USFGeoAI: Applying LLMs and Foundation Models to Assist in Image Retrieval

Hana Gamracy[1], Paramdeep Sodhi[1], George Sphicas[1], Divya D'Souza[1], Puranjai Garg[1],
Noga Gottlieb[1], Sahil Sanjay Gupta[1], Rucha Maslekar[1], Atanas Patterson[1], Noah Steaderman[1],
William Stout[1], Fernanda Lopez Ornelas[2], David Saah[2], David Guy Brizan[3] and Christopher Brooks[3]

[1] Department of Computer Science, University of San Francisco, 2130 Fulton St., San Francisco, CA 94117 USA
[2] Department of Environmental Science (faculty), University of San Francisco, 2130 Fulton St., San Francisco, CA 94117 USA
[3] Department of Computer Science (faculty), University of San Francisco, 2130 Fulton St., San Francisco, CA 94117 USA

Keywords: GIS, Geospatial Data, Environment, Generative AI, GeoAI

Abstract

Geospatial analysis is often hampered by two major obstacles: a scarcity of high-quality, annotated satellite imagery, and the complexity of interacting with machine learning tools for image analysis. To address this, we introduce USFGeoAI, a proof-of-concept system that allows users to query, retrieve, and analyze high-resolution imagery using natural language interaction and direct processing of images. The system incorporates IBM-NASA's Prithvi Foundation Model for supervised detection of environmental features and the Clay Foundation Model for unsupervised similarity search when detectors are unavailable. An interactive interface allows users to search for features (such as swimming pools, vegetation changes, and burn scars), apply detectors to TIFF images, and explore new regions for model training.

1. Introduction

Current efforts to integrate machine learning (ML) techniques into Geospatial Analysis tools are hampered by two problems: access to large, high-quality satellite images, and the challenges for end users in integrating ML tools into their workflow. In order to address these issues, we have developed an end-to-end system, USFGeoAI. USFGeoAI incorporates *foundation models* in order to avoid the problem of access to large datasets, and uses an LLM-based tool calling architecture to allow users to create their own detectors, which can then be accessed via a natural-language chatbot interface. We also show how foundation models can be used in a semi-supervised way to identify similar images; this allows our system to respond to user queries for novel features for which detectors do not exist. This system serves as a proof-of-concept for a flexible and scalable geospatial AI pipeline from data discovery to model validation.

2. Related Work

Large Language Models are extremely versatile tools that have been employed across many different application areas related to geospatial analysis, including benchmarking, natural-language interfaces, extracting geolocation data from text, disaster prediction, and inference and display of environmental variables. Table 1 captures some of their uses in geospatial applications.

Benchmarking	LaCoste, et al (Lacoste et al., 2023)
Natural Language UI	GeoForge (Akinboyewa et al., 2024), ChatGeoAI (Mansourian and Oucheikh, 2024)
Multimodal integration	Yin, et al (Yin et al., 2025)

Table 1. Uses of LLM technology in geospatial applications

A common challenge in working with foundation models is the need for common benchmarking tasks that can allow developers to measure and compare performance. To address this, Lacoste et al.(Lacoste et al., 2023) introduce GEO-Bench, a benchmark designed to evaluate Earth monitoring models through six classification and six segmentation tasks, providing a consistent evaluation and clear reporting of results. Earth observation datasets such as Sentinel-2 and Landsat 8 offer structured multispectral images with periodic revisits, while complementary data sources like Synthetic Aperture Radar (SAR) enable deeper analysis by providing visibility through cloud cover, and referenced text data from Wikipedia and OpenStreetMap enrich multimodal learning. GEO-Bench was developed by refining diverse geospatial datasets, ensuring accessibility, usability, and effective model assessment, covering multiple sensing modalities. Experiments reveal that larger downstream datasets reduce the ability to distinguish between similarly performing models, and pre-training is less beneficial for tasks with extensive training sets. A smaller benchmark allows faster downloads, faster results, and lower computational costs, making evaluations more efficient. GEO-Bench optimizes evaluation by limiting hyperparameter tuning, employing early stopping based on validation metrics, and restricting augmentations to 90-degree rotations and flips. GEO-Bench aims to advance foundation models for Earth monitoring, assess performance, identify optimal pre-trained models, and ensure transparent evaluation."

