An Introduction to the GeoAI Python Package: Bridging AI and Geospatial Analysis

Qiusheng Wu
Associate Professor
Department of Geography & Sustainability
University of Tennessee, Knoxville
https://gishub.org

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Slides: https://tinyurl.com/geoai-talk

slides

Outline

- Open Geospatial Solutions
- Introduction to GeoAI
- Key Features
  - Data Download
  - Data Visualization
  - Data Preparation
  - Model Training
  - Model Inference
  - Post-processing
- Applications
  - buildings, cars, ships, solar panels, wetlands, waters, land cover classification
- Additional Resources
- Q&A

Open Geospatial Solutions

https://github.com/opengeos

Open Geospatial Solutions

A collection of open-source software packages for the geospatial community

2.6k followers | United States of America | https://opengeos.org | @giswqs | in/giswqs | @giswqs

https://www.facebook.com/groups/... | opengeos@outlook.com

Python Packages

- geemap
- geoai
- geospatial
- geospatial-ml
- HyperCoast
- leafmap
- lidar
- mapwidget
- open-buildings
- pygis
- segment-geospatial
- whitebox-python
- whiteboxgui

Data Catalogs

- geospatial-data-catalogs
- aws-open-data
- aws-open-data-geo
- aws-open-data-stac
- Earth-Engine-Catalog
- NASA-CMR-STAC
- NASA-Earth-Data
- stac-index-catalogs
- maxar-open-data
- datasets
- data
- ee-tile-layers

ArcGIS Toolboxes

- WhiteboxTools-ArcGIS

Web Apps

- streamlit-geospatial
- streamlit-map-template
- solara-geemap
- solara-geospatial
- solara-template
- solara-maxar
- voila-geospatial
- geospatial-dataviz
- surface-water-app

R Packages

- whiteboxR

Useful Resources

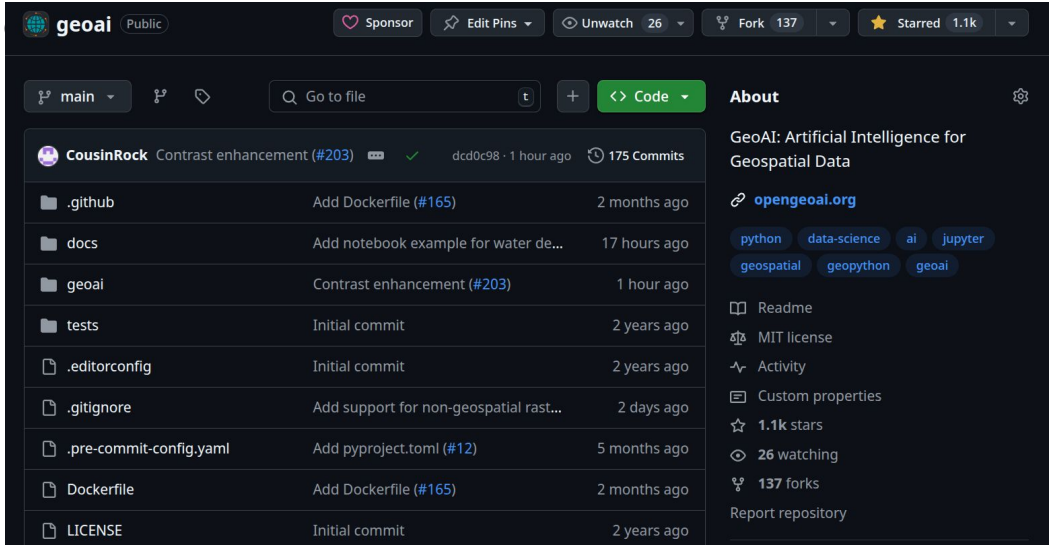
- Awesome-GEE
- python-geospatial

https://github.com/opengeos

Introduction to GeoAI

Introduction to the GeoAI Python Package

- A powerful Python package for integrating Artificial Intelligence with geospatial data analysis and visualization
- GitHub: https://github.com/opengeos/geoai



Installation

https://opengeoai.org/installation

- pip
  - pip install geoai-py
- conda/mamba
  - mamba install -c conda-forge geoai
- uv
  - uv pip install geoai-py

- GPU support
  - mamba install -c conda-forge geoai "pytorch=*=cuda*"

Key Features

Data Download

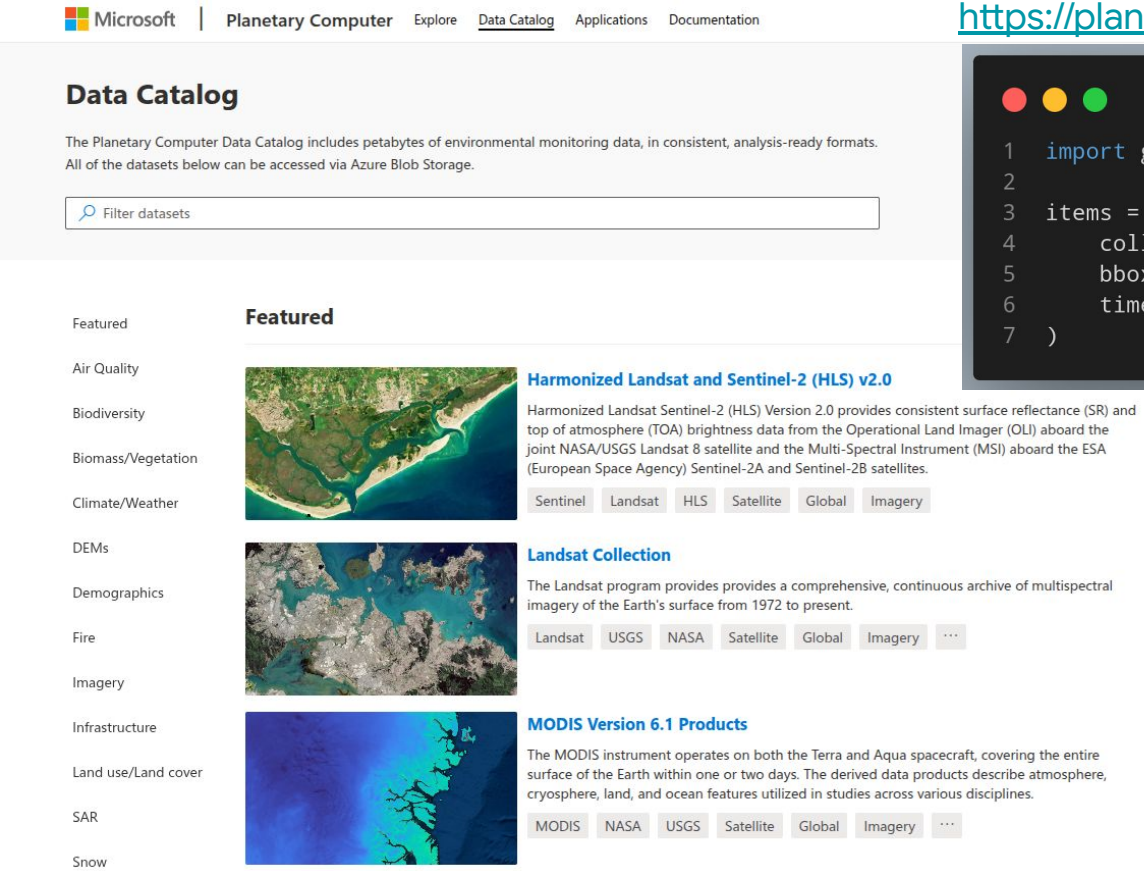
- Search and download data from SpatioTemporal Asset Catalogs, including Microsoft Planetary Computer
- Search and download data from Overture Maps, including buildings, roads, and places

```python
import geoai

bbox = (-117.6029, 47.65, -117.5936, 47.6563)
geoai.download_naip(
    bbox=bbox,
    output_dir="naip_data",
)
geoai.download_overture_buildings(
    bbox=bbox,
    output="buildings.geojson",
)
```

