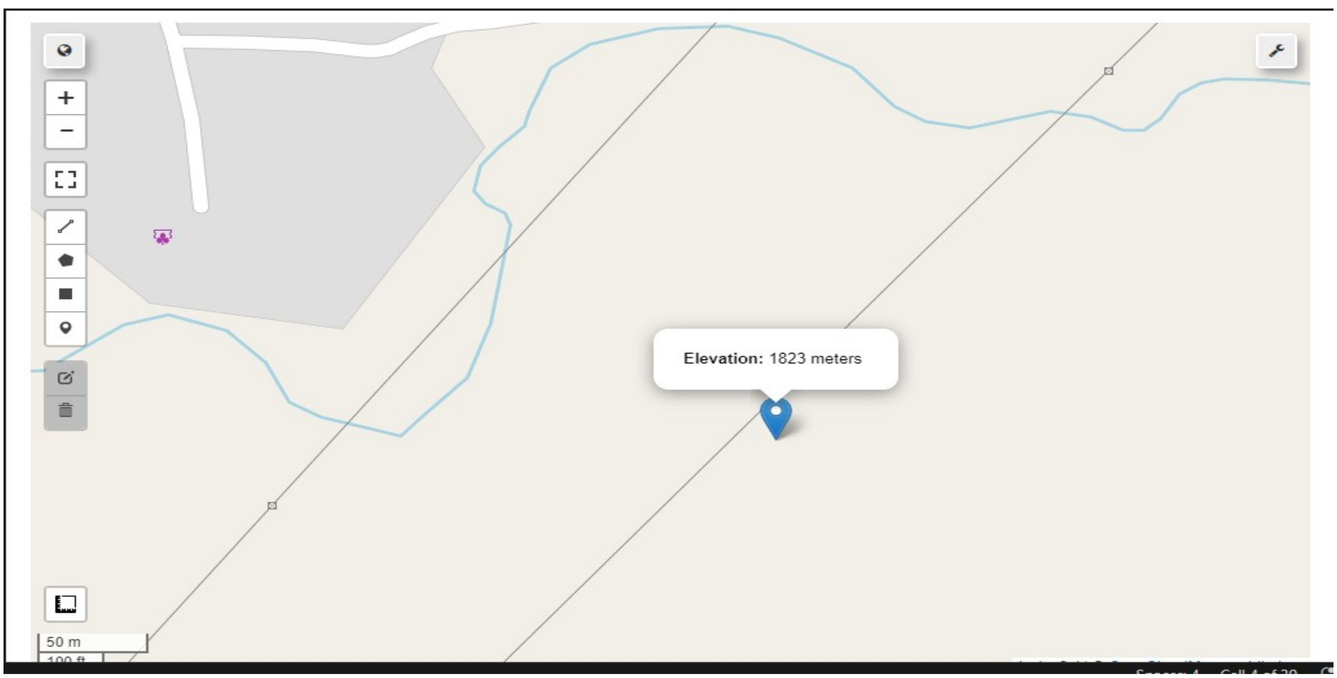

Figure 3: Image elevation

Figure 4: Exported image

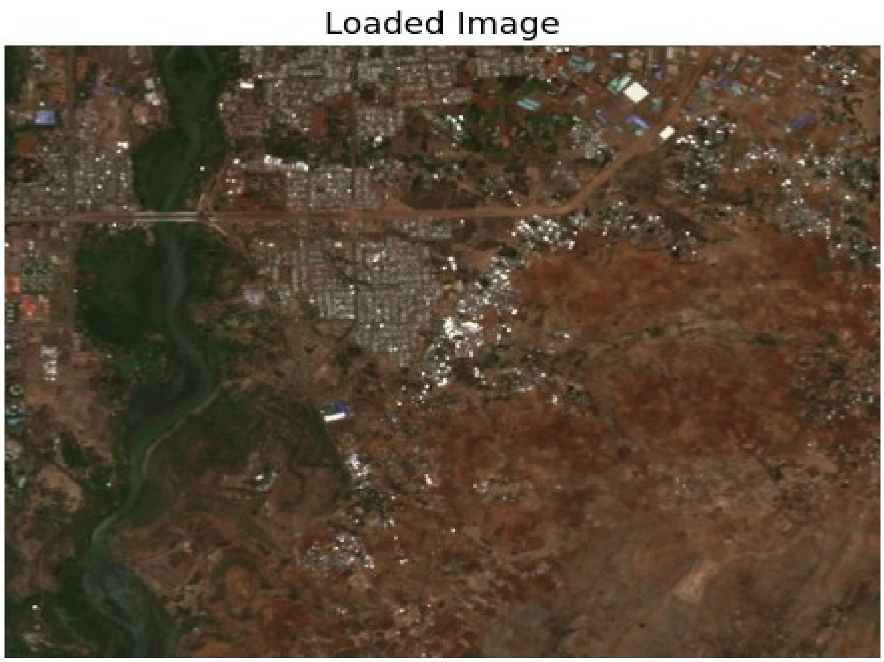

Figure 5: Converted PNG file of the image

We are abe to demonstrate an effective use of Google Earth Engine and geemap libraries for accessing, filtering, and visualizing Sentinel-2 satellite imagery sourced from Google Earth Engine. By focusing on a specific geographic location and applying rigorous filtering criteria programmatically, the script facilitates targeted environmental monitoring and analysis applications. The interactive map interface enhances user engagement and facilitates detailed exploration of satellite data, making it a valuable tool for researchers and practitioners in fields such as environmental science, agriculture, and urban planning.

2.2 Data Preprocessing

Various image preprocessing techniques are applied using libraries such as rasterio and opencv:

- **Histogram Stretching**: Enhances image contrast.
- **Histogram Equalization (CLAHE)**: Improves local contrast.
- **Image Resizing**: Adjusts image dimensions for analysis and visualization purposes.

Image Retrieval and Resizing: Images were retrieved at their original resolutions and resized to a standardized format of [350 x 350] for consistency in analysis. Resizing ensures uniformity in pixel dimensions across all images, facilitating efficient processing and comparison.

Noise Reduction and Normalization

Noise Reduction: To mitigate noise in images caused by sensor artifacts or atmospheric conditions, median filtering was applied using OpenCV's medianBlur function. This step helps in improving image quality by smoothing out irregularities while preserving important details.

Normalization: Pixel values were normalized to a range of [0, 1] to enhance consistency and facilitate accurate feature extraction and classification algorithms. Normalization ensures that pixel intensities are comparable across different images.