Many researchers have found that LLMs, typically deployed as chatbots, can provide users with an improved interface. Our system demonstrates this feature. This is also seen in GeoForge (Zhang et al., 2024). GeoForge integrates LLMs as a front-end chatbot for a web-based GIS platform. This work emphasizes the integration of LLMs into GIS platforms, the importance of human-centered design, and the development of accessible geospatial analysis tools as key approaches to building user-friendly GeoAI / GIS chatbots. GeoForge introduces "GIS Copilot," which combines Large Language Models with QGIS to enable natural language spatial analysis. It autonomously generates workflows and code, simplifying GIS tasks for users with minimal expertise. Unlike our system, it interfaces with existing mapping platforms, rather than allowign users to create custom detectors. Evaluations show high success in basic and intermediate tasks, with ongoing challenges in complex analyses. This approach exemplifies the potential of generative AI to transform traditional GIS workflows.

ChatGeoAI (Mansourian and Oucheikh, 2024) is a similar system that uses an LLM to provide a rich user experience. It has two selectable modes: single prompt, where a user enters one query about trying to find a specific type of location, and chat mode, where users can continue the conversation past their initial query in order to further interrogate the data. As with GeoForge, the emphasis is on providing a simple interface to an existing GeoAI system, as opposed to allowing users to generate their own similarity-based detectors. ChatGeoAI's purpose is to make complex geospatial analysis more easy to understand for users with limited GIS experience by using NLP for request interpretation and Generative AI to create executable code for geospatial analysis. They fine-tuned the model on PyGIS in order to "enhance its ability to learn a wide range of potential operation combinations." Using the CodeBLEU metric to compare performance, they found that at the character level ChatGeoAI outperformed the baseline with a ChrF score of 26.43 percent. In comparison, the baseline score was 23.86 percent. Their performance analysis revealed that ChatGeoAI has a failure rate of 75 percent. The team identified that LLMs struggle with GIS analysis code generation and cite the need for iterative tests and refinement.

LLMs, in conjunction with multimodal models that can process both text and imagery, have also been effective in extracting geospatial information from unstructured text and images. Many emerging use cases for LLMs in the geospatial AI space involve environmental disaster prevention and safety. For situations in which people are facing an environmental disaster, a geolocalization approach has been created that integrates LLM-enhanced disaster geolocalization by leveraging implicit geoinformation from multimodal data, specifically in disaster response scenarios (Yin et al., 2025). Their system extracts geolocation data from both images and text by using GPT-4o, combined with geocoding and retrieval-based map services to enhance location accuracy. The method achieves 81.45% accuracy within 161 km and 44.95% within 1 km when tested on a Twitter dataset on Hurricane Harvey. They compare their approach against several NLP-based toponym recognition models (Flair, Stanford CoreNLP, spaCy, and NLTK) and traditional geocoding services (Google Maps, Geoapify, and Nominatim), demonstrating that LLM-assisted multimodal integration significantly improves disaster-related geospatial analysis. However, their study highlights the limitations of implicit geolocation inference from images, citing challenges in resolving ambiguous toponyms and the need for better multimodal fusion techniques. Our system uses a simple version of this, using an LLM to attempt to extract named entities from user queries and then calling Google Maps to get the corresponding latitude/longitude coordinates.

Apart from LLMs, there has also been a great deal of work on integrating foundation models and other deep learning approaches to classify geospatial data according to hazard. Jena et al.(Jena et al., 2021) discuss a CNN-based module for earthquake classification and assessment in Northeast India. This study integrated convolutional neural networks and GIS to assess earthquake probability in Northeast India. The data was collected from the seismically active region of India. The authors used GIS-thematic layers, a map layer that represents specific geographic features such as fault lines. On that data, a CNN model was applied to determine the likelihood of an earthquake. The CNN model was built from 4 convolutional and pooling layers and trained in a manner typical of ML systems: dividing the data into train, validation and test and training to maximum performance on the validation set. The model is 94% accurate, identifying multiple cities as high-risk zones with low "coping capacity," meaning the city's infrastructure may be inadequate to withstand a natural disaster.

