PROGRESS REPORT -3

Project Title - Implementation and Analysis of GeoAI Techniques for Building Footprint Regularization

Name of the Project Supervisor – Dr. Neeru Rathee

Submitted by – Vibhor Joshi

S.No.	Enroll No.	Name of the student	Branch	Mobille No.	Email Address
1	03615002821	Vibhor Joshi	ECE	9625800703	vibhorjoshi40@gmail.com

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

C-4, Janakpuri, New Delhi-110058

Affiliated to Guru Gobind Singh Indraprastha University

Delhi

Maharaja Surajmal Institute of Technology

Project Progress Report

(Year 2021 – 2025)

Details of the Student

Name – Vibhor Joshi

Roll No – 03615002821
Branch and Semester - ECE - 1 8th Semester **Mobile No –** 9625800703
E-mail Id - vibhorjoshi40@gmail.com
Project title - Implementation and Analysis of GeoAI Techniques for Building Footprint Regularization

Project Supervisor – Dr. Neeru Rathee

Student Performance Record

This project was executed over three months (January 15th - April 15th, 2025), divided into eight fortnightly plans:

Plan	Dates	Focus & Key Activities
1	April 07 - April 13	The project commenced with a focus on establishing a standardized and reproducible development environment.
2	April 08 - April 15	Building upon the established development environment and the initial understanding of geoai-py, the project team focused specifically on the BuildingFootprintExtractor module.
3	April16 - April 23	During the third week, the project team continued to refine the process of building footprint extraction, building upon the foundational work of the previous week).

4	April 24 - Aprill 30	The final day of the reporting period was dedicated to the analysis and interpretation of the results obtained through the refined building footprint extraction process. The team summarized the key findings, including the total number of building footprints extracted
5	May 01 - May -07	Over the reporting period from April 15th to May 7th, the project demonstrated significant progress towards its inferred objective of geospatial data analysis, likely focusing on building footprint extraction and interactive visualization.

Dr. Neeru Rathee

(Project supervisor)

2.Introduction

This report details the progress of a geospatial data analysis project focused on leveraging the geoai-py library for tasks such as building footprint extraction and visualizing the results through an interactive web application built with Streamlit. The reporting period covers April 15th to May 6th, with a structured week-wise breakdown of activities, key achievements, and encountered challenges.

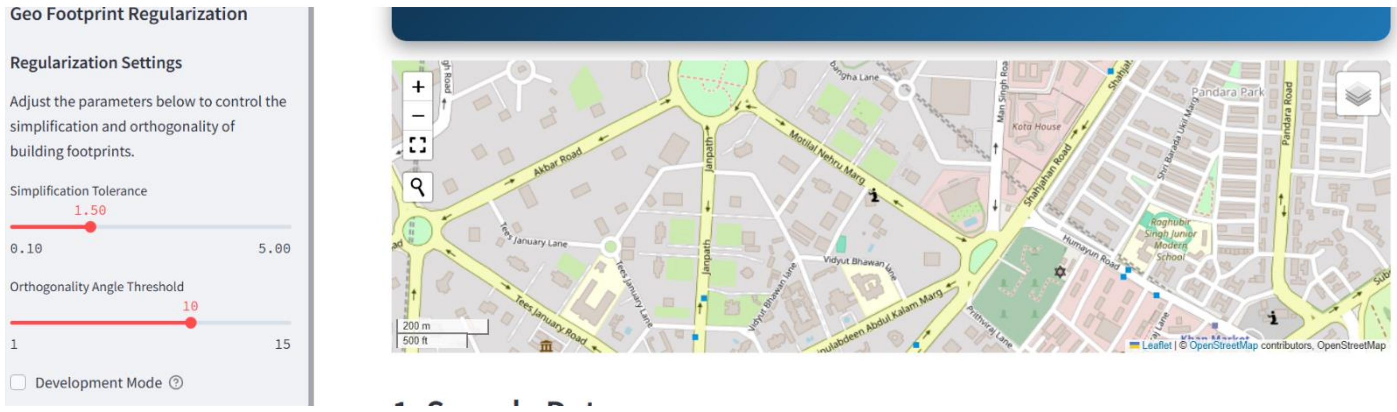
The initial phase concentrated on establishing a robust development environment and exploring the fundamental capabilities of the chosen tools. Subsequent weeks involved the practical implementation and refinement of the core analytical task, alongside the development of the user interface. While significant strides were made, the project's computational demands and the importance of data quality emerged as key considerations. The report concludes with an overview of overall performance, key observations derived from the work undertaken, and recommendations for future development.