notebook

Microsoft Planetary Computer

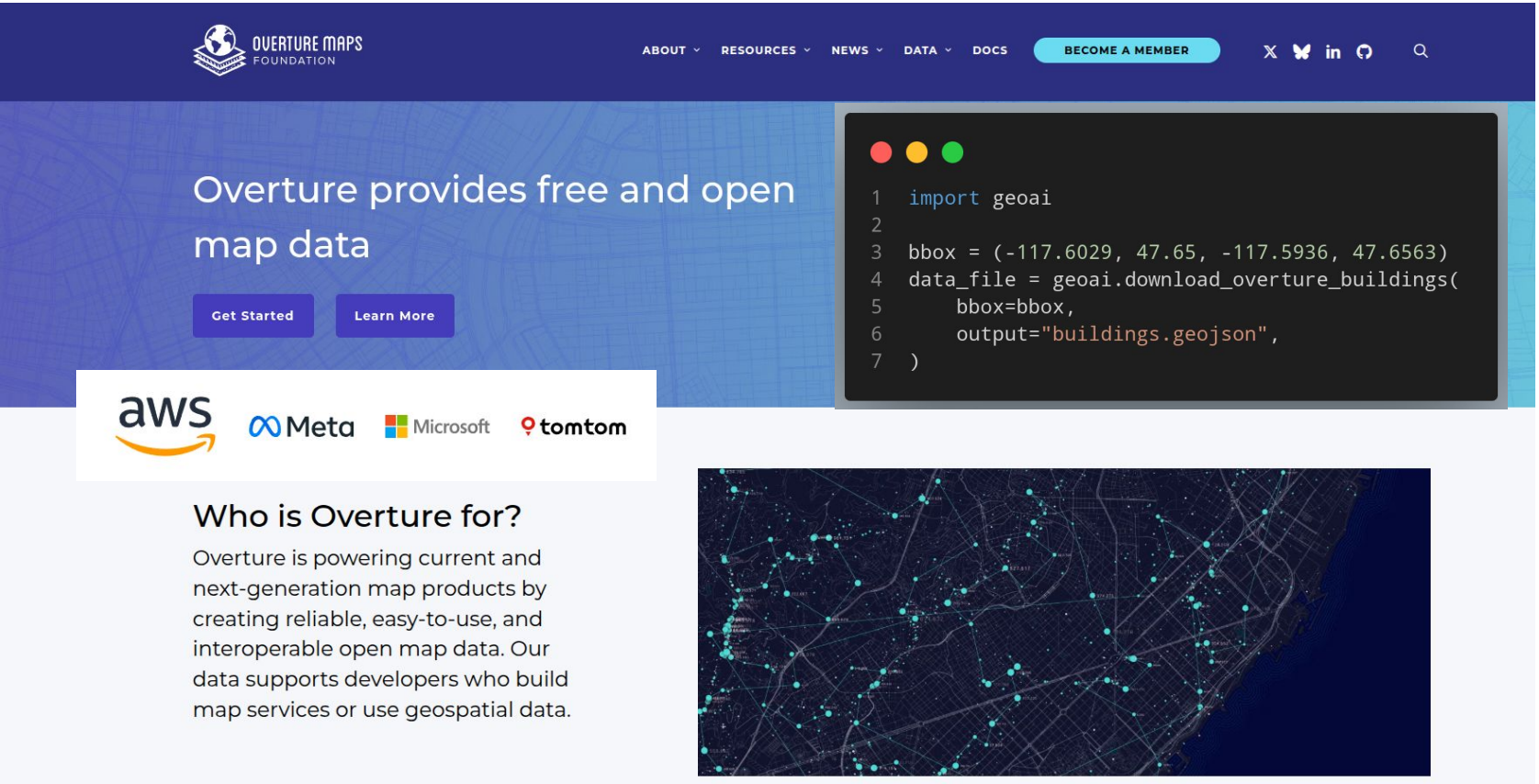
https://planetarycomputer.microsoft.com/catalog




```python
import geoai

items = geoai.pc_stac_search(
    collection="naip",
    bbox=[-76.6657, 39.2648, -76.6478, 39.2724],
    time_range="2013-01-01/2014-12-31",
)
```

Overture Maps

https://overturemaps.org

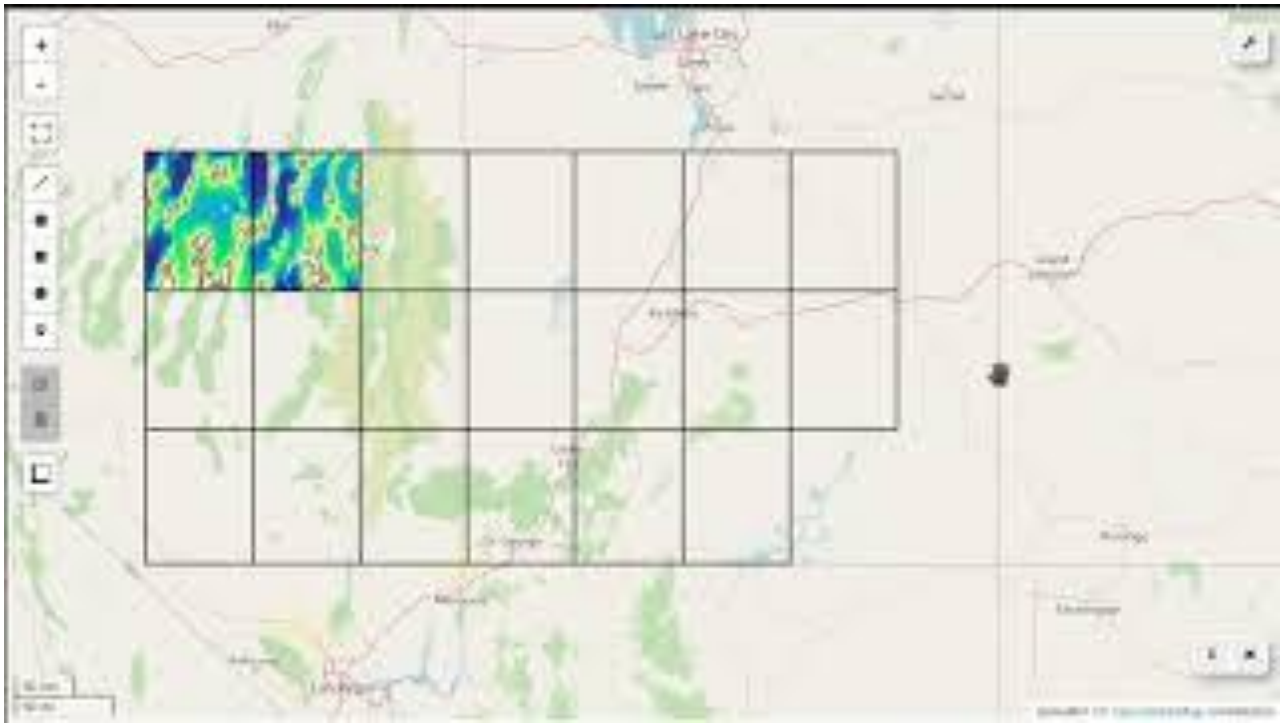


Data Visualization - Planetary Computer

notebook

- Search and visualize data from Planetary Computer

```
1  import geoai
2
3  m = geoai.Map()
4  m
```



Data Visualization - 3D Globe

notebook

- Visualize geospatial data on a 3D globe

```python
import geoai

m = geoai.MapLibre(
    center=[-100, 40],
    zoom=3,
    style="liberty"
    )
m.add_overture_3d_buildings()
m.add_globe_control()
m

```

Data Preparation - Create Vector Data

notebook

- Create vector data interactively

```python
import geoai

m = geoai.MapLibre(style="liberty")
m.add_draw_control()
properties = {
    "Type": ["Residential", "Commercial"],
    "Area": 3000,
    "Name": "Building",
    "City": "Seattle",
}
widget = geoai.create_vector_data(m, properties)
widget

```

Data Preparation - Edit Vector Data

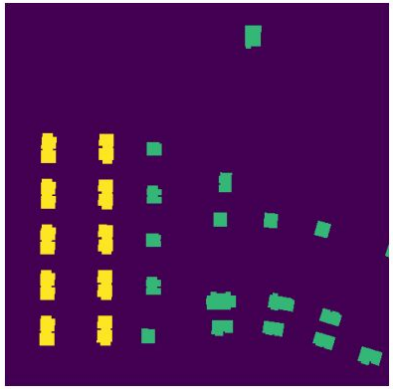
- Edit vector data interactively

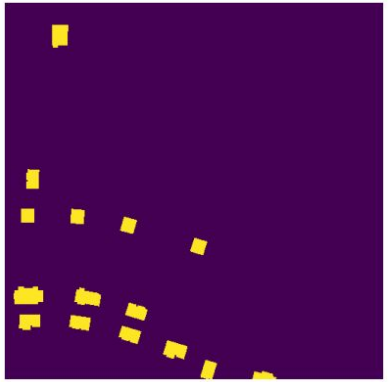
notebook

```python
import geoai

m = geoai.MapLibre(style="liberty")
url = "/path/to/your/geojson/file.geojson"
props = {
    "class": ["apartments", "terrace", "house"],
    "height": 0.0,
}
widget = geoai.edit_vector_data(m, url, properties=props)
m.add_layer_control()
widget
```

