Remote Sensing Application:
ChatGPT **and Visual Models Supported** Remote Sensing **Tasks**

MUSA 6500_HW2_Shuya Guan

Guo et al. (2024). Remote Sensing ChatGPT: Solving Remote Sensing Tasks with ChatGPT and Visual Models.

What is Remote Sensing ChatGPT?

Exploring LLM-Powered Remote Sensing Applications

◈ Current Challenges in Remote Sensing Analysis

Complexity of Interpretation: Manual processing by experts is often required	Lack of Automation: Traditional workflows require multi-step, manual execution	Barriers for Non-Experts: High technical expertise needed for AI-based remote sensing

◈ Concept

Remote Sensing ChatGPT is an intelligent geospatial agent that integrates a large language model (LLM) (e.g., ChatGPT) with remote sensing AI models to enhance geospatial data processing and interpretation. Unlike traditional GIS workflows that require specialized technical skills, this system enables users to query and analyze remote sensing data using natural language.

Recent advancements, such as Qiusheng Wu's work on geemap and leafmap (Python libraries for Earth Engine and geospatial visualization) and Development Seed's AI-driven remote sensing tools, have demonstrated the potential of integrating LLMs with geospatial AI to streamline geospatial data analysis.

What Does Remote Sensing ChatGPT Do?

Exploring LLM-Powered Remote Sensing Applications

💡 Use Case

Users can directly upload remote sensing images and task requests, and Remote Sensing ChatGPT will automate the interpretation process, reducing the need for manual image analysis.

✔ Urban Planning – Analyzing land use changes, detecting unauthorized developments
✔ Disaster Monitoring – Identifying flood-affected areas, assessing wildfire impact
✔ Environmental Assessment – Tracking deforestation, monitoring water quality

Fig1. Leafmap and Geemap

💡 Geospatial AI Tool Integration

Remote Sensing ChatGPT leverages leading geospatial AI tools for data visualization and automation:

◈ geemap – Facilitates interactive geospatial analysis with Google Earth Engine
◈ leafmap – Enables Python-based geospatial visualization and analysis

These tools, along with LLM-powered query processing, provide a low-code approach to automated remote sensing analysis, making geospatial intelligence more accessible and actionable.

Process: How ChatGPT Executes Remote Sensing Tasks?

Exploring LLM-Powered Remote Sensing Applications

◈ Key Technologies

✔ Prompt Engineering – Optimized queries improve accuracy (e.g., structured task breakdown).

✔ Vision Prompting – AI extracts and interprets visual data from remote sensing images.

✔ Multi-Step AI Processing – Tasks are executed sequentially (e.g., segmentation → vectorization → object counting).

◈ Workflow:

Generate Prompt	→	Task Planning	→	Task Execution	→	Generate response
Input a geospatial query		AI maps the query to predefined modules (land use classification, object detection).		Segmentation: Extracts objects Vectorization: Converts features into shapes. Counting: Identifies and quantifies objects.		AI compiles and summarizes results

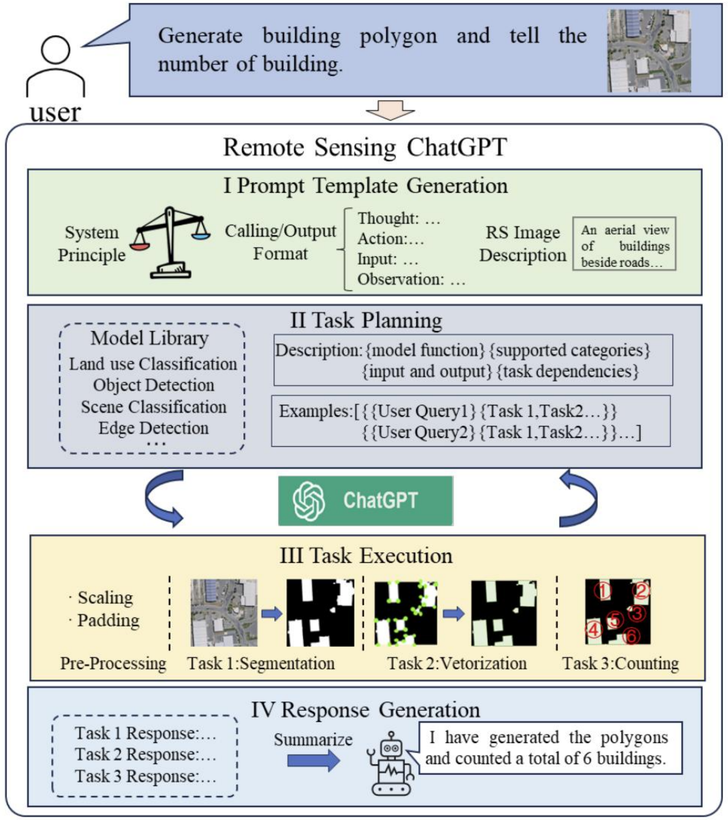

Fig2. Remote Sensing ChatGPT Work Flow(Guo et al., 2024)