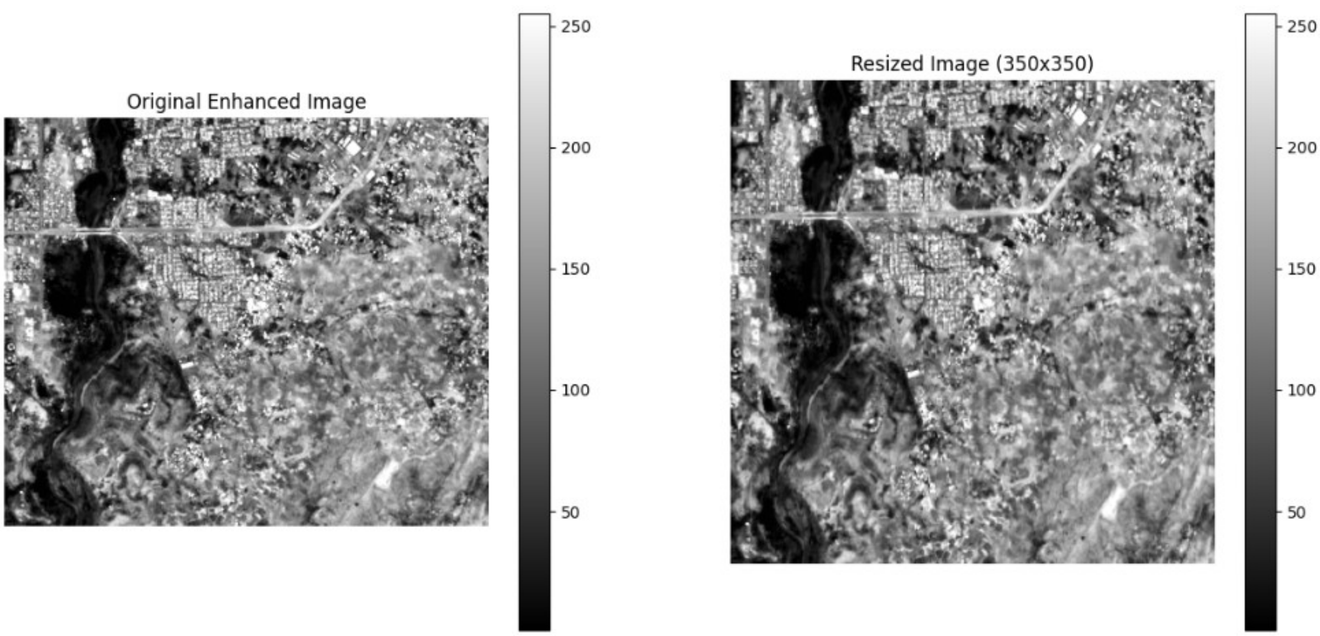

Figure 6: Preprocessed Images

Image Preprocessing and Display:

- Loads the downloaded image or a local image file.
- Applies pre-processing steps like contrast stretching and histogram equalization to enhance image features.
- Resizes the image to a consistent resolution (e.g., 350x350 pixels).
- Displays the original and preprocessed images for visualization.

3. Model Formulation

3.1 Undirected Graphical Model

An undirected graphical model was employed to represent spatial relationships between pixels in the satellite images. Nodes in the graph correspond to individual pixels, and edges capture pairwise dependencies based on proximity and similarity in spectral characteristics.

Undirected Graphical Model (UGM) for Segmentation:

- Constructs a UGM using the networkx library.
- Represents pixels as nodes and spatial relationships between neighboring pixels as edges.
- Defines source and sink nodes to guide the segmentation process.

3.2 Feature Detection

- Forest Regions: Detection of forest regions relied on spectral characteristics such as green hues in RGB color space and specific texture patterns indicative of dense vegetation cover.
- Grassy Areas: Grassy areas were identified based on spectral signatures associated with vegetation cover that differs from forests, often characterized by lighter green tones and distinct texture features.

Grassy Area Detection:

- Converts the image to grayscale if necessary.
- Implements a basic segmentation approach to identify potential grassy areas based on intensity thresholds (you can modify these thresholds for better green area detection).

4. Implementation Details

Green areas are detected within the image based on predefined intensity thresholds. Segmentation is performed using maximum flow algorithms on the constructed graphical model to isolate green areas from the background.

- Leverages the UGM and maximum flow algorithm from networkx to perform image segmentation.
- Assigns higher capacities to edges connecting grassy pixels, encouraging the flow towards the sink node.
- Generates a binary mask where white pixels represent segmented grassy areas.

In this project, It implemented a segmentation algorithm based on a maximum flow approach using the Push-Relabel algorithm. This method was chosen due to its efficiency compared to other algorithms like belief propagation, which proved computationally expensive and impractical for real-time applications.
Initially, I attempted to use belief propagation for segmentation. However, the algorithm's intensive computational demands caused my system to freeze multiple times, and I encountered significant delays, with compilation times exceeding three hours on Google Colab sessions that often timed out before completion. These challenges prompted me to explore alternative approaches, ultimately leading to the adoption of the maximum flow algorithm.

The implemented solution efficiently segments regions corresponding to grassy areas based on a provided mask. By leveraging graph theory and the NetworkX library's maximum_flow function, the algorithm effectively determines optimal paths within the graph structure, achieving segmentation results in a more manageable timeframe.

This approach not only addressed the computational limitations encountered with belief propagation but also provided a robust framework for future optimizations and scalability in similar image processing tasks.