Data Preparation - Create Image Chips

notebook

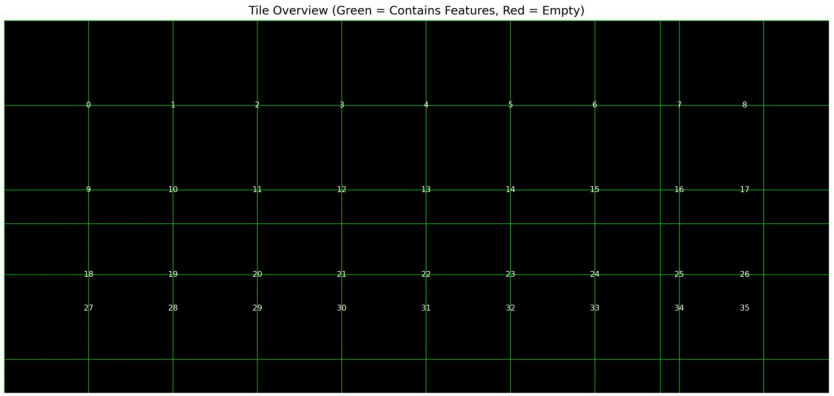


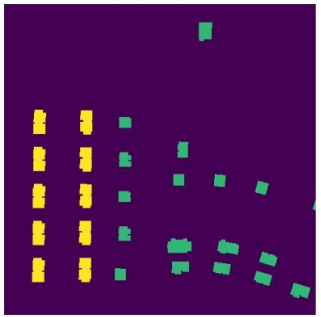


Data Preparation - Create Image Chips

notebook

```python
import geoai

raster_path = "/path/to/image.tif"
vector_path = "/path/to/vector.geojson"
tiles = geoai.export_geotiff_tiles(
    in_raster=raster_path,
    out_folder="output",
    in_class_data=vector_path,
    tile_size=512,
    stride=256,
    buffer_radius=0,
    create_overview=True,
)
```





Model Training

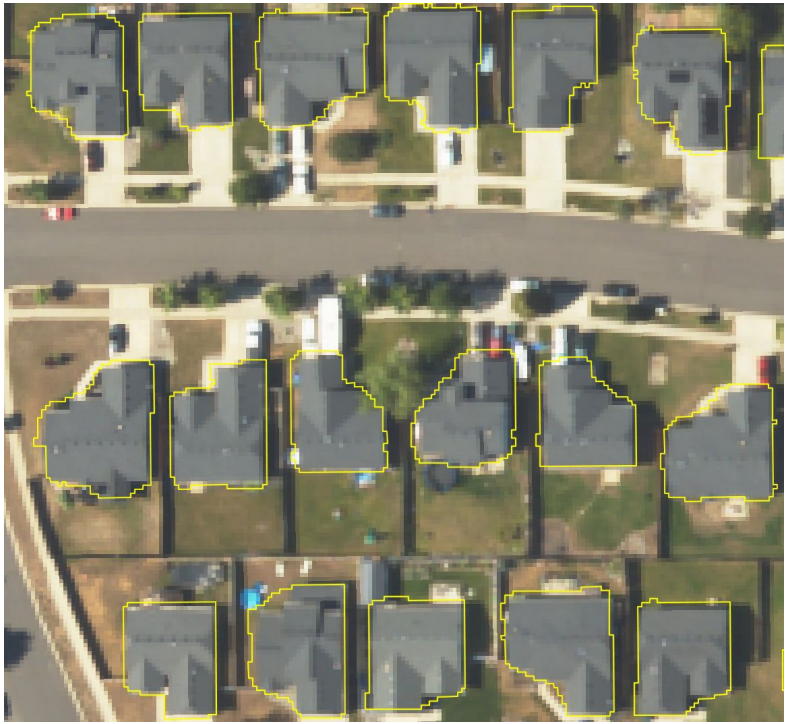
```python
import geoai

raster_path = "/path/to/image.tif"
vector_path = "/path/to/vector.geojson"

out_folder = "output"
tiles = geoai.export_geotiff_tiles(
    in_raster=raster_path,
    out_folder=out_folder,
    in_class_data=vector_path,
    tile_size=512,
    stride=256,
    buffer_radius=0,
)
geoai.train_MaskRCNN_model(
    images_dir=f"{out_folder}/images",
    labels_dir=f"{out_folder}/labels",
    output_dir=f"{out_folder}/models",
    num_channels=4,
    pretrained=True,
    batch_size=4,
    num_epochs=10,
    learning_rate=0.005,
    val_split=0.2,
)
```

notebook

Model Inference

notebook

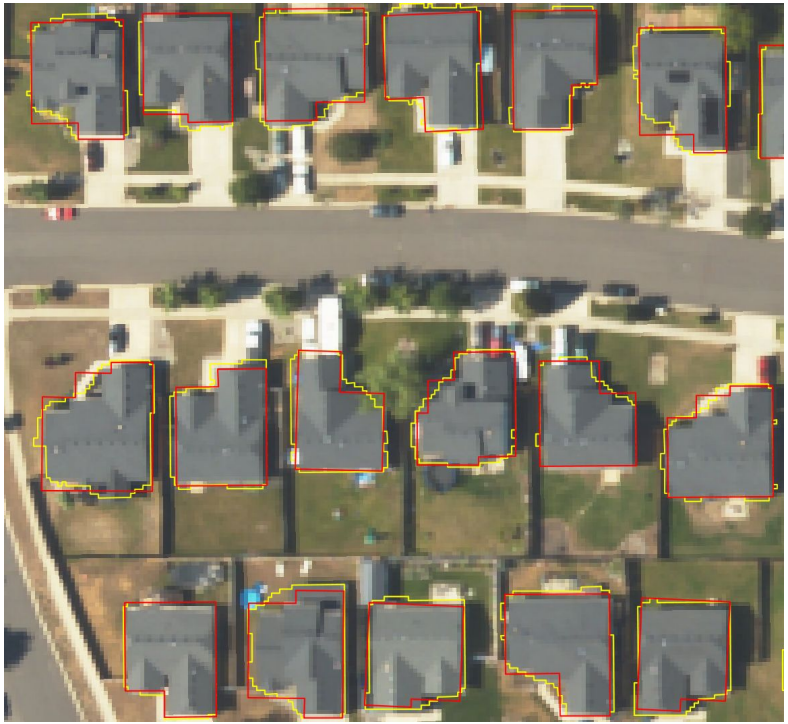
```python
import geoai

test_raster_path = "/path/to/test_image.tif"
out_folder = "output"
masks_path = "naip_test_prediction.tif"
model_path = f"{out_folder}/models/best_model.pth"
geoai.object_detection(
    test_raster_path,
    masks_path,
    model_path,
    window_size=512,
    overlap=256,
    confidence_threshold=0.5,
    batch_size=4,
    num_channels=4,
)
```



Post-processing - Regularization

notebook

```python
import geoai

vector_path = "/path/to/vector/file.geojson"
gdf = geoai.regularize(
    data=vector_path,
    simplify_tolerance=2.0,
    allow_45_degree=True,
    diagonal_threshold_reduction=30,
    allow_circles=True,
    circle_threshold=0.9,
)
```



