

# TerraQuery User Guide

December 5, 2025

## 1 Introduction

This document explains the installation, setup, and usage of the TerraQuery website for satellite imagery analysis.

## 2 System Setup and Installation

This section outlines the deployment steps and system architecture. For detailed command-line instructions and configuration files, please refer to the `installation.md` file located in the project root.

### 2.1 Prerequisites

The system requires the following environment:

- **OS:** Ubuntu 20.04 LTS, 22.04 LTS, or 24.04 LTS.
- **Python:** Version 3.10 (Conda/Micromamba recommended), libraries are installed in the given scripts or following commands.
- **Hardware:** GPU node with NVIDIA drivers and CUDA support for SAM and VLM models.
- **Network:** We used institute VPN to connect the backend to the GPU Node.

### 2.2 Installation Steps

You can use `deployment_script.sh` to install the whole app. Alternatively the steps are also listed here:

#### 2.2.1 1. Models Setup

First, create the environment and install dependencies using the `install` script:

```
conda create -n geonli python=3.10 -y
conda activate geonli
chmod +x install.sh
./install.sh
```

Then, expose the APIs in separate terminals or using `nohup`/`systemd` services:

```
# Terminal 1 (SAM3)
cd sam3
uvicorn api:app --host 0.0.0.0 --port 6767

# Terminal 2 (VQA)
cd vqa
uvicorn unified_api:app --host 0.0.0.0 --port 8001
```

## 2.2.2 2. Frontend Setup

Move the frontend build and install Nginx:

```
cd frontend
npm i
npm run build
cd ..
sudo mv frontend /var/www/
sudo apt install nginx
```

Configure Nginx (create /etc/nginx/sites-available/interiit):

```
server {
    listen 80;
    server_name <YOUR-SERVER-IP>;
    client_max_body_size 50M;
    root /var/www/frontend/dist;
    index index.html;

    location / {
        try_files $uri $uri/ /index.html;
    }

    location /api/ {
        proxy_pass http://127.0.0.1:8000/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

Enable the site:

```
sudo rm /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/interiit /etc/nginx/sites-enabled/interiit
sudo systemctl enable --now nginx
```

## 2.2.3 3. Backend Setup

Install dependencies and setup the service:

```
cd backend
sudo apt install gunicorn uvicorn python3-fastapi python3-httptools \
    python3-dotenv python3-openai python3-pil
```

Create the systemd service file at /etc/systemd/system/interiit-backend.service:

```
[Unit]
Description=FastAPI Backend
After=network.target

[Service]
User=ubuntu
Group=www-data
WorkingDirectory=/home/ubuntu/backend
ExecStart=/usr/bin/gunicorn -w 2 -k uvicorn.workers.UvicornWorker \
    api:app --bind 127.0.0.1:8000
```

```
[Install]
WantedBy=multi-user.target
```

Finally, start the backend service:

```
sudo systemctl daemon-reload
sudo systemctl enable --now interiit-backend
```

In case this doesn't work, you can run this command inside the backend folder:

```
gunicorn -w 2 -k uvicorn.workers.UvicornWorker api:app --bind 127.0.0.1:8000
```

## 2.3 API Reference

The system exposes several endpoints across its services:

### VQA Service (Port 8001)

- POST `/router`: Classify prompts.
- POST `/vqa/{mode}`: Run VQA tasks (attribute, numerical, binary, filtering).
- POST `/bbox`: Generate bounding boxes.
- POST `/caption`: Generate image captions.

### SAM3 Service (Port 6767)

- POST `/masks`: Generate segmentation masks.
- POST `/boxes`: Generate bounding boxes.
- POST `/merged_masks`: Generate the merged masks using vanilla and qwen conditioned SAM3

**Backend Proxy** The backend proxy forwards requests to the VQA service and is accessible via the Nginx reverse proxy at `/api`.

### 3 Website User Guide

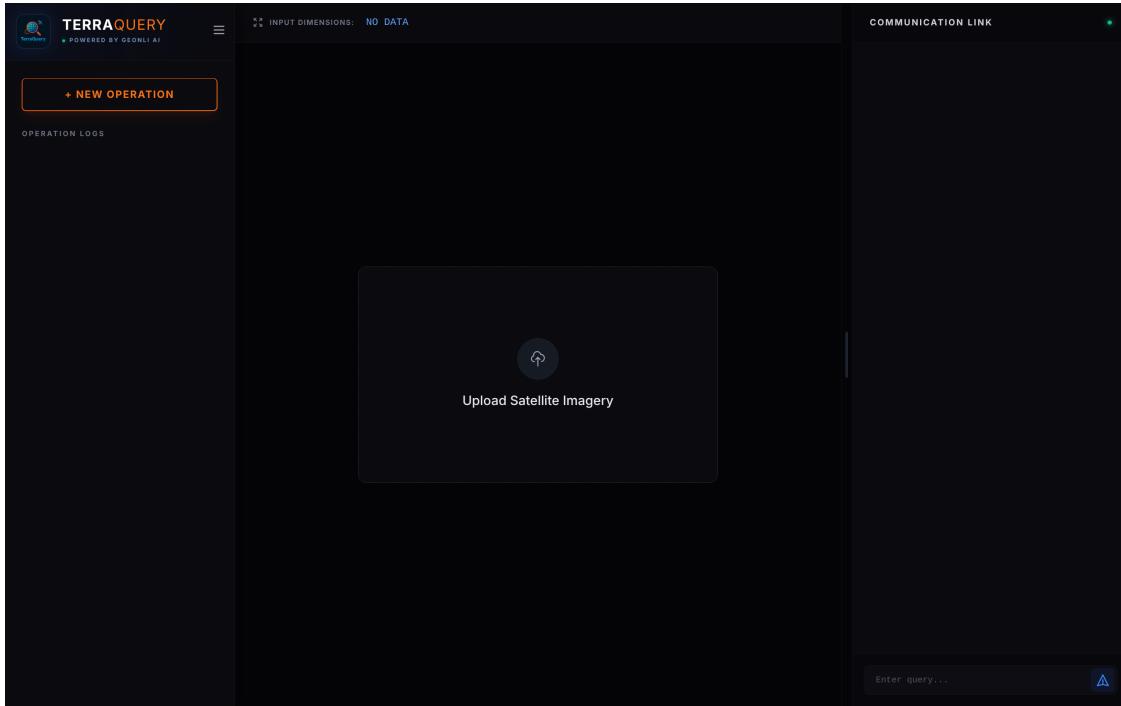


Figure 1: Initial Screen

To begin, click on the **Upload Satellite Imagery** button in the center of the screen and choose your input file from your device.

## 4 Image Analysis and Interface

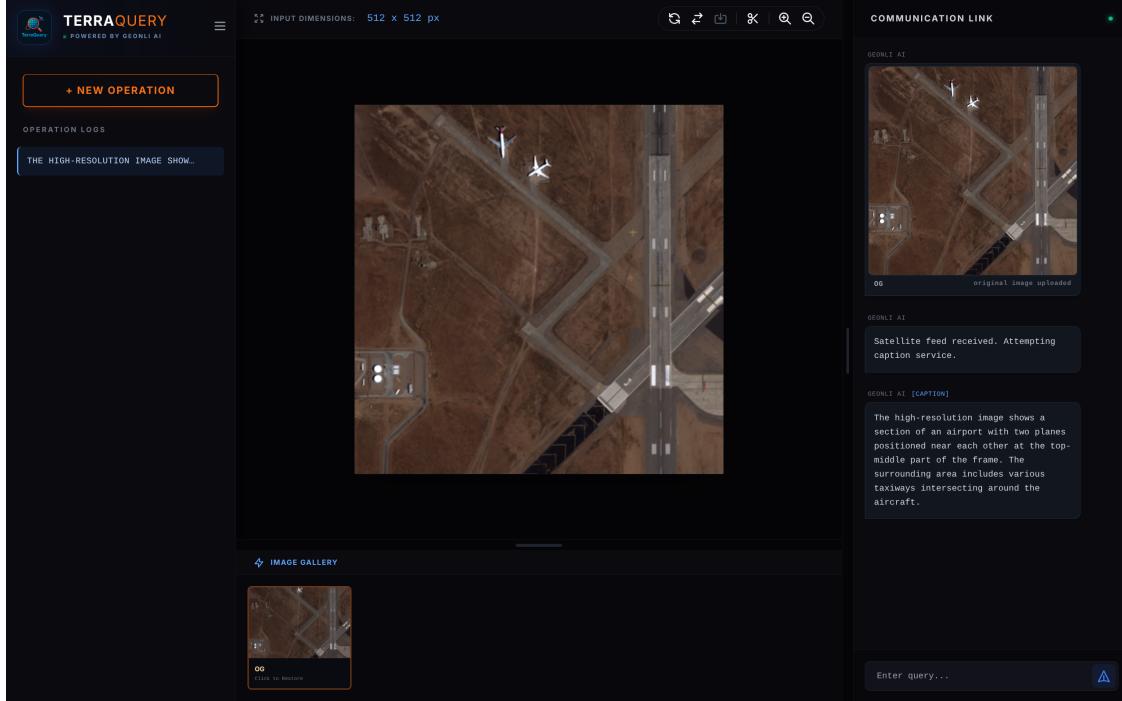


Figure 2: Image Loaded with Caption

Once the image is uploaded, it gets a caption automatically, which is displayed in the chat window on the right. You can ask further questions about the image using the prompt box located in the bottom right corner.