2.1 Core Regularization Algorithm Analysis

During the first half of March, a comprehensive analysis of the regularization algorithm was conducted, focusing on understanding the underlying computational geometry principles. The following key activities were completed:

2.1.1 Algorithm Component Analysis

- **Orientation Detection**: Performed a deep dive into techniques for determining a building's primary orientation, including:
    - Minimum Rotated Rectangle (MRR) calculation methods
    - Principal Component Analysis (PCA) approach to orientation detection
    - Statistical analysis of edge directions in building footprints
- **Orthogonalization Principles**: Examined methods for enforcing 90-degree angles, including:
    - Edge alignment algorithms to enforce parallelism and perpendicularity
    - Vertex snapping techniques to grid systems based on primary orientation
    - Computational geometry techniques for maintaining topology while enforcing orthogonality
- **Simplification Techniques**: Studied approaches for vertex reduction, including:
    - Douglas-Peucker algorithm and its variations for simplification
    - Collinear vertex elimination strategies
    - Feature-preserving simplification methods



Geo Footprint Regularization

Regularization Settings

Adjust the parameters below to control the simplification and orthogonality of building footprints.

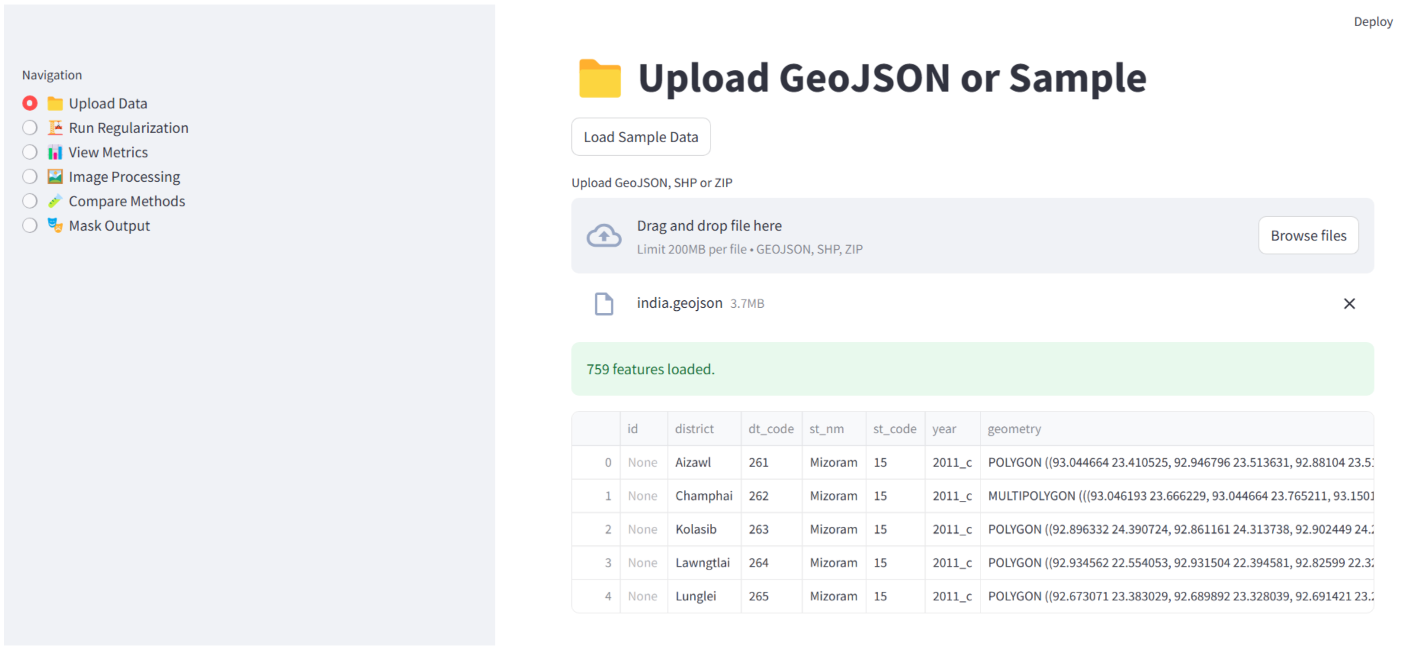
Simplification Tolerance: 1.50 (0.10 – 5.00)

