

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 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

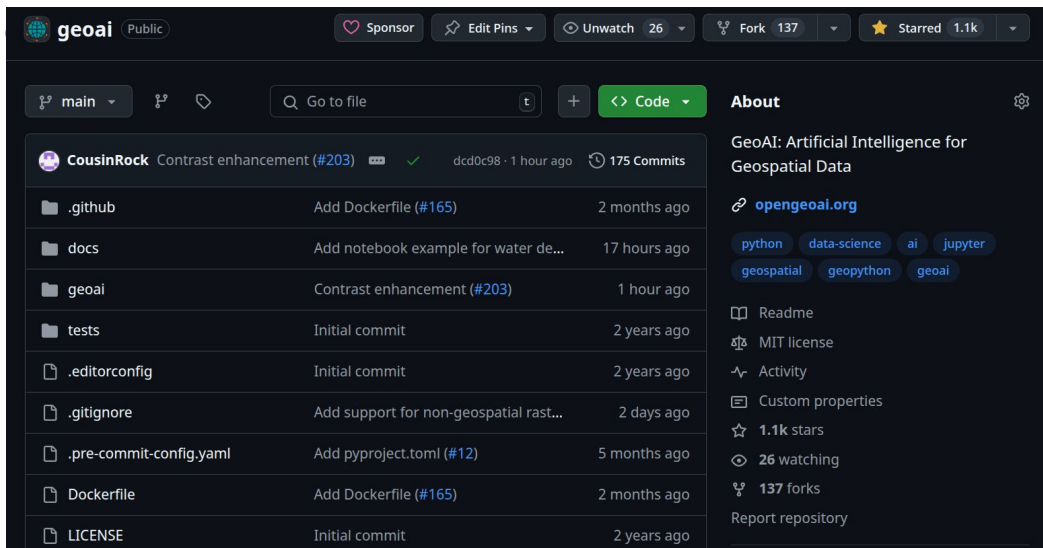
- [Awesome-GEE](#)
- [python-geospatial](#)

<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>



The screenshot shows the GitHub repository page for 'opengeos/geoai'. The repository is public and has 1.1k stars, 137 forks, and 26 watchers. The main branch is 'main'. The repository description is 'GeoAI: Artificial Intelligence for Geospatial Data'. The repository is licensed under MIT. The repository is tagged with 'python', 'data-science', 'ai', 'jupyter', 'geospatial', 'geopython', and 'geoai'. The repository is also tagged with 'Readme', 'MIT license', 'Activity', 'Custom properties', '1.1k stars', '26 watching', '137 forks', and 'Report repository'.

File	Commit Message	Commit Time
.github	Add Dockerfile (#165)	2 months ago
docs	Add notebook example for water de...	17 hours ago
geoai	Contrast enhancement (#203)	1 hour ago
tests	Initial commit	2 years ago
.editorconfig	Initial commit	2 years ago
.gitignore	Add support for non-geospatial rast...	2 days ago
.pre-commit-config.yaml	Add pyproject.toml (#12)	5 months ago
Dockerfile	Add Dockerfile (#165)	2 months ago
LICENSE	Initial commit	2 years ago



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

[notebook](#)

```
1 import geoai
2
3 bbox = (-117.6029, 47.65, -117.5936, 47.6563)
4 geoai.download_naip(
5     bbox=bbox,
6     output_dir="naip_data",
7 )
8 geoai.download_overture_buildings(
9     bbox=bbox,
10    output="buildings.geojson",
11 )
```



Microsoft Planetary Computer

Microsoft | Planetary Computer Explore Data Catalog Applications Documentation

<https://planetarycomputer.microsoft.com/catalog>

Data Catalog

The Planetary Computer Data Catalog includes petabytes of environmental monitoring data, in consistent, analysis-ready formats. All of the datasets below can be accessed via Azure Blob Storage.