The interface is customizable:

- You can resize the bottom and right panels using the drag handles.
- The left sidebar can be collapsed using the button on the top right of the sidebar.

## 5 Object Detection Example

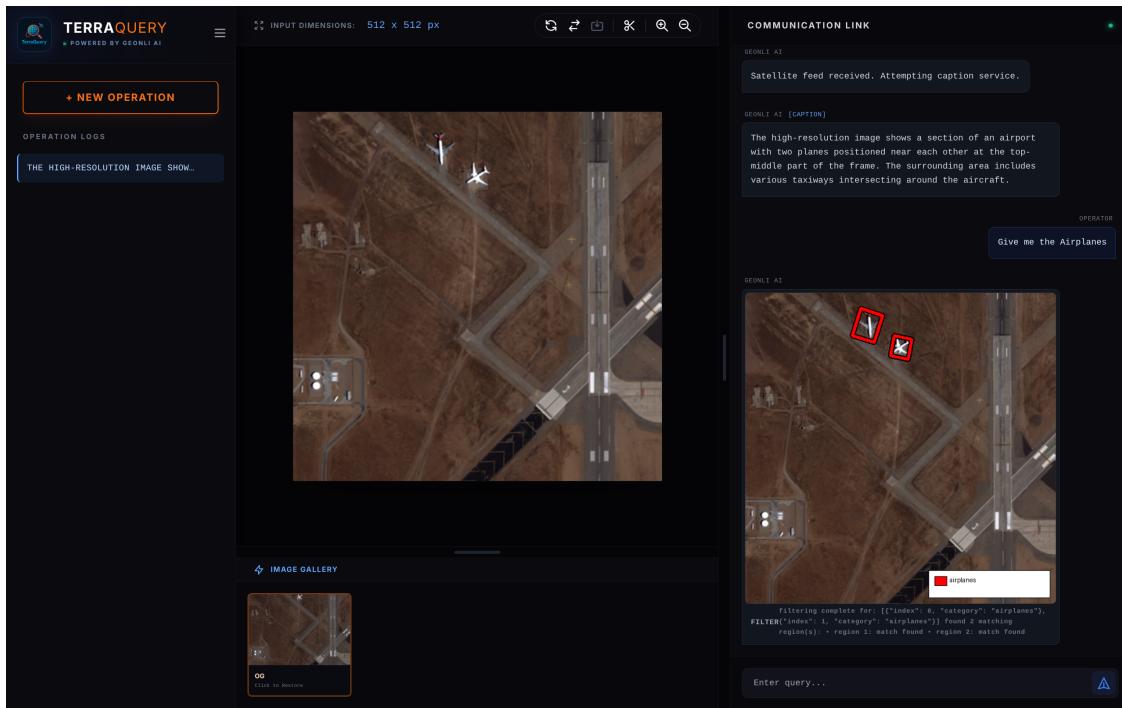


Figure 3: Querying for Objects

Here is an example of a prompt asking to "Give me the Airplanes". The system processes the request and displays the output in the chat window, highlighting the detected objects on the image.

## 6 Starting a New Session

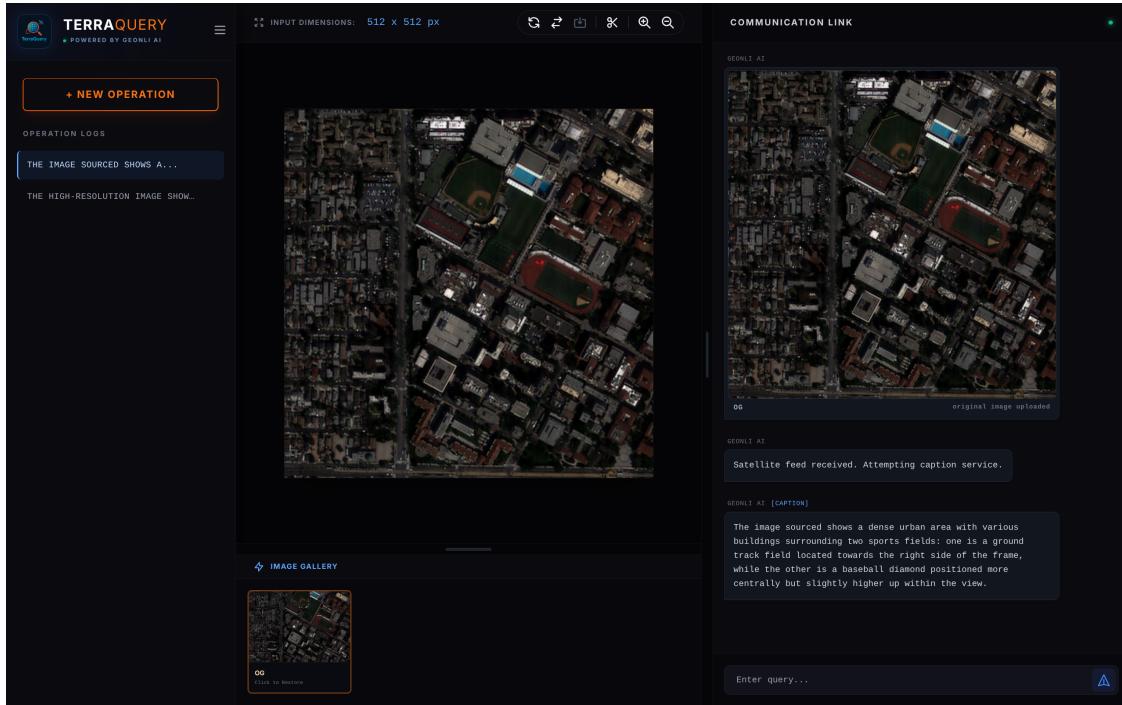


Figure 4: New Operation