Similarly to Jena et al.(Jena et al., 2021), Haq et al. (Haq et al., 2022) uses AI for forecasting the future of environmental variables. They utilize Long Short Term Memory (LSTM), which is a type of Recurrent Neural Network (RNN) capable of learning long-term dependencies. LSTM is designed to evade the problem of vanishing gradient problems and extracts patterns from time-series data. They used the Moving Forward Window technique to forecast environmental variables for seven years in the future. Their research specialized in developing and implementing LSTM models on many combinations of parameters to increase the efficiency and stability of the model. Rather than predicting the effects of natural disasters, this team uses these models to understand patterns and perform long-term predictions. The data, which were collected from the Himachal Pradesh region, included temperature in Kelvin, snow cover, and vegetation in pixels from 2001 to 2017. Their work is similar to our system because we also analyzed specific environmental characteristics using detectors that locate and locate specific features.

New approaches are being studied for predicting outdoor particulate matter (PM) concentrations in Southern Taiwan using an ensemble learning model. Lu et al. (Lu and Lee, 2025) take the use of LLMs further by offering a new method for finding a relationship between multiple environmental characteristics at once, leading to a model with high predictive accuracy. Given the critical impact of PM10 and PM2.5 on public health, the study integrates multiple datasets from EPA monitoring stations, AirBox sensors, and meteorological, land-use, traffic, and geospatial data collected between 2018 and 2020. The study uses ensemble modeling, combining land-use regression (LUR), inverse distance weighting, and three machine learning algorithms—support vector machines (SVM), random forest (RF), and multilayer perceptron (MLP)—to improve PM prediction accuracy. The ensemble model achieved high predictive accuracy surpassing previous air quality modeling efforts. Results indicate that meteorological factors like temperature, relative humidity, and wind speed, along with emission sources like traffic, industrial areas, and temple burning activities influence PM levels. The authors highlight that integrating real-time sensor data and land-use information can lead to the development of early warning systems and more effective air pollution mitigation measures. The study sets a precedent for utilizing machine learning in geospatial air quality modeling.

The systems described in these works could be improved upon by having access to high-quality satellite imagery for testing, a limitation that the geospatial analysis community is facing as they are difficult to acquire at such great volumes needed to accurately train models. We have developed a GeoAI system that streamlines the geospatial analysis pipeline, allowing non-expert users to perform environmental data analysis. GIS tools and remote sensing analysis are abstracted from the user which gives them the opportunity to quickly create and assess new detectors through our unsupervised learning approach. Our system offers the solution to the industry limitation of detector creation, without the need for heaps of satellite imagery.

3. System Architecture

Our system is constructed to provide users with two features: the ability to create detectors that will identify plots containing features of interest, and the ability to enter natural-language queries invoking these detectors. Figure 1 illustrates how a user can make use of the detector system.

Users can interact with the system either by providing an image and asking for similar images, or by providing a natural language prompt. To begin, we illustrate how a user can utilize our system to find related imagery.

We began by creating three specialized detectors (burn scar, crop field, or flood plain) by fine-tuning the Prithvi foundation model. At classification time, the user uploads a TIFF image through a React + Next.js frontend. This image is then processed via a FastAPI backend. The back end then uses a tool-calling LLM to route the image is routed to one of the pre-trained detectors selected by the system using the closest available match.

This works well, but has a significant limitation: the user must know in advance what features they are interested in locating and create a detector to recognize those features. In order to address this limitation, we also incorporate unsupervised nearest-neighbor similarity using the Clay foundation model.

In those cases where the system has a detector that yields a high-confidence match, results are packaged and returned to the user. In cases where a detector fails or confidence is low, the system falls back to a similarity-based retrieval approach using the Clay foundation model, which compares embeddings against a labeled vector store. This dual-path design enables robustness: specialized detectors provide high accuracy on known tasks, while Clay generalizes to novel or unsupported queries. This allows our system to support an unlimited number of queries, and supports usability for non-expert users and reliability in diverse geospatial scenarios.