 Filter datasets

Featured

Air Quality

Biodiversity

Biomass/Vegetation

Climate/Weather

DEMs

Demographics

Fire

Imagery

Infrastructure

Land use/Land cover

SAR

Snow

Featured



Harmonized Landsat and Sentinel-2 (HLS) v2.0

Harmonized Landsat Sentinel-2 (HLS) Version 2.0 provides consistent surface reflectance (SR) and top of atmosphere (TOA) brightness data from the Operational Land Imager (OLI) aboard the joint NASA/USGS Landsat 8 satellite and the Multi-Spectral Instrument (MSI) aboard the ESA (European Space Agency) Sentinel-2A and Sentinel-2B satellites.

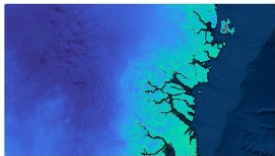
[Sentinel](#) [Landsat](#) [HLS](#) [Satellite](#) [Global](#) [Imagery](#)



Landsat Collection

The Landsat program provides a comprehensive, continuous archive of multispectral imagery of the Earth's surface from 1972 to present.

[Landsat](#) [USGS](#) [NASA](#) [Satellite](#) [Global](#) [Imagery](#) ...



MODIS Version 6.1 Products

The MODIS instrument operates on both the Terra and Aqua spacecraft, covering the entire surface of the Earth within one or two days. The derived data products describe atmosphere, cryosphere, land, and ocean features utilized in studies across various disciplines.

[MODIS](#) [NASA](#) [USGS](#) [Satellite](#) [Global](#) [Imagery](#) ...

```
1 import geoai
2
3 items = geoai.pc_stac_search(
4     collection="naip",
5     bbox=[-76.6657, 39.2648, -76.6478, 39.2724],
6     time_range="2013-01-01/2014-12-31",
7 )
```

Overture Maps

<https://overturemaps.org>



[ABOUT](#) [RESOURCES](#) [NEWS](#) [DATA](#) [DOCS](#)

[BECOME A MEMBER](#)

[X](#) [Twitter](#) [in](#) [GitHub](#) [Search](#)

Overture provides free and open
map data

[Get Started](#)

[Learn More](#)



```
1 import geoai
2
3 bbox = (-117.6029, 47.65, -117.5936, 47.6563)
4 data_file = geoai.download_overture_buildings(
5     bbox=bbox,
6     output="buildings.geojson",
7 )
```

Who is Overture for?

Overture is powering current and next-generation map products by creating reliable, easy-to-use, and interoperable open map data. Our data supports developers who build map services or use geospatial data.