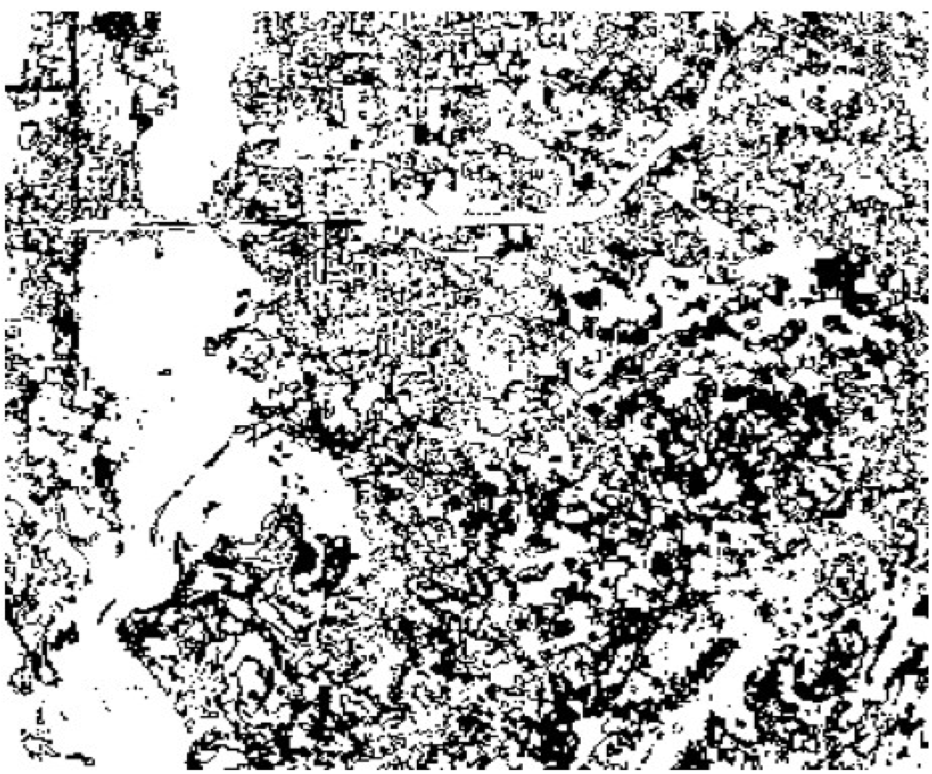

Figure 7: Segmented Image

5. Visualization Techniques

Visual representations of segmented images with color-coded boundaries were generated using matplotlib to highlight detected features such as forests and grassy areas. Color gradients were utilized to distinguish different land cover types for enhanced visual interpretation.

Visualization of Segmented Image and Environmental Features:

- Overlays the segmented image on the original image to highlight detected grassy regions.
- Implements a function to detect potential environmental features (like forests) based on connected regions in the image.
- Visualizes the detected features with color-coded circles and connects them to the specified location using lines.