Post-processing - Regularization

notebook

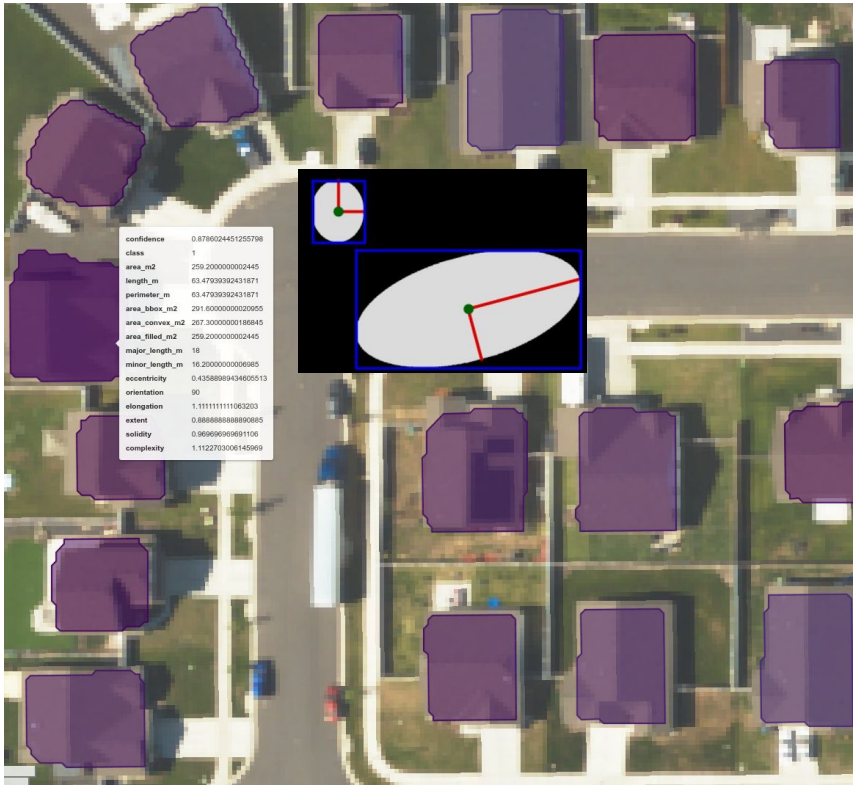
- Regularize building footprints

Post-processing - Geometric Properties

notebook

- Calculate geometric properties

```python
import geoai

vector_path = "/path/to/vector/file.geojson"
gdf_props = geoai.add_geometric_properties(
    data=vector_path,
    area_unit="m2",
    length_unit="m"
    )
gdf_props.head()
```



Applications

Building Footprint Extraction

notebook

```python
import geoai

raster_path = "/path/to/your/image.tif"
masks_path = "prediction.tif"
model_path = f"building_footprints_usa.pth"
geoai.object_detection(
    raster_path,
    masks_path,
    model_path,
    window_size=512,
    overlap=256,
    confidence_threshold=0.5,
    batch_size=4,
    num_channels=3,
)
```

Solar Panel Extraction

notebook

```python
import geoai

raster_path = "/path/to/your/image.tif"
masks_path = "prediction.tif"
model_path = "solar_panel_detection.pth"
geoai.object_detection(
    raster_path,
    masks_path,
    model_path,
    window_size=512,
    overlap=256,
    confidence_threshold=0.5,
    batch_size=4,
    num_channels=3,
)
```

Car Detection

notebook

```python
import geoai

raster_path = "/path/to/your/image.tif"
masks_path = "prediction.tif"
model_path = "car_detection_usa.pth"
geoai.object_detection(
    raster_path,
    masks_path,
    model_path,
    window_size=512,
    overlap=256,
    confidence_threshold=0.5,
    batch_size=4,
    num_channels=3,
)

```

Ship Detection

notebook

```python
import geoai

raster_path = "/path/to/your/image.tif"
masks_path = "prediction.tif"
model_path = "ship_detection.pth"
geoai.object_detection(
    raster_path,
    masks_path,
    model_path,
    window_size=512,
    overlap=256,
    confidence_threshold=0.5,
    batch_size=4,
    num_channels=3,
)
```

Surface Water Mapping

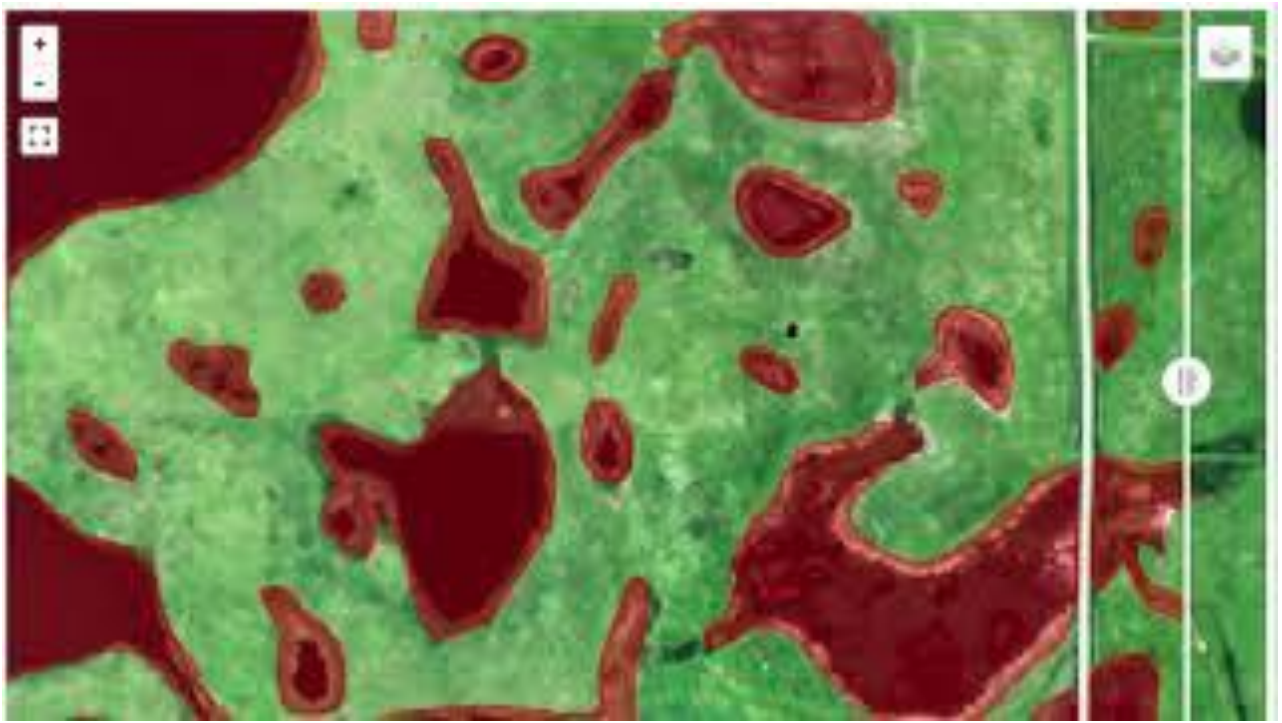
notebook

```python
import geoai

raster_path = "/path/to/your/image.tif"
masks_path = "prediction.tif"
model_path = "water_detection.pth"
geoai.object_detection(
    raster_path,
    masks_path,
    model_path,
    window_size=512,
    overlap=256,
    confidence_threshold=0.5,
    batch_size=4,
    num_channels=3,
)
```

Wetland Mapping

notebook

```python
import geoai

raster_path = "/path/to/your/image.tif"
masks_path = "prediction.tif"
model_path = "wetland_detection.pth"
geoai.object_detection(
    raster_path,
    masks_path,
    model_path,
    window_size=512,
    overlap=256,
    confidence_threshold=0.5,
    batch_size=4,
    num_channels=3,
)