4. Foundation Models Utilized

4.1 Clay Foundation Model

The Clay Foundation Model (Brown et al., 2025) is a large, unsupervised foundation model designed to identify patterns and similarities across diverse datasets. Rather than relying on labeled data, it learns representations by analyzing raw input from various modalities, such as images or text, and finding structure within the data itself. This unsupervised approach allows the model to generalize well across tasks and domains without the need for task-specific fine-tuning. Clay excels at similarity-based reasoning, making it effective for detecting changes, clustering related items, and retrieving semantically similar content. Its versatility stems from being pre-trained on a wide range of datasets, enabling robust performance even in unfamiliar or complex environments. As a result, the Clay Foundation Model is particularly well-suited for applications in fields like geospatial analysis, where labeled data is scarce but pattern recognition is critical. In our work, we leveraged Clay to build a swimming pool detector in Los Angeles and as a fallback mechanism in our GeoAI application when a specialized detector for a given image is unavailable.

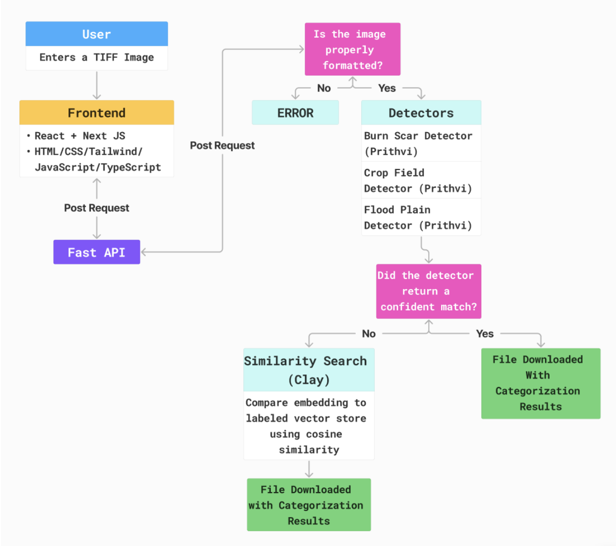

Figure 1. System Design of image processing. The user uploads a TIFF image through our frontend. The image is passed to our backend, which then uses the Groq API to verify formatting. If valid, the image is processed by one of our three working detectors and if not, the system falls back to an unsupervised approach using image embeddings.

4.2 Prithvi Foundation Model

IBM-NASA’s Prithvi Foundation Model (Jakubik et al., 2023) is a large-scale self-supervised geospatial transformer that can model spatial, temporal, and spectral patterns in multi-spectral satellite data. Prithvi is built upon a Vision Transformer Architecture through the use of a Masked AutoEncoder learning strategy. The model is trained to reconstruct masked portions of multi-temporal Harmonized Landsat-Sentinel 2 imagery, giving it the ability to learn rich representations without the need for labeled annotations. Prithvi differs from traditional models as it excels in labeling different areas of land cover types and detecting dynamic environmental events, including burn scars, floods, and other significant landscape changes. The model processes multiple spectral bands, including Blue, Green, Red, Narrow NIR, SWIR 1, and SWIR 2, converting these into high-dimensional feature vectors that represent important geospatial characteristics. These vectors encapsulate key spatial, temporal, and spectral information, allowing Prithvi to perform precise classification tasks and detect temporal changes. Pre-trained on extensive datasets, Prithvi leverages these feature vectors to enable high-resolution land analysis and provide effective monitoring and detection capabilities for a wide range of geospatial applications.

5. Integrating LLMs

Two challenges in working with geospatial data are supporting non-expert users and automating multi-step workflows. In order to address this, we provide a natural-language interface using a large language model (LLM), along with tool-calling capability using the Groq API to allow the LLM to automatically select and call detectors.