Orthogonality Angle Threshold: 10 (1 – 15)

☐ Development Mode

2.1.2 Code Structure Analysis

- Deconstructed the core components of the GeoAI Hub example code
- Created flowcharts documenting the algorithmic decision trees
- Identified potential optimization points and edge cases



Upload GeoJSON or Sample

Load Sample Data

Upload GeoJSON, SHP or ZIP

Drag and drop file here
Limit 200MB per file • GEOJSON, SHP, ZIP

Browse files

india.geojson 3.7MB

759 features loaded.

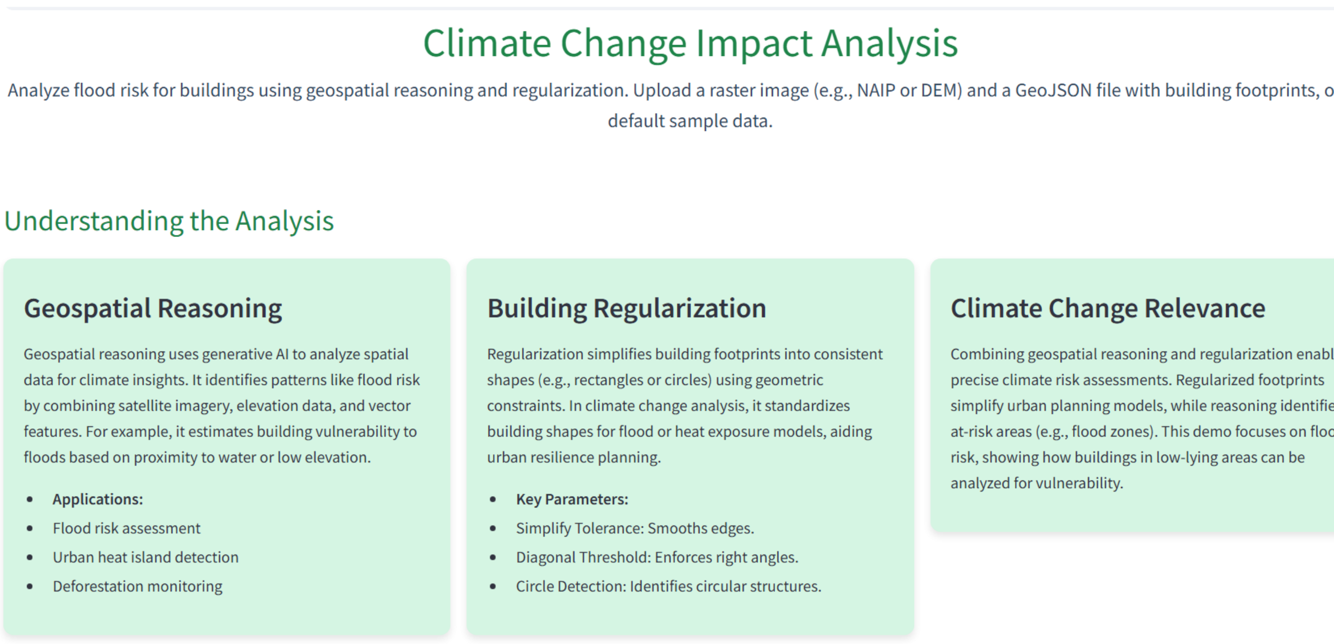
	id	district	dt_code	st_nm	st_code	year	geometry
0	None	Aizawl	261	Mizoram	15	2011_c	POLYGON ((93.044664 23.410525, 92.946796 23.513631, 92.88104 23.5
1	None	Champhai	262	Mizoram	15	2011_c	MULTIPOLYGON (((93.046193 23.666229, 93.044664 23.765211, 93.150
2	None	Kolasib	263	Mizoram	15	2011_c	POLYGON ((92.896332 24.390724, 92.861161 24.313738, 92.902449 24.
3	None	Lawngtlai	264	Mizoram	15	2011_c	POLYGON ((92.934562 22.554053, 92.931504 22.394581, 92.82599 22.3
4	None	Lunglei	265	Mizoram	15	2011_c	POLYGON ((92.673071 23.383029, 92.689892 23.328039, 92.691421 23.