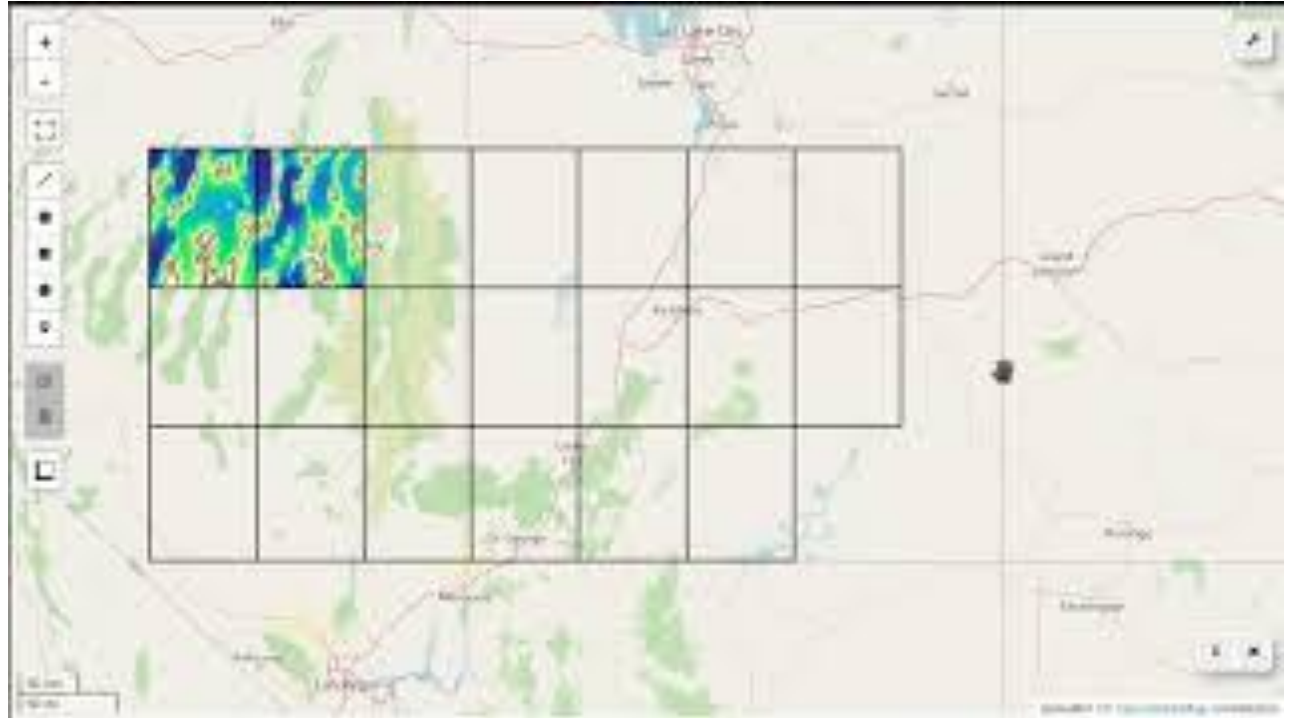
Data Visualization - Planetary Computer

[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

```
1 import geoai
2
3 m = geoai.MapLibre(
4     center=[-100, 40],
5     zoom=3,
6     style="liberty"
7 )
8 m.add_overture_3d_buildings()
9 m.add_globe_control()
10 m
11
```



Data Preparation - Create Vector Data

[notebook](#)

- Create vector data interactively



```
1 import geoai
2
3 m = geoai.MapLibre(style="liberty")
4 m.add_draw_control()
5 properties = {
6     "Type": ["Residential", "Commercial"],
7     "Area": 3000,
8     "Name": "Building",
9     "City": "Seattle",
10 }
11 widget = geoai.create_vector_data(m, properties)
12 widget
13
```

Data Preparation - Edit Vector Data

- Edit vector data interactively



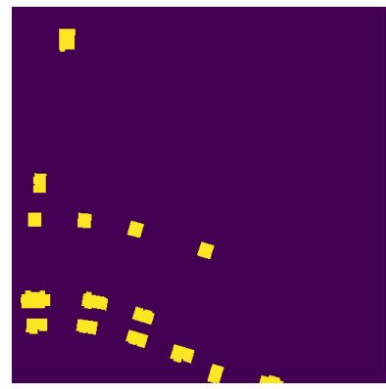
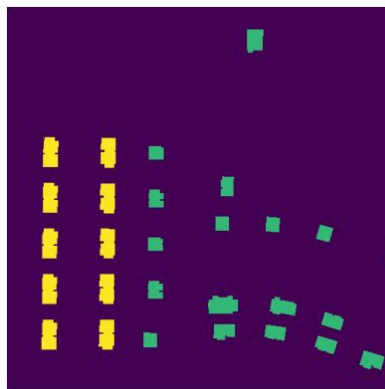
[notebook](#)



```
1 import geoai
2
3 m = geoai.MapLibre(style="liberty")
4 url = "/path/to/your/geojson/file.geojson"
5 props = {
6     "class": ["apartments", "terrace", "house"],
7     "height": 0.0,
8 }
9 widget = geoai.edit_vector_data(m, url, properties=props)
10 m.add_layer_control()
11 widget
```