An example use case of our system design is as follows:

1. The user enters a query, for example: *”Show me palm oil plantations in Indonesia.”*

2. If a relevant detector exists:
   - The system returns images and coordinates of areas in Indonesia containing this environmental feature, marked with pins on an interactive map.
3. If no relevant detector exists:
   - The system responds with a list of sample coordinates (e.g., *“Here are 10 XY coordinates of this land feature.”*)
   - The system provides corresponding images for these locations.
   - The user verifies which images are correct by selecting **Yes** or **No**.

The system is capable of identifying specific land features, retrieving their images and coordinates through the Query-Processing mode, and applying Prithvi detectors to uploaded TIFF images. This streamlined workflow allows users to search and analyze environmental data without writing any code, while also laying the foundation for faster training of future detectors.

The system operates in two primary modes:

1. **Image-Processing Mode:** Enables users to apply or test detectors on `.tiff` images directly within the platform, supporting the development and refinement of custom environmental detection models.
2. **Query-Processing Mode:** Allows users to enter a query to find and download satellite images of specific land features using AI and geospatial tools.

5.1 Image-Processing

To build our detectors, we used the Clay foundation model together with the Prithvi model for image processing. Figure 1 showcases the image rasterization process that we perform using Clay. We created our own detectors which are able to analyze a .tiff image, identify the desired environmental features, perform an image mask, and return the image with only the desired feature highlighted to the user. HuggingFace datasets were used for model training. In Figure 1 we provide multiple images of pools. The top panel of each pair shows raw Sentinel-2 imagery at multiple time points, while the bottom panel displays the detector’s output mask, with swimming pools highlighted in blue and non-pool areas masked in yellow. This temporal visualization demonstrates the model’s ability to consistently identify pools across different dates and environmental conditions, enabling both static feature mapping and change detection over time.

Figure 3 illustrates another detector that we created: Vegetation with Heat Proxy. The images selected are of the same area throughout the changing seasons, illustrating the changes in vegetation. The left panel shows raw detector outputs, where healthy vegetation is highlighted in green and non-vegetated areas are masked in yellow. The right panel displays NDVI maps, using a red-yellow-green colormap, in which green indicates high vegetation health (cooler, more moisture-rich areas) and red indicates low NDVI values (warmer, drier areas). This paired view illustrates seasonal vegetation changes and highlights the detector’s capability to monitor environmental variations over time.

The system currently supports three fully functioning detectors, trained on HuggingFace datasets, which can be applied to user-uploaded TIFF images. The HuggingFace burn scars, vegetation, and flood plain detection datasets were used to train our model.

Our system was designed with flexibility at its core. Users can add custom detectors and upload their own imagery, streamlining the development and testing of new models. By handling all back-end configuration for detector execution, the platform abstracts away complex code dependencies and environment setup, allowing researchers to focus solely on experimentation. However, a fundamental challenge emerged: maintaining functionality when no satellite imagery or pre-trained detector exists for a target object. To address this, we incorporate an interactive, user-driven workflow. In this mode, users upload a TIFF image through a React + Next.js front-end which is then processed by a FastAPI back-end. The backend uses the Groq API to validate image formatting before running it through available detectors (e.g., Burn Scar, Crop Field, Flood Plain). If no confident prediction is produced, the system defaults to an unsupervised approach, embedding the image and using Clay’s similarity search to assign a probable category label. The result is returned as a downloadable file, ensuring that even in the absence of a specialized detector the system can deliver meaningful, actionable outputs.

While this system significantly accelerates the reuse of detectors, a fundamental challenge remains: many high-quality satellite images for specific environmental features are not easily accessible to the general public due to licensing, cost, or availability restrictions. To address this, we provide users with a complementary solution, our query-processing mode, which enables open-ended exploration using natural language input to locate and retrieve publicly accessible imagery whenever possible.