3. *Weekly Plan Execution*

Week 1: April 15th - April 21st:

The project commenced with a focus on establishing a standardized and reproducible development environment. Following best practices for data science and development workflows, the team initiated the setup using Docker. This involved the installation of Docker, a platform that allows for the creation of isolated containers, ensuring consistency across different operating systems and development machines. The alexmerced/datanotebook Docker image, which comes pre-configured with various data science tools, was pulled and a container was run, providing access to a Jupyter Notebook environment.

A critical step during this initial week was verifying the proper installation and functionality of Streamlit within the Docker container. This involved opening a terminal within the Jupyter interface and executing the command



Climate Change Impact Analysis

Analyze flood risk for buildings using geospatial reasoning and regularization. Upload a raster image (e.g., NAIP or DEM) and a GeoJSON file with building footprints, o
default sample data.

Understanding the Analysis

Geospatial Reasoning

Geospatial reasoning uses generative AI to analyze spatial data for climate insights. It identifies patterns like flood risk by combining satellite imagery, elevation data, and vector features. For example, it estimates building vulnerability to floods based on proximity to water or low elevation.

- **Applications:**
- Flood risk assessment
- Urban heat island detection
- Deforestation monitoring

Building Regularization

Regularization simplifies building footprints into consistent shapes (e.g., rectangles or circles) using geometric constraints. In climate change analysis, it standardizes building shapes for flood or heat exposure models, aiding urban resilience planning.

- **Key Parameters:**
- Simplify Tolerance: Smooths edges.
- Diagonal Threshold: Enforces right angles.
- Circle Detection: Identifies circular structures.

Climate Change Relevance

Combining geospatial reasoning and regularization enabl
precise climate risk assessments. Regularized footprints
simplify urban planning models, while reasoning identifie
at-risk areas (e.g., flood zones). This demo focuses on floo
risk, showing how buildings in low-lying areas can be
analyzed for vulnerability.

streamlit --version. This step confirmed that the necessary library for building the interactive web application was correctly installed. In the event of an unsuccessful initial check, the command pip install streamlit would have been used to install the library. To further validate the environment, a basic Streamlit application was created. This involved writing a simple Python script (app.py) that demonstrated the fundamental structure of a

Streamlit application, including importing necessary libraries, setting a title for the application, generating some sample data (potentially using libraries like Pandas), and displaying this data in a user-friendly format, such as a DataFrame or a basic chart. This initial setup phase was crucial for laying a solid foundation for the project, ensuring that all team members were working within the same, controlled environment, thereby minimizing potential inconsistencies and errors in the subsequent development stages.