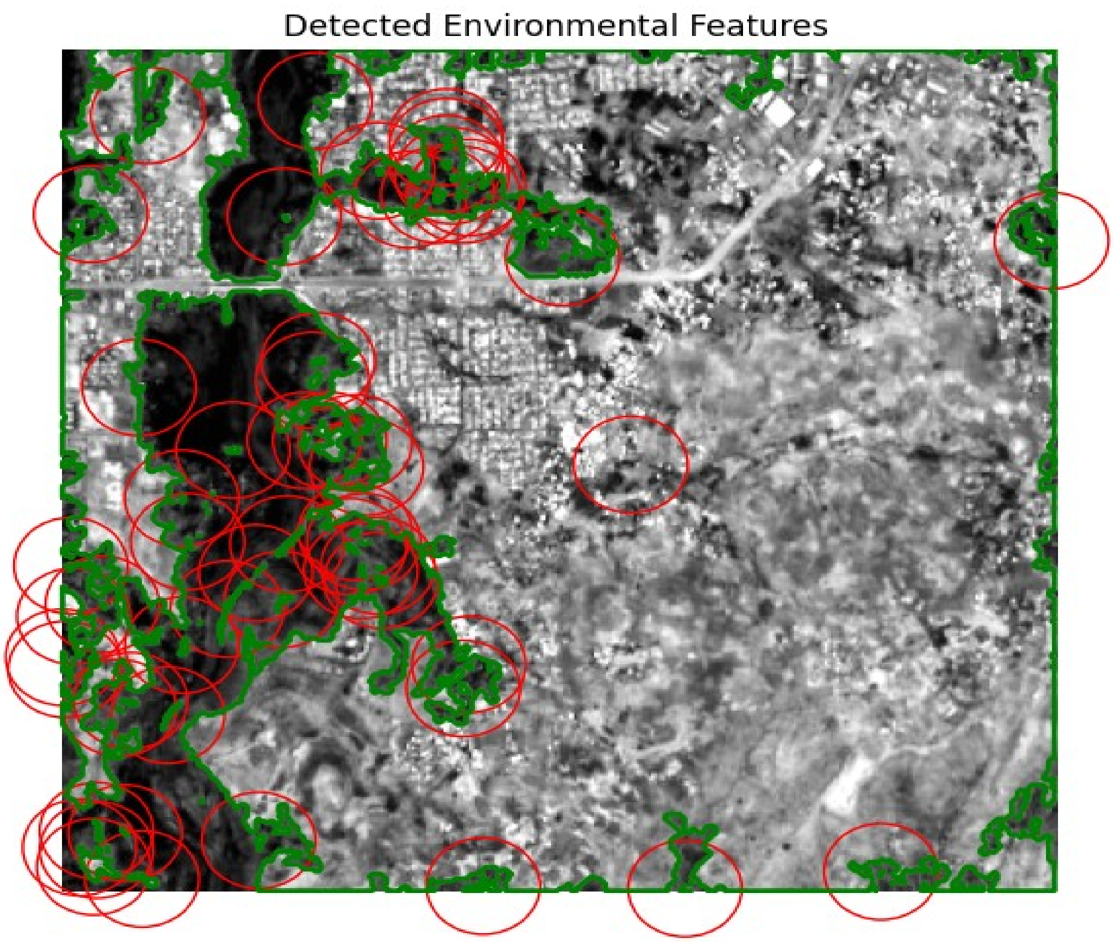

Figure 8: Detected Environmental Features image

- Detected environmental features, such as green areas and other significant contours, are visualized on the original image. This visualization aids in understanding spatial distributions and relationships within the mapped area.