5.2 Query-Processing

The Query-Processing mode of our system leverages the Groq API (Abts et al., 2022) and Google Earth Engine (Gorelick et al., 2017) to process user queries and identify specific land features. The system displays interactive pins on a map or globe. Users can click a pin to download an image of the selected area, as seen in figure 5, which can then be processed in the image-processing module. The Homepage User Interface, seen in Figure 4, includes a search form for entering queries, serving as the starting point of the process. Users can input text queries, which are then processed by Groq to generate a list of coordinates for matches. These coordinates are fed into Google Earth Engine, which returns a corresponding TIFF image of the queried location, seen in Figure 6. The user is then able to view and interact with the map to pin specific locations, which can be downloaded for further image processing.

This approach not only streamlines the search process but also ensures that users can quickly locate publicly accessible satellite imagery without needing specialized GIS expertise. By combining Groq’s processing capabilities with Google Earth Engine’s vast geospatial dataset, the system lowers the barrier to environmental analysis and broadens access for researchers, policymakers, and the public. Additionally, it provides a foundation for integrating more data sources in the future, further expanding the range of features and regions that can be explored.

Figure 2. Swimming Pool Detection - Here we provide multiple images of pools, which our system highlights in blue, masking the rest of the image without pools in yellow.

Figure 3. Vegetation Detector with Heat Proxy - Using our heat proxy detection, we are able to identify and mask out the healthy vegetation. This experiment demonstrates the complex detections that can be completed using our system.

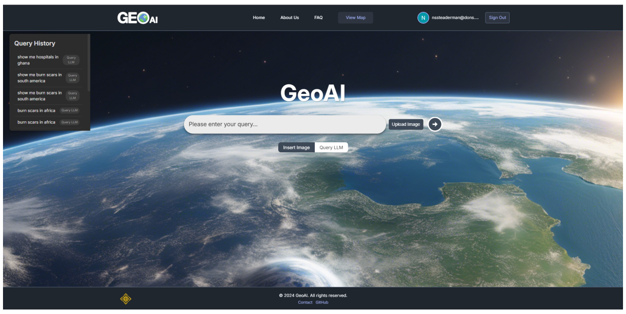

Figure 4. Homepage User Interface

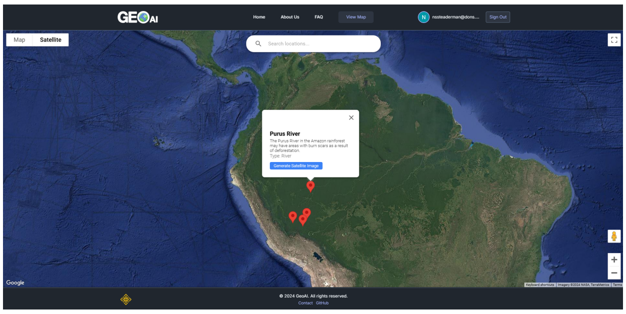

Figure 5. Map Pinning System

To evaluate the accuracy of our system in identifying real-world locations (e.g., parks or pools), we compared its predicted outputs against manually verified ground-truth datasets.

We used the following metrics:

- **True Positives (TP):** Locations correctly identified by the model within a threshold radius (e.g., 50 meters) of an actual target feature.
- **False Positives (FP):** Predicted locations where no target feature exists.
- **False Negatives (FN):** Actual features missed by the model.

From these metrics, we calculate precision, recall, and F1.

As a demonstration, we queried the number of golf courses in San Francisco and compared the results with our system's predictions. This methodology can be scaled to assess the overall precision and recall of our system across hundreds of landmarks. The ability to quickly assess the performance of our system using these metrics accelerates the process of fine-tuning new detectors.

In evaluation, the system achieved a precision of 0.84, recall of 0.8, and an overall F1 score of 0.81, demonstrating reliable detection performance in various geospatial features.

Beyond location discovery, the query-processing mode also serves as a streamlined pipeline for acquiring imagery that can be analyzed in the image-processing module. Once a user identifies and downloads a TIFF image through query-processing, it can be directly uploaded into the image-processing mode, where detectors such as Burn Scar, Crop Field, or Flood Plain can be applied. This integration ensures that even individuals without prior GIS expertise can progress from a natural language search to detailed feature detection in just a few steps, bridging the gap between finding relevant satellite imagery and conducting meaningful geospatial analysis.