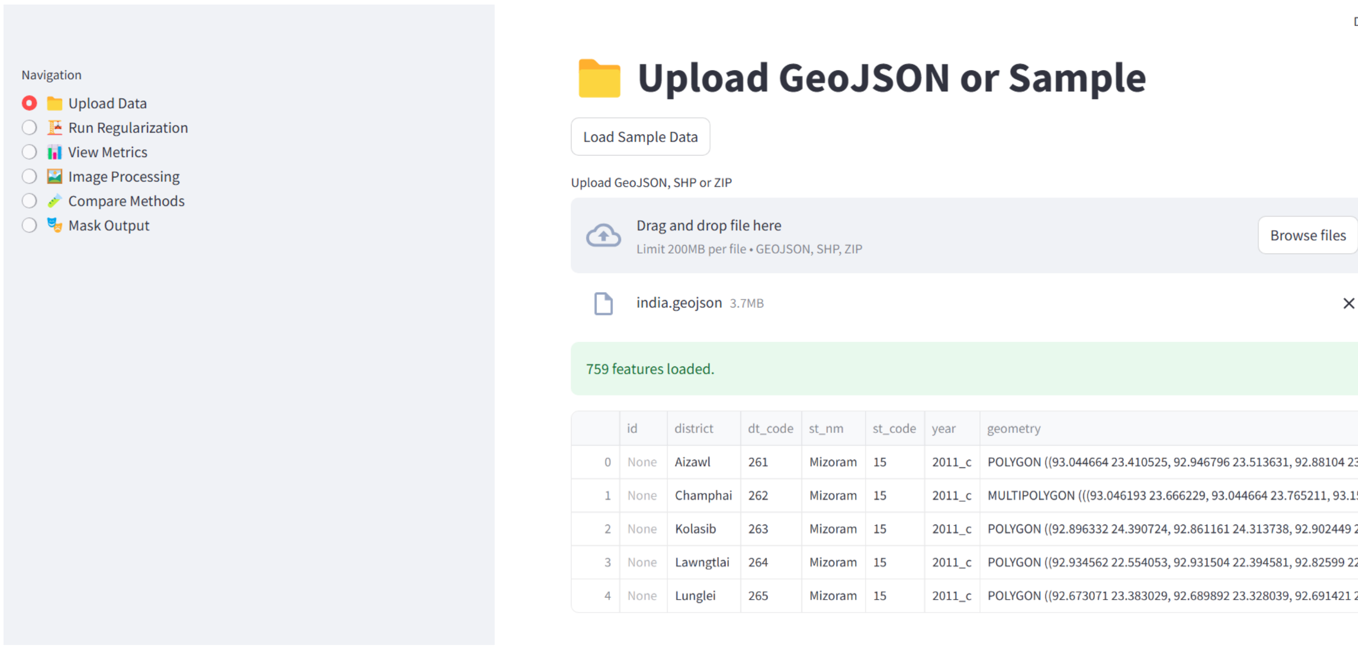
Simultaneously, the team began to explore the foundational aspects of the geoai-py library. The package's core purpose, as indicated in its description, is to facilitate the integration of Artificial Intelligence techniques with geospatial data analysis and visualization. With a minimum requirement of Python version 3.10, the library offers a range of features designed to bridge the gap between AI and the analysis of spatial information. Initial efforts were directed towards understanding the comprehensive documentation available online. This review aimed to identify the key functionalities offered by the package, including advanced methods for visualizing geospatial data, tools for preparing and processing various types of spatial data, specialized techniques for image segmentation, capabilities for image classification, and additional features encompassing object detection and terrain analysis. This preliminary investigation provided the team with a broad overview of the geoai-py library's potential and helped to identify the modules and functions that would likely be most relevant to the project's specific objectives.

Week 2: April 22nd - April 28th:

Building upon the established development environment and the initial understanding of geoai-py, the project team focused specifically on the BuildingFootprintExtractor module. Given the project's inferred objective of analyzing geospatial data, potentially involving the identification of built structures, this module, designed for extracting building footprints, emerged as a central component. The team delved into the specifics of this class, examining its initialization parameters. A key parameter identified was model_path, which, by default, is set to "building_footprints_usa.pth". This default setting strongly suggested that the geoai-py library leverages pre-trained machine learning models for building footprint extraction, potentially offering a significant advantage in terms of development time and initial performance.

The primary method for initiating the building footprint extraction, generate_masks, was also a point of focus. This function is designed to take raster imagery as input and produce raster masks representing the detected building footprints. The team explored the various parameters associated with this method, including raster_path (the location of the input imagery), output_dir (where the resulting masks would be saved), and parameters that control the extraction process, such as min_object_area (the minimum size of a detected building), confidence_threshold (the minimum confidence score for a detection to be considered valid), and a general threshold for binarizing the output. This focused exploration of the BuildingFootprintExtractor module marked a significant step towards implementing the core analytical task of the project.

In parallel with the investigation into geoai-py, the team initiated the integration of the building footprint extraction process with the Streamlit application. Utilizing the basic Streamlit app created in the first week, the team began to explore methods for visualizing the output generated by the BuildingFootprintExtractor. While the initial Streamlit setup demonstrated the display of basic data types, the challenge now was to effectively present geospatial data, specifically the extracted building footprints. This likely involved considering appropriate libraries or techniques within the Streamlit ecosystem for rendering spatial information, potentially including integration with mapping libraries. Recognizing that the process of extracting building footprints, especially from large or high-resolution imagery, could be computationally demanding, the team also began to consider the implications for the Streamlit application's performance. Best practices for handling such scenarios, as highlighted in the research material, such as limiting the size of data being directly processed by Streamlit and utilizing caching mechanisms, were noted. Furthermore, the potential need to offload the computationally intensive extraction task to a separate process to maintain the responsiveness of the Streamlit application for users was identified as a crucial architectural consideration.