A new session can be started by clicking on the **+ NEW OPERATION** button on the top left. After selecting a new image, the automatic captioning process will repeat for the new image.

## 7 Image Manipulation

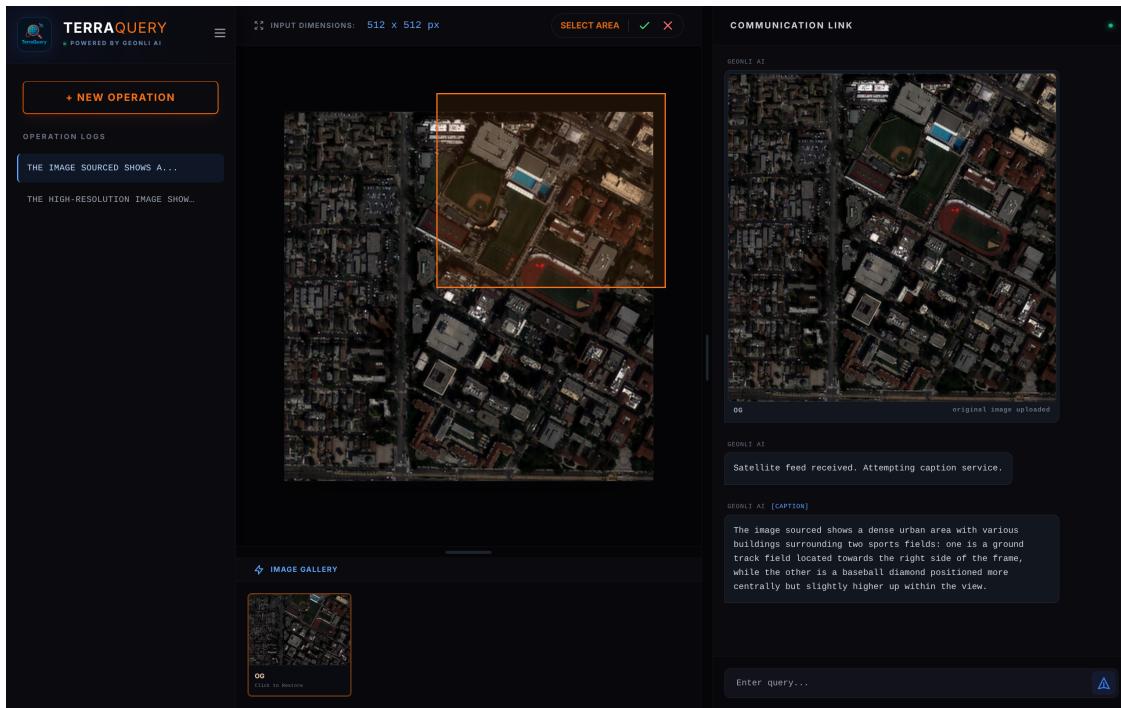


Figure 5: Image Tools and Gallery

The toolbar on the top provides several image manipulation features:

- The image can be rotated or flipped. If you do this, you need to save the changes using the third save button in the toolbar.
- The image can also be cropped and saved.
- Previous original images are preserved and available in the bottom panel (Image Gallery). You can switch between them by clicking on the thumbnails.