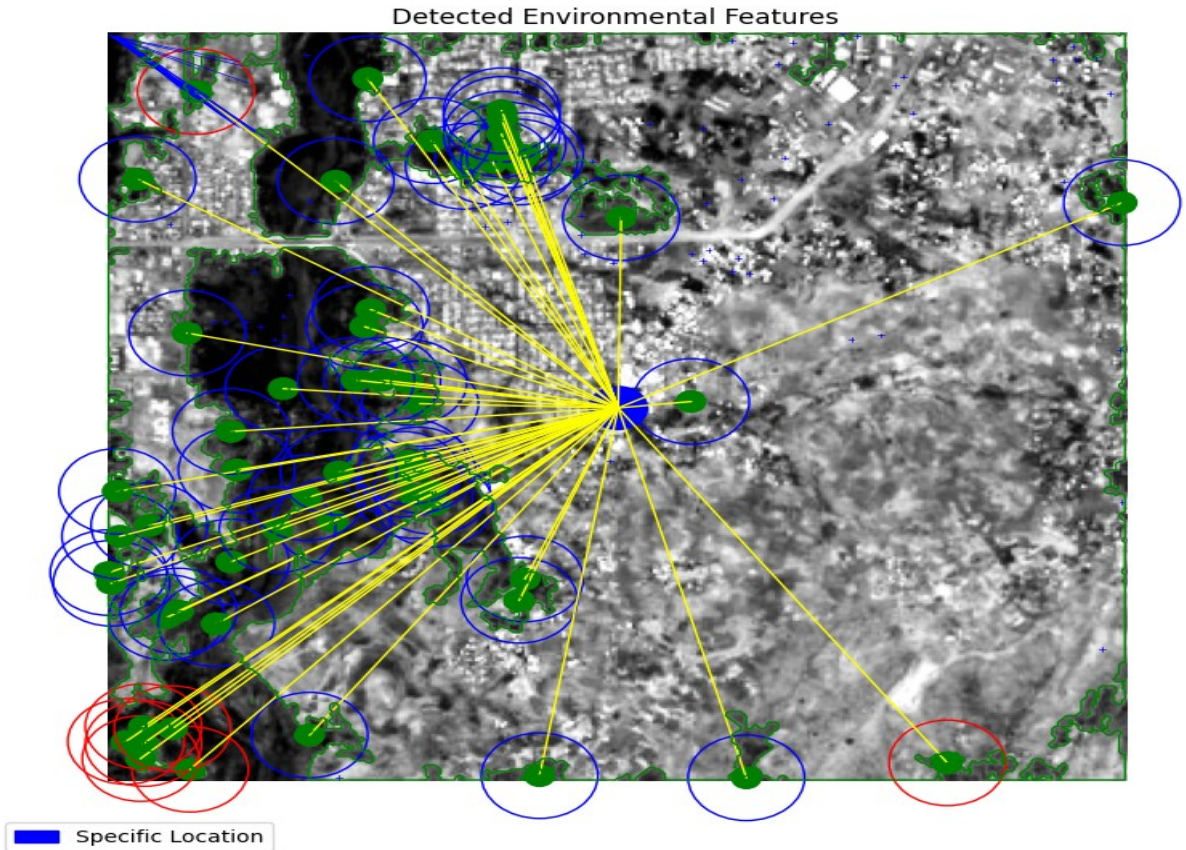

Figure 9: Detected Environmental Features in specific location image

6.**Evaluation Results**

6.1 Performance Metrics

Evaluation of segmentation accuracy was conducted using metrics including Intersection over Union (IoU), Precision, and Recall scores compared against ground truth data or expert annotations.

Evaluation Metrics:

- Calculates Intersection over Union (IoU), Precision, Recall, and F1-score to evaluate the segmentation performance.
- Compares the segmented image with a sample green area image (provided manually) to calculate these metrics.
- Displays the original, sample, and segmented masks alongside the IoU value for better understanding.