```



Land Cover Classification

notebook

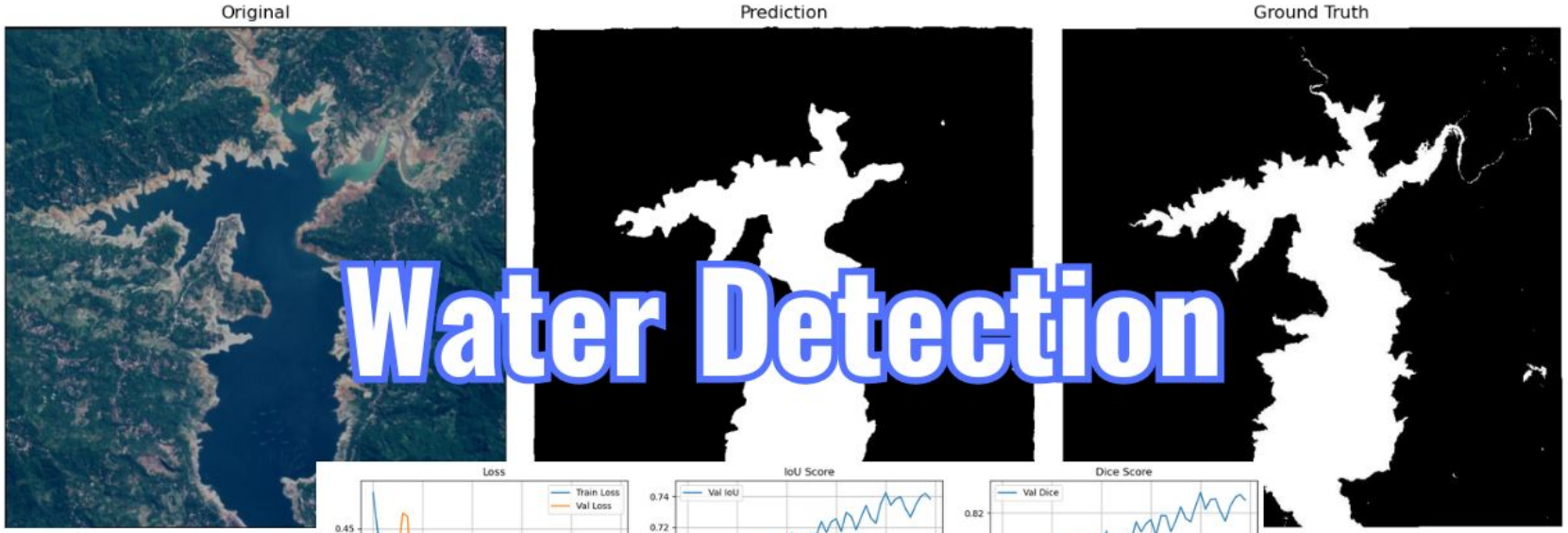
```python
import geoai

geoai.semantic_segmentation(
    input_path=test_raster_path,
    output_path=masks_path,
    model_path=model_path,
    architecture="unet",
    encoder_name="resnet34",
    num_channels=4,
    num_classes=13,
    window_size=512,
    overlap=256,
    batch_size=4,
)
```

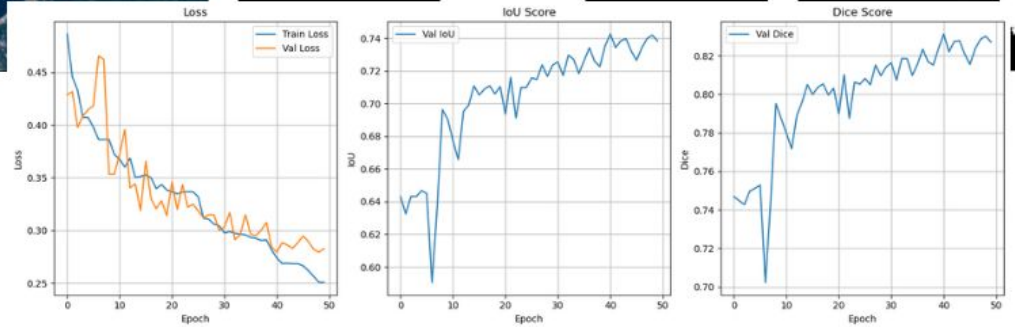
Create Image Chips from Multiple Images and Masks

notebook

Using Non-geospatial Raster Formats (JPG/PNG)