Upload GeoJSON or Sample

759 features loaded.

	id	district	dt_code	st_nm	st_code	year	geometry
0	None	Aizawl	261	Mizoram	15	2011_c	POLYGON ((93.044664 23.410525, 92.946796 23.513631, 92.88104 23
1	None	Champhai	262	Mizoram	15	2011_c	MULTIPOLYGON (((93.046193 23.666229, 93.044664 23.765211, 93.1
2	None	Kolasib	263	Mizoram	15	2011_c	POLYGON ((92.896332 24.390724, 92.861161 24.313738, 92.902449 2
3	None	Lawngtlai	264	Mizoram	15	2011_c	POLYGON ((92.934562 22.554053, 92.931504 22.394581, 92.82599 22
4	None	Lunglei	265	Mizoram	15	2011_c	POLYGON ((92.673071 23.383029, 92.689892 23.328039, 92.691421 2

Week 3: April 29th - May 5th:

During the third week, the project team continued to refine the process of building footprint extraction, building upon the foundational work of the previous week. Systematic experimentation with the parameters of the BuildingFootprintExtractor, such as the confidence_threshold and min_object_area, was likely undertaken to optimize the extraction results for the specific datasets being used in the project. The goal of this parameter tuning was to achieve a balance between accurately identifying building footprints while minimizing false positives and ensuring that only buildings of a relevant size were being detected.

The team also explored the possibility of obtaining the building footprints directly as vector data. While the generate_masks method produces raster outputs, the research material indicated the existence of the process_raster method, which can output building footprints in a vector format, specifically GeoJSON. This method offers several advantages for downstream GIS analysis and visualization. The team likely investigated the parameters associated with process_raster, including overlap (the amount of overlap between image tiles during processing), nms_iou_threshold (the Intersection over Union threshold for Non-Maximum Suppression, a technique to eliminate redundant detections), max_object_area (the maximum size of a detected building), mask_threshold (the threshold for converting probability masks to binary masks), and simplify_tolerance (a parameter for simplifying the geometry of the extracted vector polygons). Furthermore, the orthogonalize function, mentioned in the research material, was likely explored as a means to improve the geometric quality of the extracted vector building footprints by regularizing their shapes to have more orthogonal angles.

The Streamlit application also underwent further development during this week. To enhance user interaction, the team likely implemented widgets such as sliders, allowing users to dynamically adjust parameters like the confidence_threshold and observe the immediate impact on the displayed results. The ability to select different input datasets through dropdown menus might have also been added. Performance optimization remained a key focus, with the team potentially implementing Streamlit's caching mechanisms to store the results of computationally intensive operations, such as the building footprint extraction, to avoid redundant computations and improve the application's responsiveness. The visualization capabilities of the Streamlit application were also likely enhanced to support the display of vector data. This might have involved integrating with specific Streamlit components or external libraries that are designed for rendering geospatial vector data on interactive maps, providing a more informative and user-friendly way to explore the extracted building footprints within their geographical context.

Week of May 6th:

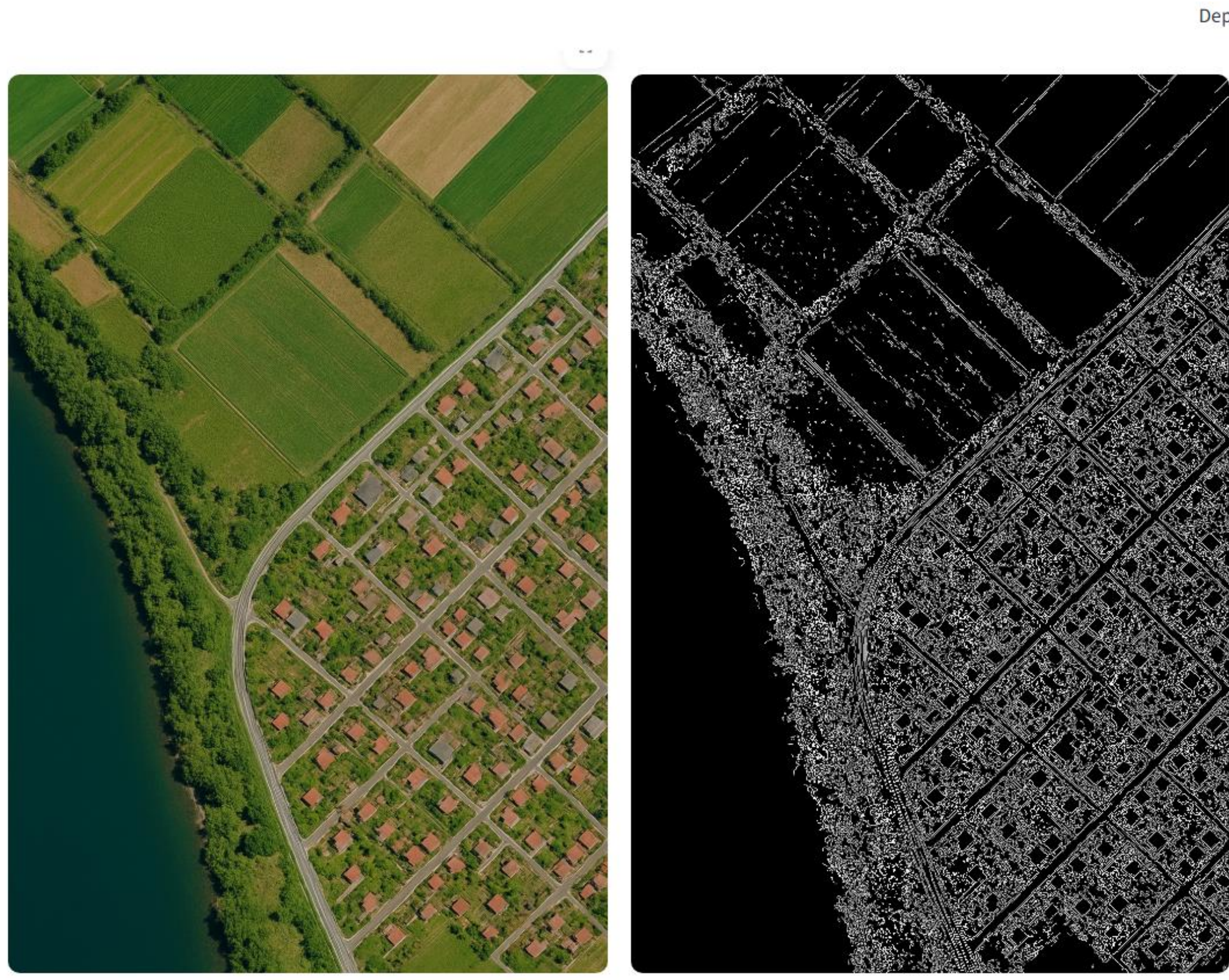
The final day of the reporting period was dedicated to the analysis and interpretation of the results obtained through the refined building footprint extraction process. The team summarized the key findings, including the total number of building footprints extracted, their spatial distribution across the study area, and any associated attributes, such as the confidence scores assigned to each detection. These results were then interpreted in the context of the project's overarching objectives, aiming to understand the significance of the findings and their implications for the project's goals.

While the initial research material did not explicitly define specific metrics for evaluating the accuracy and completeness of the building footprint extraction, the team likely considered potential metrics that could be used for future evaluations. Common metrics in this domain include Intersection over Union (IoU), Precision, Recall, and F1-Score. These metrics provide a quantitative measure of the overlap and agreement between the extracted building footprints and any available ground truth data.

In addition to the result analysis, the team initiated the process of compiling the project report, which this document represents. This involved structuring the report according to the planned outline, ensuring that all key activities, findings, and insights from the reporting period were documented in a clear and organized manner. The report structure included an Executive Summary to provide a high-level overview, a week-wise breakdown of the project's progression, and dedicated sections for summarizing overall performance, highlighting key observations, and providing recommendations for future work. Adherence to best practices for project report writing, emphasizing clarity, conciseness, accuracy, and appropriate formatting, was a guiding principle in the initial compilation of the report.

Overall Performance and Key Metrics:

Over the reporting period from April 15th to May 6th, the project demonstrated significant progress towards its inferred objective of geospatial data analysis, likely focusing on building footprint extraction and interactive visualization. The team successfully established a consistent development environment using Docker, explored the fundamental capabilities of the geoai-py library, and implemented the core task of building footprint extraction using the BuildingFootprintExtractor module. Furthermore, substantial effort was dedicated to developing a Streamlit



Original Image

Edge Detection

application to visualize and potentially interact with the extracted data. The application saw enhancements in terms of user interactivity and initial considerations for performance optimization were made.