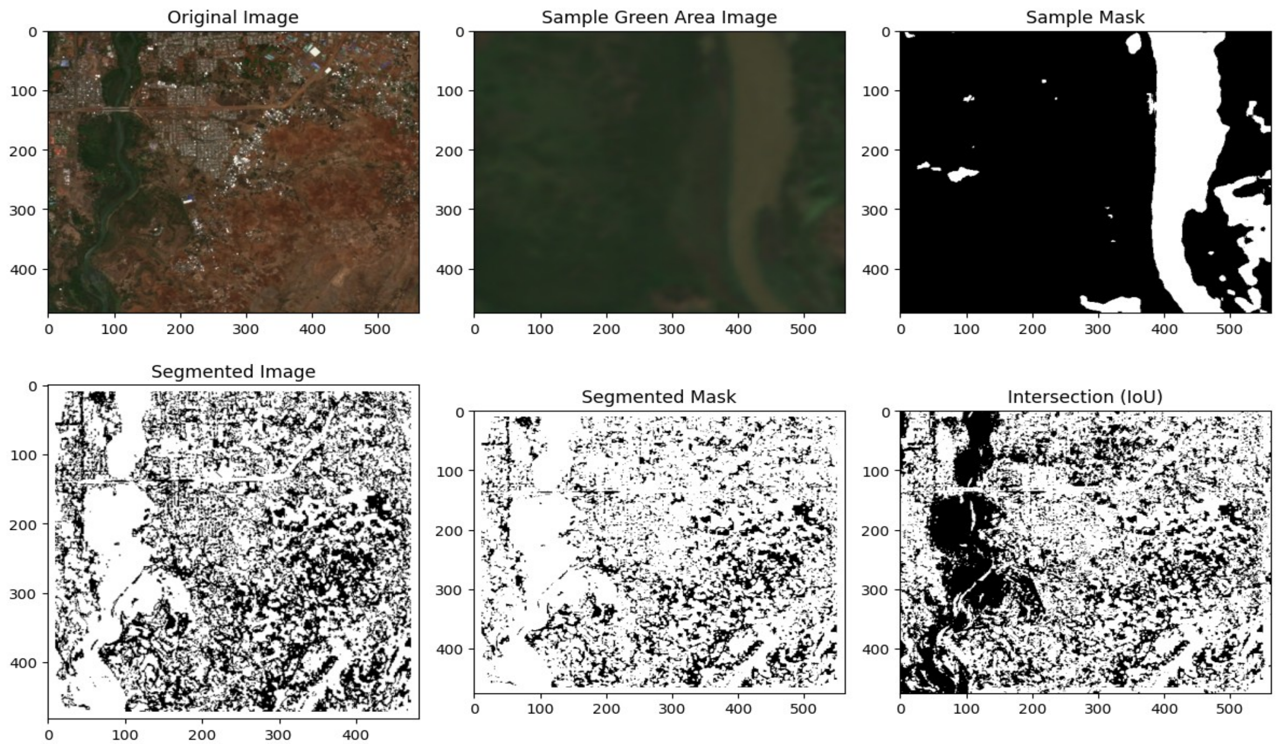

Figure 10: Intersection image generation

After implementing the segmentation algorithm using undirected graphical models on satellite imagery, the performance was evaluated using the following metrics:

- **Intersection over Union (IoU):** 0.6454
- **Precision:** 0.7723
- **Recall:** 0.7970
- **F1 Score:** 0.7845

These metrics provide a quantitative assessment of the segmentation algorithm's effectiveness in detecting environmental features such as forests and grassy areas. Here's a detailed interpretation of each metric:

- **Intersection over Union (IoU):** IoU measures the overlap between the predicted segmentation and the ground truth segmentation. An IoU of 0.6454 indicates that approximately 64.54% of the predicted region overlaps with the ground truth region, reflecting moderate accuracy in spatial localization of features.

- **Precision:** Precision quantifies the accuracy of positive predictions made by the model. With a precision of 0.7723, the model accurately identifies 77.23% of the predicted environmental features correctly, minimizing false positives.

- **Recall:** Recall measures the completeness of positive predictions by the model. A recall of 0.7970 indicates that the model successfully identifies 79.70% of all actual environmental features present in the satellite imagery, minimizing false negatives.

- **F1 Score:** The F1 score is the harmonic mean of precision and recall, providing a balanced measure of the model's overall performance. An F1 score of 0.7845 indicates a good balance between precision and recall, suggesting robust performance in detecting and accurately delineating environmental features.

These metrics collectively demonstrate the segmentation algorithm's capability in environmental mapping using satellite imagery, highlighting its effectiveness in identifying and delineating features of interest. Further analysis and interpretation of these metrics help validate the accuracy and reliability of the segmentation results, crucial for environmental monitoring and analysis applications.

- Potential applications and implications of the developed pipeline for environmental mapping and monitoring.
    - Potential Applications and Implications
    - Environmental monitoring for deforestation detection.
    - Urban planning for green spaces identification.
    - Improved accuracy in identifying specific land cover types.
    - Facilitating data-driven decision-making in environmental management.

6.3 Improvement Opportunities

The project successfully demonstrates the application of Python libraries and Earth observation data for environmental mapping. Techniques such as image retrieval, preprocessing, graphical modeling, and feature detection contribute to comprehensive analysis and visualization of satellite imagery.

Opportunities for improvement focused on refining feature detection algorithms to handle complex landscapes more effectively. Proposed enhancements included adaptive thresholding techniques, parameter optimization in graphical models, and integration of additional spectral bands for enhanced classification.

- Incorporating texture or color features in addition to intensity for more robust detection of green areas.
- Using reference datasets for ground truth validation and evaluating the effectiveness of different segmentation parameters.
- Exploring advanced visualization techniques to create more informative environmental maps.

7. Conclusion

This project demonstrates the potential of using undirected graphical models for environmental feature mapping with satellite imagery. The UGM approach allowed us to effectively segment the image and extract relevant information about the target environment. The project highlights the significance of satellite remote sensing and UGM-based techniques in environmental monitoring and analysis. Future research directions can involve exploring advanced UGM models, incorporating multispectral or hyperspectral imagery, and developing more robust feature extraction methods for improved environmental mapping capabilities