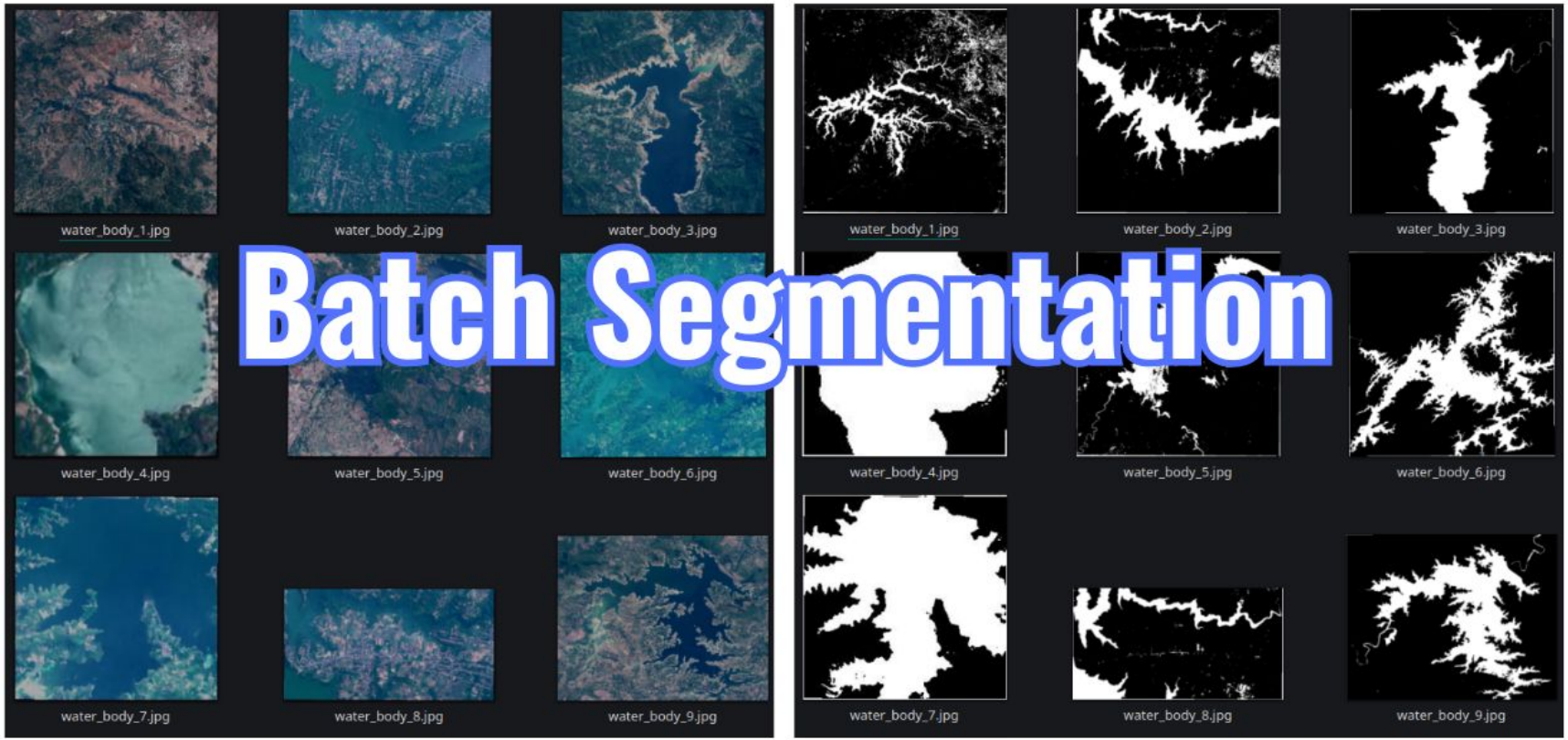
notebook





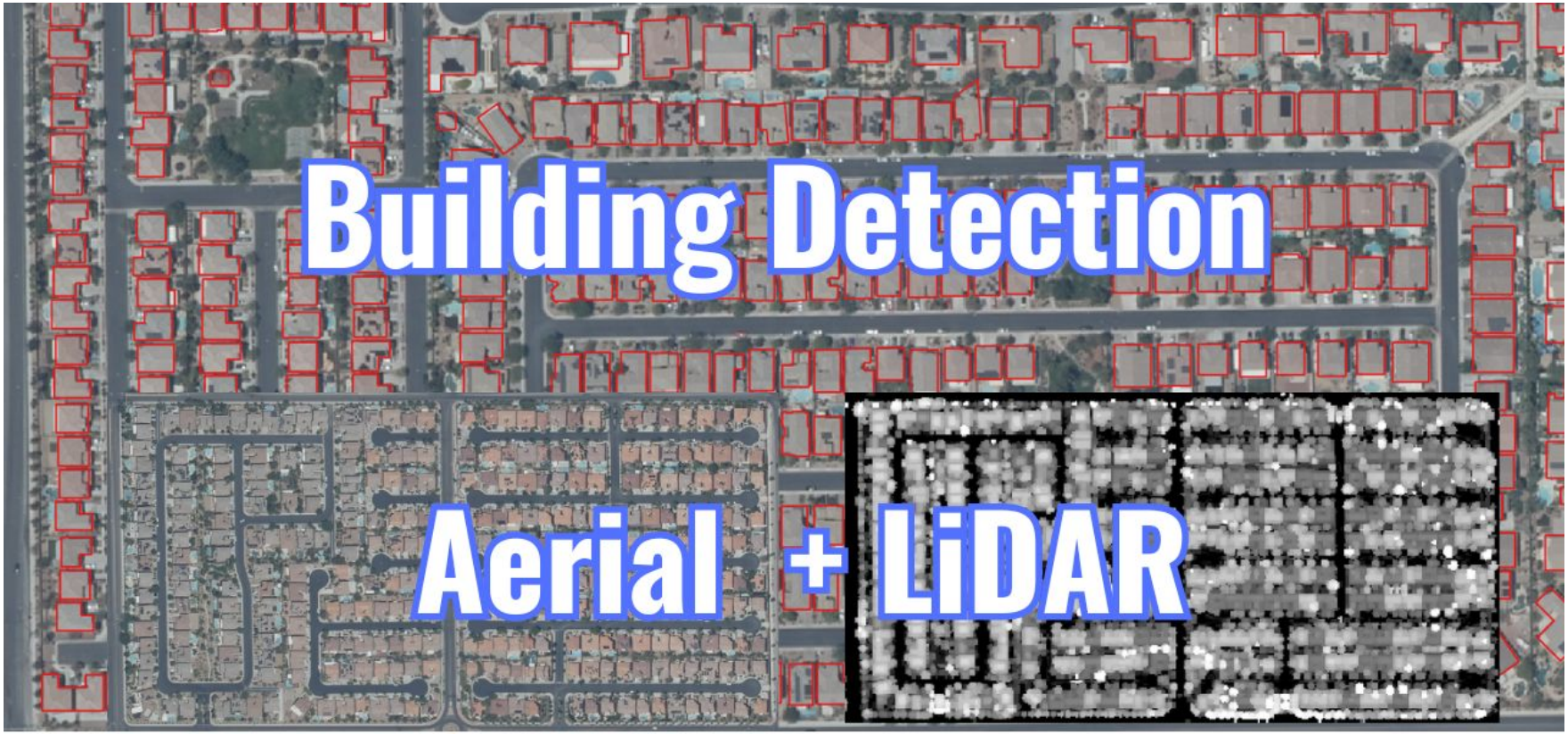
Batch Segmentation on Multiple Satellite Images

notebook



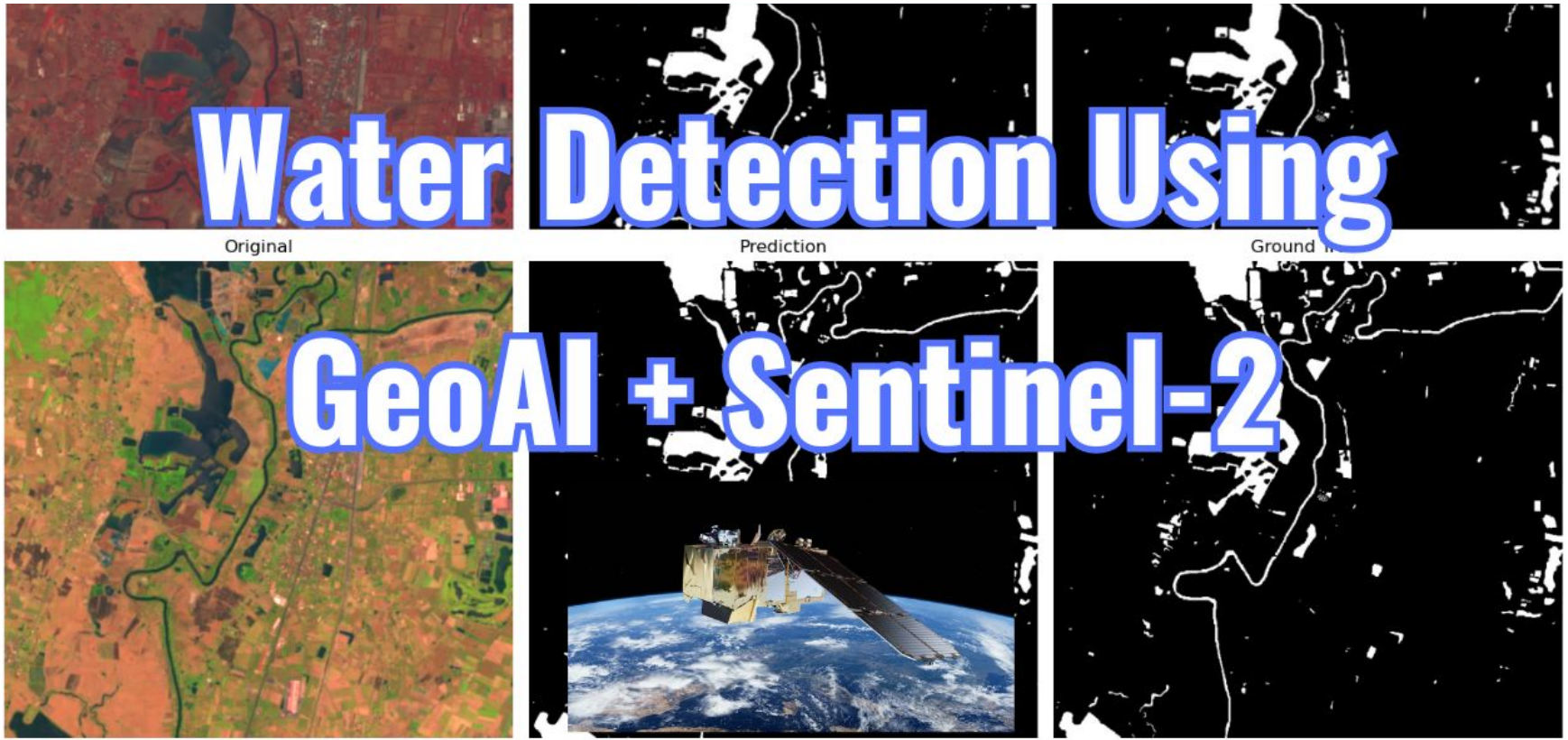
Building Detection with Aerial Imagery and LiDAR

notebook



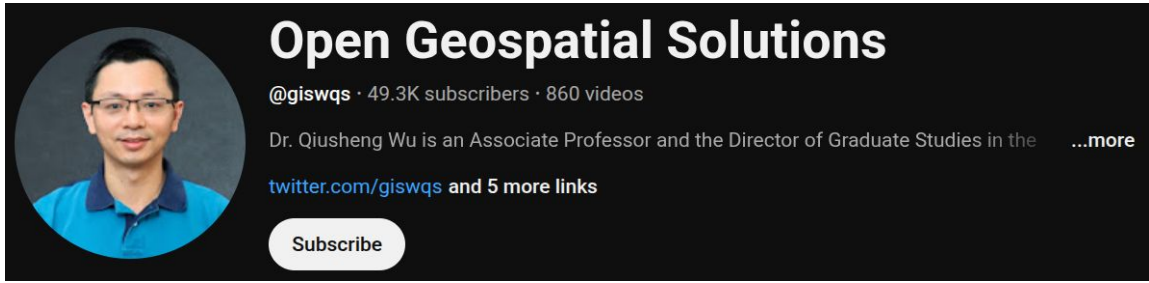
Water Detection Using Sentinel-2 Satellite Imagery

notebook



Additional Resources

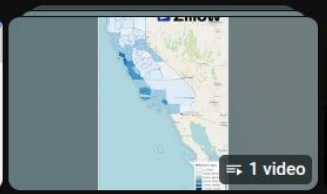
YouTube



https://youtube.com/@giswqs

Open Science Workshop
View full playlist

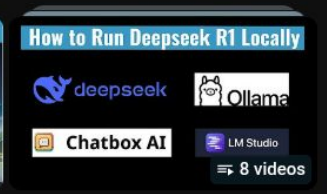
GeoAI Tutorials
Updated 3 days ago
View full playlist