Figure 6. Example of a raw satellite image (TIFF) that can be created by query-processing and can be used by image-processing for land feature detection.

Addressing Limitations with Clay

Currently, the system supports only three detectors: floodplain identification, burn scar detection, and vegetation loss. This is due to the limited availability of sufficiently labeled satellite imagery for training.

To address this constraint, the system incorporates an unsupervised learning approach using Clay. When reliable detection is not possible using Prithvi, an image embedding is constructed and a similarity search is performed against a labeled vector store managed by Clay. The most similar match is then used to assign a category to the image. While this approach is less accurate than supervised detection, it enables broader categorization without requiring extensive labeled training data. This hybrid workflow allows the system to operate even in low-data regimes, leveraging prior knowledge from similar imagery. Additionally, it provides a scalable path for adding new categories in the future, as more labeled examples become available.

Scalability

The underlying architecture has been designed with scalability in mind. The modular API-driven structure enables the seamless integration of new detectors as more satellite imagery becomes available. Scaling does not require redesigning the backend or frontend; it simply involves adding new detector models and extending the existing processing pipeline. In addition, the query-processing mode expands scalability beyond detectors by ensuring a steady and accessible supply of relevant imagery. By allowing users to locate and download publicly accessible TIFF images that can be fed directly into the image-processing module, the system reduces dependence on pre-curated datasets and empowers users to contribute to the detection process. This dual-mode design, combining open-ended imagery acquisition with modular detector integration, ensures that the platform can rapidly grow its analytical capabilities while remaining adaptable to evolving environmental monitoring needs.

6. Results

Our system successfully integrates user-driven geospatial analysis with machine learning-based image detection. A key component developed during the course of this project is a RESTful

API that enables end-to-end satellite image processing based on natural language queries.

- **User Query and Image Processing** The system accepts both a satellite image and a user query describing the target land feature (e.g., "show me areas with swimming pools"). This query is interpreted by a large language model (LLM), which identifies the most appropriate Prithvi-based detector for the task.
- **Model Selection and Inference:** Once the detector is selected, the API executes the corresponding model on the provided image. The output is a masked image, with detected regions visually highlighted. This demonstrates the system's ability to translate high-level user input into precise geospatial inference.
- **Frontend Interface** To support real-time user interaction, we developed a Streamlit-based front-end that allows users to upload TIFF satellite imagery, enter custom queries, and receive processed results in seconds. This interactive Web application streamlines the GeoAI workflow and significantly reduces the barrier to entry for environmental data analysis. We successfully deployed an interactive website built with Next.js 13, TypeScript, and integrated the latest geospatial analysis tools.

7. Conclusion

This paper presents a proof-of-concept of a AI-driven system for geospatial analysis, leaveraging large language models to streamline user queries in two modalities: natural language queries and image similarity. By enabling non-expert users to analyze environmental features through natural language, our system removes the steep learning curve often associated with GIS tools and remote sensing analysis.

Our integration of the Prithvi model, LLM-driven query understanding, and a visual Streamlit interface demonstrates that complex geospatial workflows can be democratized and made accessible without sacrificing performance. The results confirm the feasibility of building responsive, intelligent, and interpretable systems for detecting land features on a scale.

By utilizing our system, there is a streamlined and simple process to inputting more detectors - the only thing a user would have to do is upload it to our library. Opening our system up to others will allow the development and testing of detectors to be more efficient. In addition, our query-processing mode addresses the limitations of detector usage. Users can use our system to find more examples of unique environmental features, retrieve the coordinates and images of the matches, verify the accuracy of the input, and then use these images for model training. These features capture how our system can push the environmental detection industry further than before.

8. Future Work

Practical Learnings and Extensions from Capstone Implementation

During the development of our proof-of-concept in a senior capstone project, we encountered real-world limitations in applying GeoAI at scale. Two primary obstacles emerged:

1. **Limited High-Quality Imagery** Access to consistent, high-resolution satellite imagery was sparse and geographically uneven.
2. **Improper Data Formatting** Available imagery datasets often required extensive preprocessing before they could be used in model training pipelines.