While specific key performance indicators (KPIs) were not explicitly defined in the initial research material, several potential metrics could be considered for future evaluation of the project's success. These include the number of building footprints extracted, the accuracy of these extractions when compared against ground truth data (using metrics such as Intersection over Union, Precision, and Recall), the performance of the Streamlit application in terms of load times and responsiveness, and the geometric quality of the extracted vector building footprints, potentially measured by their adherence to orthogonality or regularity.

Insights and Observations:

The project's progression reveals several important considerations for geospatial data analysis and the development of interactive visualization tools. The recurring emphasis on handling computationally intensive tasks within the Streamlit environment underscores the inherent computational demands associated with geospatial AI applications like building footprint extraction. This necessitates careful architectural planning and the adoption of strategies such as offloading computations or employing caching mechanisms to ensure a responsive user experience. The reliance on pre-trained models within the geoai-py library appears to have been a significant factor in accelerating the initial development, allowing the team to achieve tangible results relatively quickly. This highlights the value of leveraging existing AI models for specific geospatial tasks. Furthermore, the choice of Streamlit as the platform for building the interactive web application demonstrates the library's effectiveness in enabling rapid prototyping and facilitating the creation of user-friendly interfaces for exploring complex data. Its low barrier to entry allows developers with data science and machine learning expertise to quickly translate analytical outputs into interactive visualizations.

The iterative nature of the project's development, moving from environment setup to tool exploration, implementation, and visualization, reflects a common approach in software development and data science. Each week built upon the progress of the previous one, allowing the team to gradually refine the analytical process and enhance the user interface. The potential need to balance the accuracy of the building footprint extraction with the computational performance of the application is also evident. Achieving higher accuracy often requires more complex processing, which can impact the application's speed and responsiveness. Therefore, finding an optimal trade-off is crucial. The project's dependence on the quality of the input raster imagery for accurate building footprint extraction highlights the fundamental importance of data quality in AI-driven geospatial analysis. The resolution, clarity, and presence of any obstructions in the imagery can significantly influence the reliability of the results. Finally, the entire process underscores the continuous learning and adaptation required in this rapidly evolving field. The team's ability to effectively utilize new libraries like geoai-py and to address technical challenges as they arise is essential for the project's ongoing success.

Recommendations and Next Steps:

Based on the progress and insights gained during this reporting period, the following recommendations and next steps are proposed:

Short-Term:

- Continue to refine the building footprint extraction process by systematically tuning parameters to achieve optimal accuracy and efficiency for the specific geospatial datasets being used in the project.
- Further enhance the Streamlit application by incorporating additional interactive features that allow users to explore the data in more detail, such as filtering building footprints based on various attributes (e.g., confidence scores, area) and visualizing different evaluation metrics once they are implemented.
- Prioritize the implementation of a robust strategy for handling computationally intensive tasks within the Streamlit application. This could involve exploring background processing techniques or deploying the extraction process as a separate API that the Streamlit application can call asynchronously.
- Conduct a formal evaluation of the accuracy and completeness of the extracted building footprints using appropriate geospatial metrics, such as Intersection over Union (IoU), Precision, Recall, and potentially metrics related to the geometric quality of the extracted vectors.

Mid-Term:

- Explore the potential integration of other relevant functionalities offered by the geoai-py library, such as object detection for identifying other features of interest in the geospatial data or image segmentation techniques for delineating different land cover types.
- Consider deploying the Streamlit application to a cloud-based platform to enable wider accessibility for stakeholders and end-users.
- Develop a comprehensive plan for managing and versioning both the input geospatial data used in the project and the resulting extracted building footprint data to ensure data integrity and reproducibility.

Long-Term:

- Investigate the feasibility of fine-tuning the pre-trained machine learning models used by the geoai-py library with project-specific geospatial data. This could potentially lead to significant improvements in the accuracy and performance of the building footprint extraction process for the specific study areas of interest.
- Develop a detailed documentation strategy for the entire project, encompassing the data analysis pipelines, the architecture and functionality of the Streamlit application, and any custom code that has been developed. This documentation will be crucial for knowledge transfer, future maintenance, and ensuring the long-term reproducibility of the project's work.
- Begin to formulate a plan for archiving the software project, including the code, data, documentation, and environment configurations, following best practices for software and geospatial data archiving to ensure long-term preservation and accessibility for future research or use.