Geospatial YouTube Shorts
View full playlist

Live Streaming
View full playlist

Manjaro Linux Tutorials
View full playlist



Deepseek and Open WebUI Tutorials
View full playlist

Geographic Software Design
Updated 2 days ago
View full playlist

PDFM Tutorials
View full playlist

Drone Imagery Data Processing and Visualization
View full playlist

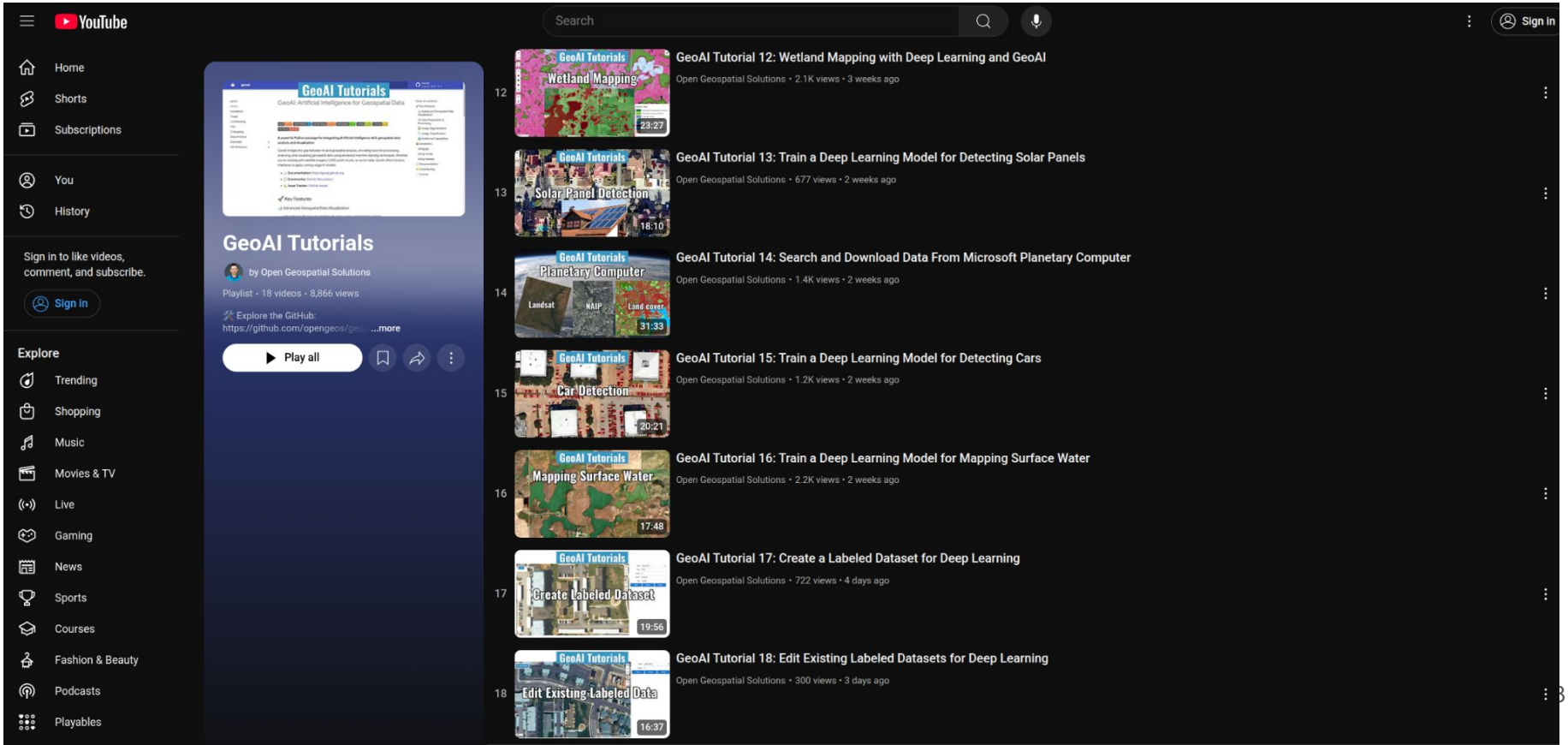
QGIS Tutorials
View full playlist

Introduction to GIS Programming
View full playlist

3D Mapping with MapLibre and Leafmap
Updated today
View full playlist

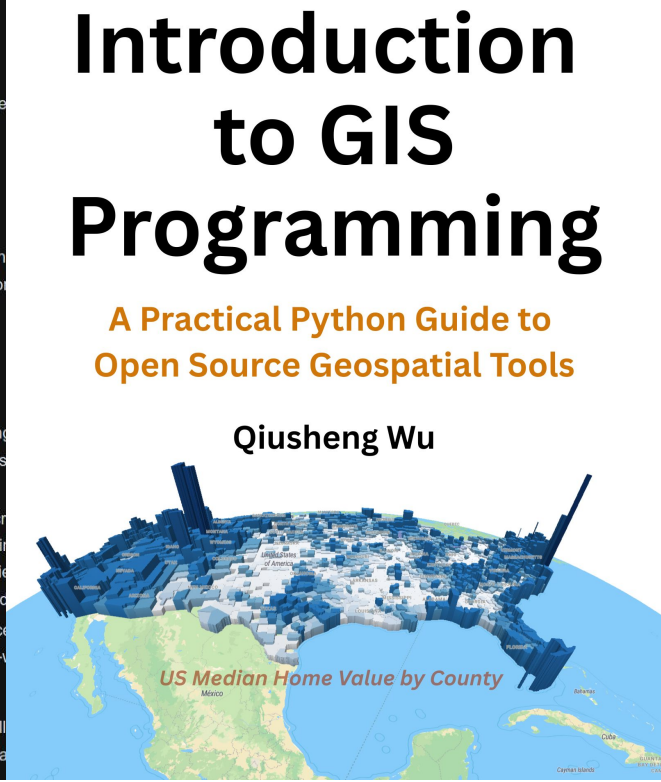