Data Preparation - Create Image Chips

[notebook](#)



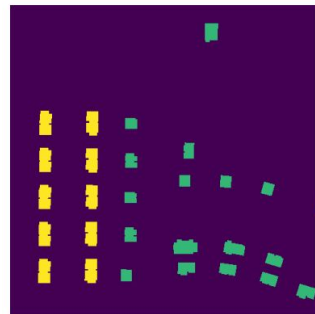
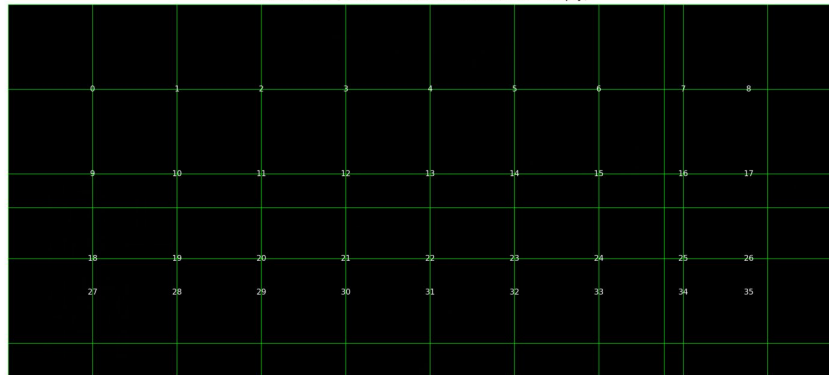
Data Preparation - Create Image Chips

[notebook](#)



```
1 import geoai
2
3 raster_path = "/path/to/image.tif"
4 vector_path = "/path/to/vector.geojson"
5 tiles = geoai.export_geotiff_tiles(
6     in_raster=raster_path,
7     out_folder="output",
8     in_class_data=vector_path,
9     tile_size=512,
10    stride=256,
11    buffer_radius=0,
12    create_overview=True,
13 )
```

Tile Overview (Green = Contains Features, Red = Empty)



Model Training

[notebook](#)



```
1 import geoai
2
3 raster_path = "/path/to/image.tif"
4 vector_path = "/path/to/vector.geojson"
5
6 out_folder = "output"
7 tiles = geoai.export_geotiff_tiles(
8     in_raster=raster_path,
9     out_folder=out_folder,
10    in_class_data=vector_path,
11    tile_size=512,
12    stride=256,
13    buffer_radius=0,
14 )
15 geoai.train_MaskRCNN_model(
16     images_dir=f"{out_folder}/images",
17     labels_dir=f"{out_folder}/labels",
18     output_dir=f"{out_folder}/models",
19     num_channels=4,
20     pretrained=True,
21     batch_size=4,
22     num_epochs=10,
23     learning_rate=0.005,
24     val_split=0.2,
25 )
```