## 8 Saving Edits

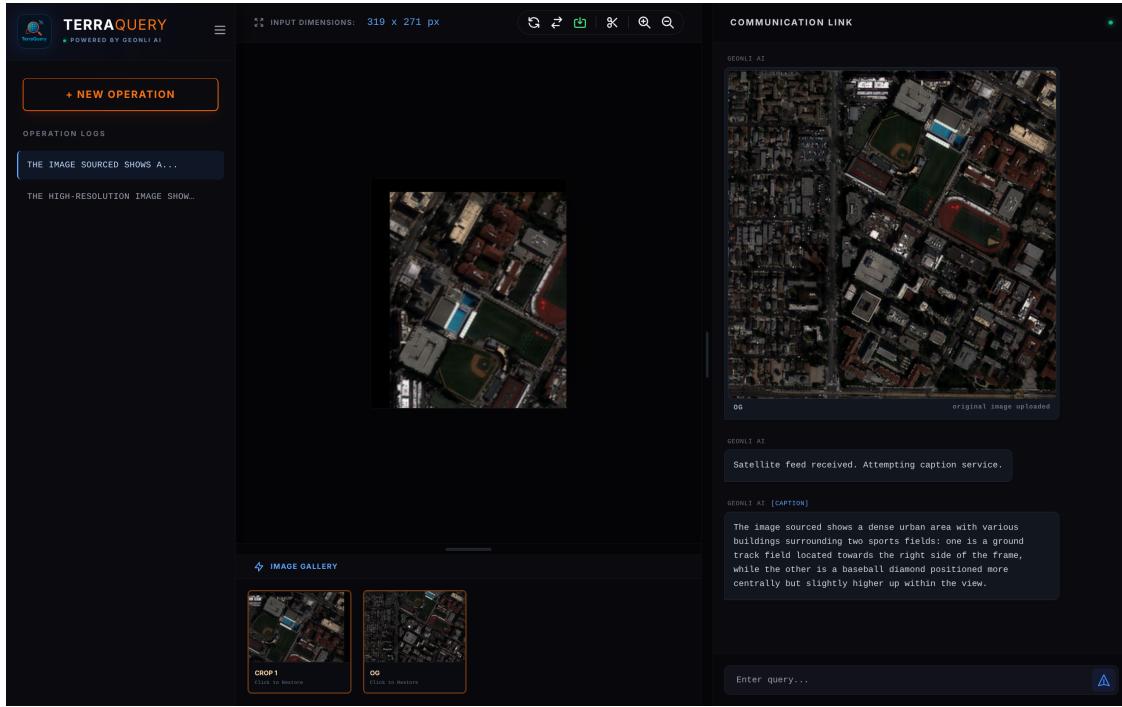


Figure 6: Saving Rotated/Cropped Image

This screenshot shows a cropped and rotated image. To save this specific rotation and crop, click the third save button in the toolbar, which will appear green to indicate the save action.

## 9 Querying Edited Images

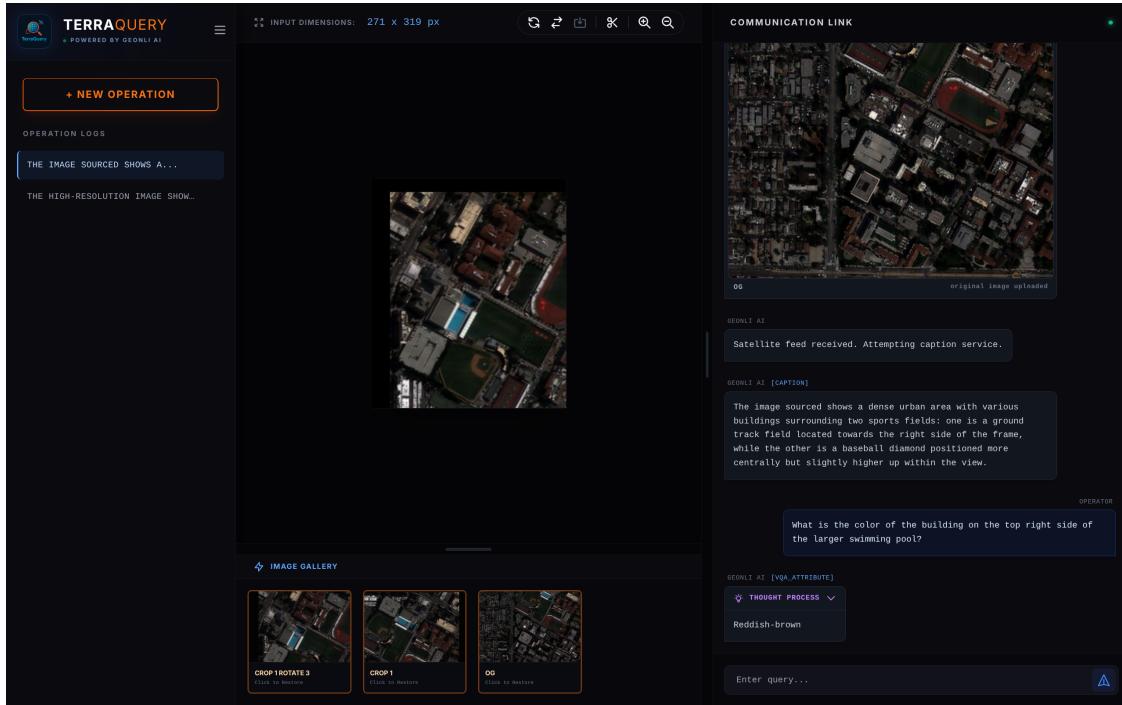


Figure 7: Querying the Edited Image

You can continue to ask questions on the cropped and rotated image, as shown in this example where a specific question about the building color is asked.

## 10 Evaluation

You can run the evaluation script at the backend endpoint /api/eval.