Task Type	Description	AI Model	Dataset
Scene Classification	Determines the overall category or environment depicted in a remote sensing image. By analyzing color, texture, and spatial patterns, models can classify images as urban areas, forests, farmland, or other types of scenes. This approach is widely used for large-scale land cover mapping, environmental assessment, and resource monitoring.	ResNet	AID
Land Use Classification	Focuses on mapping the specific types of land use or land cover at the pixel or region level. Instead of labeling an entire image as one category, it identifies features like buildings, roads, vegetation, and water. This detailed breakdown is critical for urban planning, agricultural management, and ecological protection.	HRNet	LoveDA
Object Detection	Identifies and localizes specific objects—such as vehicles, ships, buildings, or airplanes—within aerial or satellite images. By determining both the category and position of each object, this task supports infrastructure monitoring, disaster damage assessment, and resource management.	YOLOv5	DOTA
Image Captioning	Generates concise textual descriptions of remote sensing images, capturing key elements and overall context. This helps automate the interpretation of complex scenes, making it easier for experts and non-experts alike to understand large volumes of imagery in applications like emergency response and environmental monitoring.	BLIP	BLIP Dataset
Edge Detection	Extracts the boundaries or outlines of objects and regions in an image. By detecting sharp intensity changes or structural differences, it highlights roads, coastlines, and building edges, and often serves as a foundation for more advanced image analysis tasks, such as segmentation or feature extraction.	Canny	——
Object Counting	Estimates the number of specific targets—like trees, cars, or livestock—within a given area. This provides valuable insights for resource assessment, agricultural yield estimation, population studies, and urban planning, helping stakeholders make data-driven decisions based on accurate counts of objects of interest.	——	——

Tab1. Remote Sensing ChatGPT (Guo et al., 2024)

Application Examples

Exploring LLM-Powered Remote Sensing Applications

Counting Airplanes on a Runway
The system handles a user request like "Count the number of airplanes on the runway" by sequentially performing runway segmentation, airplane detection, and object counting. Demonstrates how multiple remote sensing tasks can be orchestrated to solve a more complex query.

Locating a Baseball Diamond
For a query such as "Locate the baseball diamond in the aerial image," the system applies object detection to identify and pinpoint the diamond's position.
Shows the ability to tackle targeted object localization in remote sensing images.

Segmenting Cultivated Land
Highlights a failure case where the land use classification model does not include "cultivated land" as a category in its training set.Reveals the importance of comprehensive model training for real-world land use scenarios.

Handling Unsupported Queries
When tools or data do not fully address a user's request, the system may generate an imagined response instead of requesting more information.
Emphasizes the need for improved error handling and query clarification.

Experiment Results and Limitations

Exploring LLM-Powered Remote Sensing Applications

Experimental Setup

138 user queries tested
ChatGPT automatically planned and executed tasks
Evaluation: Accuracy of AI task assignment & execution

Key Challenges

Limited AI model coverage
(e.g., land types not in training data)
ChatGPT occasionally guesses instead of asking for clarification
Dependence on high-performance remote sensing foundation models

ChatGPT Version	Task Planning Accuracy	
GPT-4	63%	
GPT-4-1106	84.10%	
GPT-3.5-turbo	94.90%	√
GPT-3.5-turbo-1106	29%	×

"While LLMs like ChatGPT show promise in task planning, their hallucination tendencies highlight the necessity for explicit knowledge constraints and improved reasoning mechanisms."

Future Considerations

Exploring LLM-Powered Remote Sensing Applications

How can we effectively **integrate multi-source, multi-modal remote sensing data** (e.g., optical, SAR, LiDAR, UAV imagery) within an LLM-based agent to enhance interpretation accuracy and robustness?

When dealing with **high-resolution or large-scale imagery**, how can we balance computational resources and algorithmic efficiency while ensuring stable performance across different resolutions?

Remote sensing imagery often contains **time-series information** (e.g., change detection, temporal prediction).
How can we combine LLMs with spatiotemporal analysis models to automate the interpretation of dynamic changes in geospatial features?

Remote sensing data may **include uncertainties** such as cloud cover, shadows, or noise.
How can we incorporate uncertainty estimation or quality control into task planning and execution within an LLM-based agent?

Deep learning and large models are frequently viewed as “black boxes.”
How can we **improve explainability** in remote sensing interpretation, making the results more auditable and trustworthy?