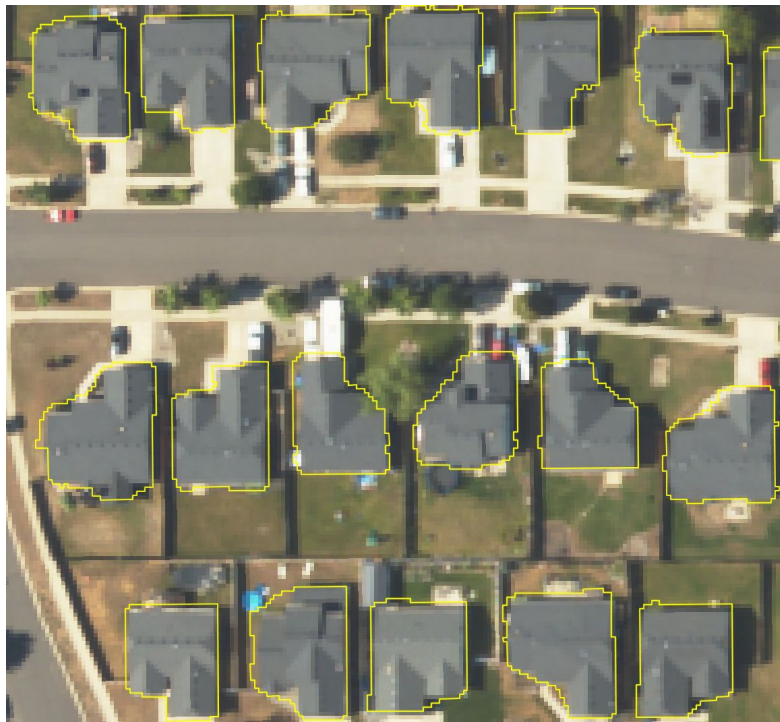


Model Inference



```
1 import geoai
2
3 test_raster_path = "/path/to/test_image.tif"
4 out_folder = "output"
5 masks_path = "naip_test_prediction.tif"
6 model_path = f"{out_folder}/models/best_model.pth"
7 geoai.object_detection(
8     test_raster_path,
9     masks_path,
10    model_path,
11    window_size=512,
12    overlap=256,
13    confidence_threshold=0.5,
14    batch_size=4,
15    num_channels=4,
16 )
```

[notebook](#)

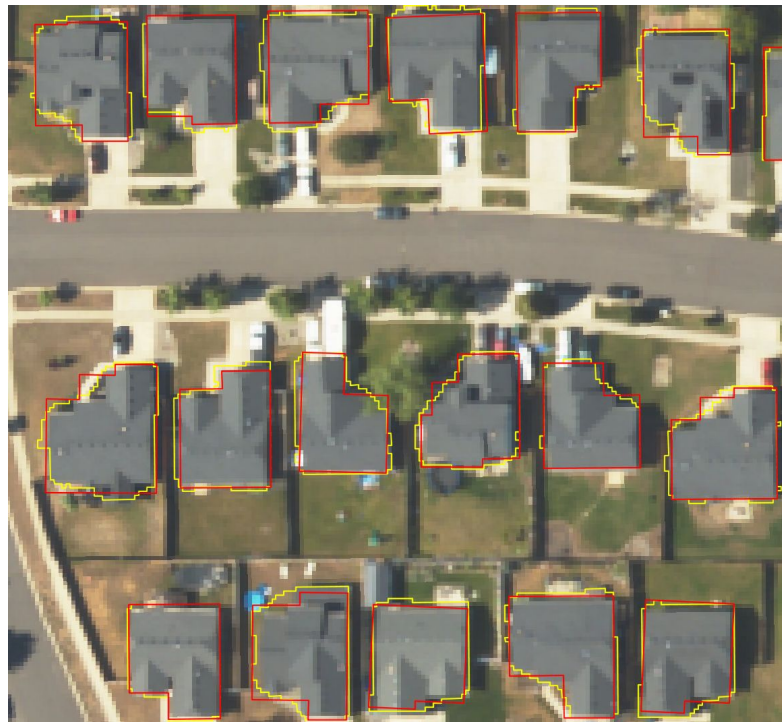


Post-processing - Regularization

[notebook](#)



```
1 import geoai
2
3 vector_path = "/path/to/vector/file.geojson"
4 gdf = geoai.regularize(
5     data=vector_path,
6     simplify_tolerance=2.0,
7     allow_45_degree=True,
8     diagonal_threshold_reduction=30,
9     allow_circles=True,
10    circle_threshold=0.9,
11 )
```



Post-processing - Regularization

[notebook](#)