GeoAI Tutorials

https://tinyurl.com/GeoAI-Tutorials



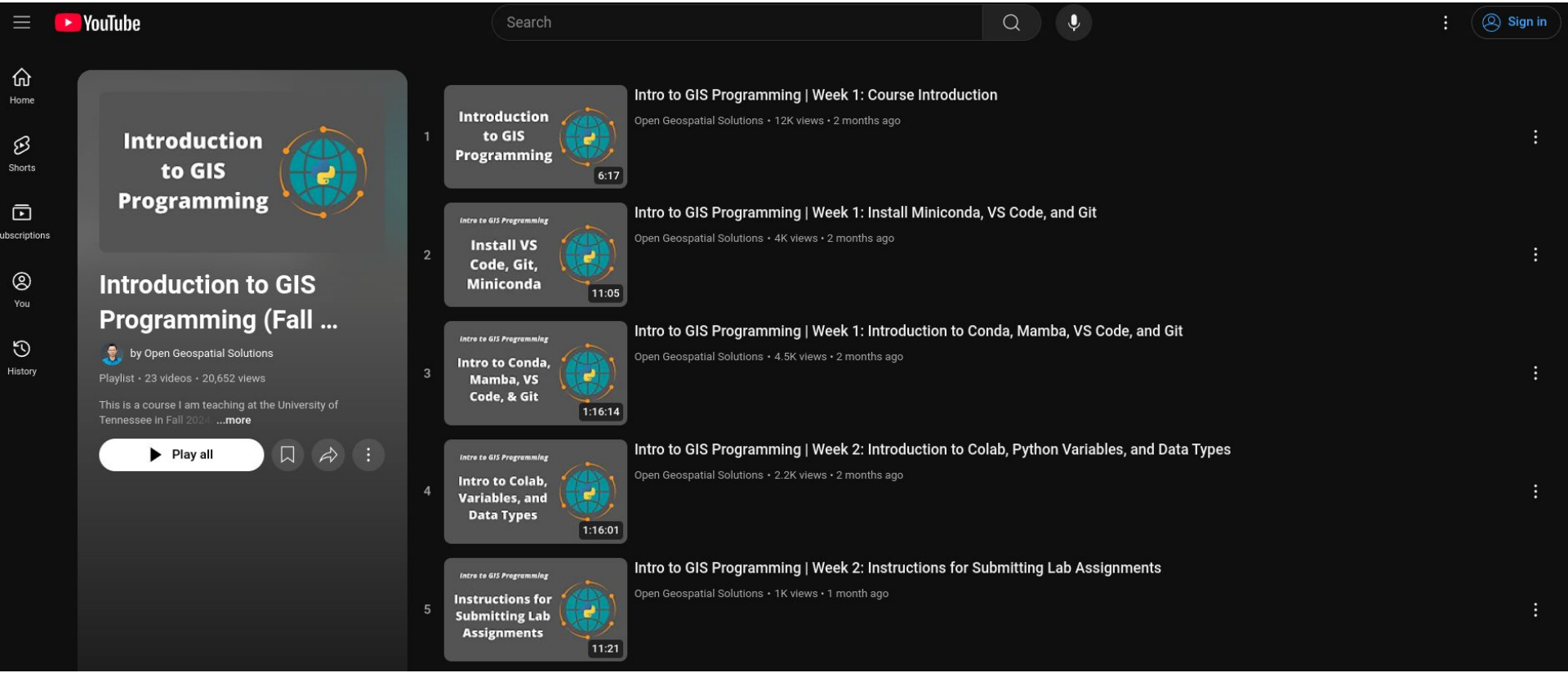
Introduction to GIS Programming

https://gispro.gishub.org/



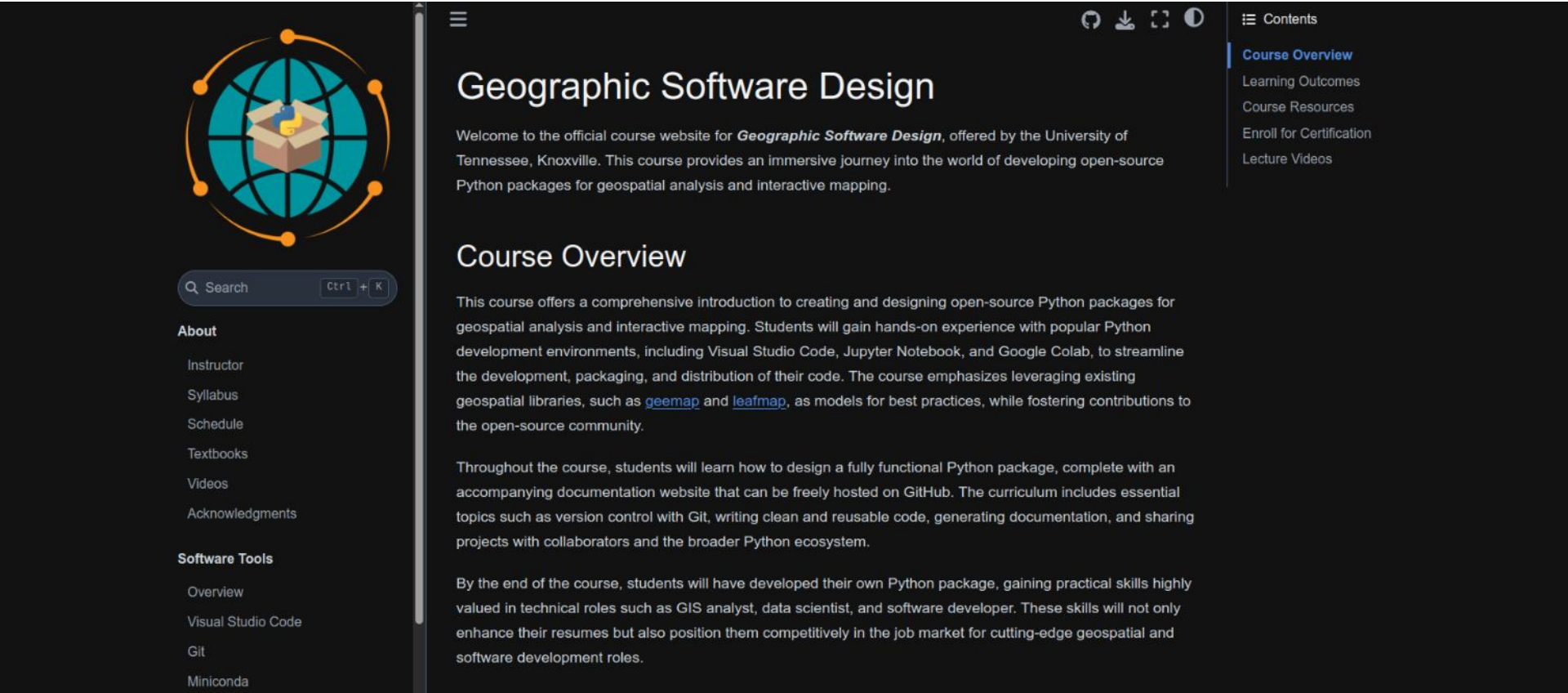
Introduction to GIS Programming

https://bit.ly/intro-gis-programming



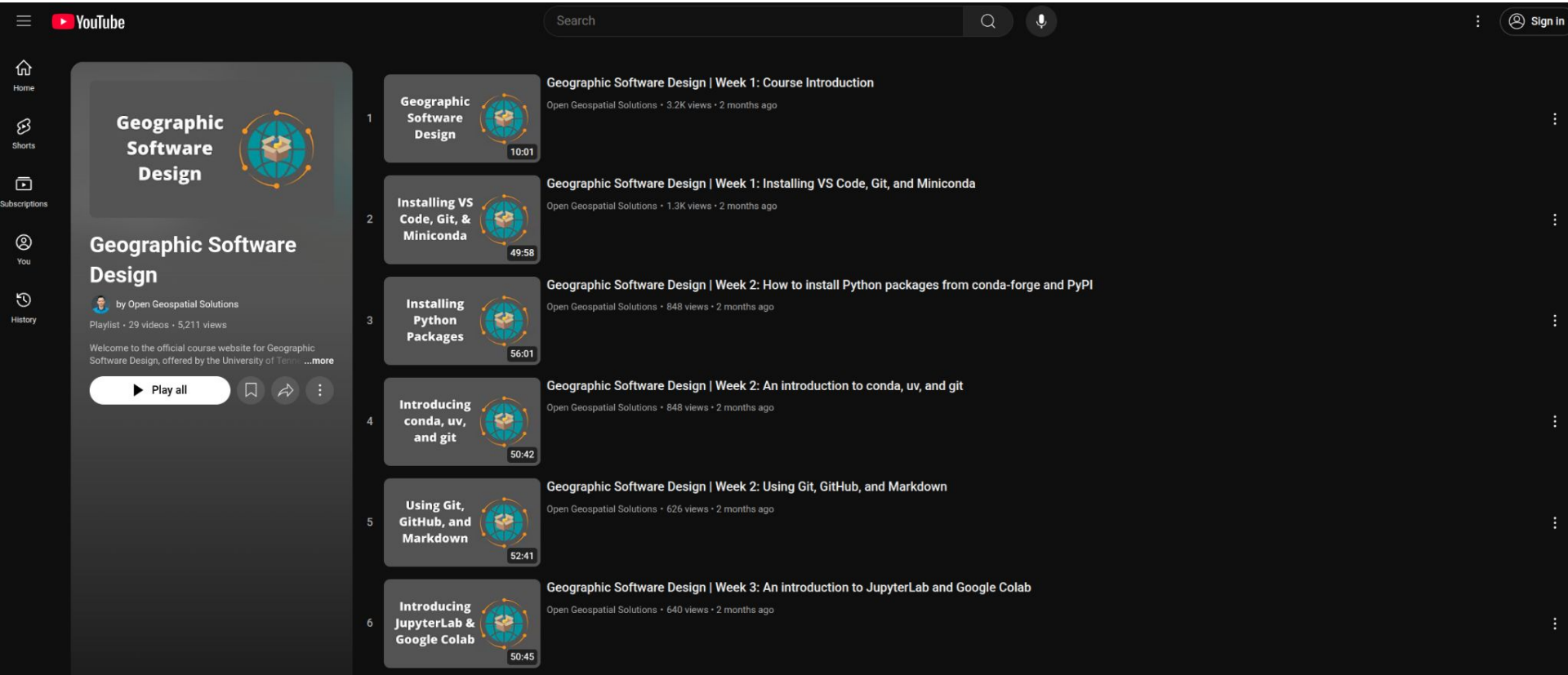
Geographic Software Design

https://geog-510.gishub.org



Geographic Software Design

https://geog-510.gishub.org



Demo

https://opengeoai.org/workshops/GeoAI_Workshop_2025

Thank you!

Any questions

Qiusheng Wu

https://gishub.org