In response, we explored two complementary directions:

1. **Synthetic Imagery via LLMs:** We began exploring the feasibility of using large language models to generate synthetic satellite imagery, a promising method for overcoming data scarcity. Although time constraints prevented full integration, this remains a compelling area for future exploration.
2. **Coordinate-Based Image Retrieval:** As a more immediate workaround, a Python script capable of retrieving satellite imagery for a single coordinate via Google Earth Engine was implemented. The image output adhered to the formatting standards required for model ingestion. Although scaling this solution to support multiple sites remains a future goal, the working proof-of-concept validated the technical feasibility of the pipeline.

In addition, work on the development of foundation models for geospatial inference has continued rapidly, allowing for the integration of more sophisticated datasets. The capability of agentic LLMs has also been rapidly increasing; we anticipate that this will allow us to add more sophisticated reasoning, analysis and workflow automation to future versions of our tool.

References

Abts, D., Kim, J., Kimmell, G., Boyd, M., Kang, K., Parmar, S., Ling, A., Bitar, A., Ahmed, I., Ross, J., 2022. The groq software-defined scale-out tensor streaming multiprocessor: From chips-to-systems architectural overview. *2022 IEEE Hot Chips 34 Symposium (HCS)*, IEEE Computer Society, 1–69.

Akinboyewa, T., Li, Z., Ning, H., Lessani, M. N., 2024. Gis copilot: Towards an autonomous gis agent for spatial analysis.

Brown, C. F., Kazmierski, M. R., Pasquarella, V. J., Rucklidge, W. J., Samsikova, M., Zhang, C., Shelhamer, E., Lahera, E., Wiles, O., Ilyushchenko, S. et al., 2025. AlphaEarth Foundations: An embedding field model for accurate and efficient global mapping from sparse label data. *arXiv preprint arXiv:2507.22291*.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R., 2017. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202, 18–27.

Haq, M. A., Ahmed, A., Khan, I., Gyani, J., Mohamed, A., Attia, E.-A., Mangan, P., Pandi, D., 2022. Analysis of environmental factors using AI and ML methods. *Scientific Reports*, 12(1), 13267.

Jakubik, J., Roy, S., Phillips, C., Fraccaro, P., Godwin, D., Zadrozny, B., Szwarcman, D., Gomes, C., Nyirjesy, G., Edwards, B. et al., 2023. Foundation models for generalist geospatial artificial intelligence. *arXiv preprint arXiv:2310.18660*.

Jena, R., Pradhan, B., Naik, S. P., Alamri, A. M., 2021. Earthquake risk assessment in NE India using deep learning and geospatial analysis. *Geoscience Frontiers*, 12(3), 101110.

Lacoste, A., Lehmann, N., Rodriguez, P., Sherwin, E., Kerner, H., Lütjens, B., Irvin, J., Dao, D., Alemohammad, H., Drouin, A. et al., 2023. Geo-bench: Toward foundation models for earth monitoring. *Advances in Neural Information Processing Systems*, 36, 51080–51093.

Lu, Q.-O., Lee, C.-C., 2025. Innovative Geo-AI model: An enhance of outdoor PM estimations based on land use and outdoor environmental factors in a highly polluted area. *Chemosphere*, 373, 144178.

Mansourian, A., Oucheikh, R., 2024. ChatGeoAI: Enabling geospatial analysis for public through natural language, with large language models. *ISPRS International Journal of Geo-Information*, 13(10), 348.

Yin, W., Xue, Y., Liu, Z., Li, H., Werner, M., 2025. LLM-enhanced disaster geolocalization using implicit geoinformation from multimodal data: A case study of Hurricane Harvey. *International Journal of Applied Earth Observation and Geoinformation*, 137, 104423.

Zhang, A., Gao, S., Li, Z., Janowicz, K., 2024. On the Opportunities and Challenges of Foundation Models for Geospatial Artificial Intelligence. *ACM Transactions on Intelligent Systems and Technology*, 15(4), 1–21.