- Regularize building footprints



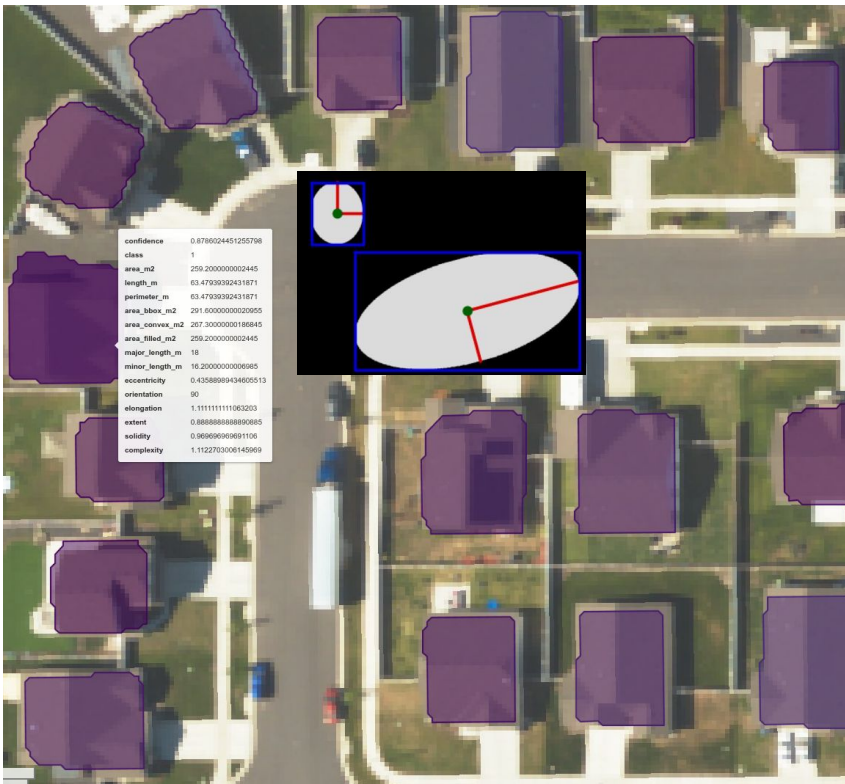
Post-processing - Geometric Properties

[notebook](#)

- Calculate geometric properties



```
1 import geoai
2
3 vector_path = "/path/to/vector/file.geojson"
4 gdf_props = geoai.add_geometric_properties(
5     data=vector_path,
6     area_unit="m2",
7     length_unit="m"
8 )
9 gdf_props.head()
```



Applications

Building Footprint Extraction

[notebook](#)

```
1 import geoai
2
3 raster_path = "/path/to/your/image.tif"
4 masks_path = "prediction.tif"
5 model_path = f"building_footprints_usa.pth"
6 geoai.object_detection(
7     raster_path,
8     masks_path,
9     model_path,
10    window_size=512,
11    overlap=256,
12    confidence_threshold=0.5,
13    batch_size=4,
14    num_channels=3,
15 )
```



Solar Panel Extraction

[notebook](#)

```
1 import geoai
2
3 raster_path = "/path/to/your/image.tif"
4 masks_path = "prediction.tif"
5 model_path = "solar_panel_detection.pth"
6 geoai.object_detection(
7     raster_path,
8     masks_path,
9     model_path,
10    window_size=512,
11    overlap=256,
12    confidence_threshold=0.5,
13    batch_size=4,
14    num_channels=3,
15 )
```



Car Detection

[notebook](#)

```
1 import geoai
2
3 raster_path = "/path/to/your/image.tif"
4 masks_path = "prediction.tif"
5 model_path = "car_detection_usa.pth"
6 geoai.object_detection(
7     raster_path,
8     masks_path,
9     model_path,
10    window_size=512,
11    overlap=256,
12    confidence_threshold=0.5,
13    batch_size=4,
14    num_channels=3,
15 )
16
```



Ship Detection

[notebook](#)

```
1 import geoai
2
3 raster_path = "/path/to/your/image.tif"
4 masks_path = "prediction.tif"
5 model_path = "ship_detection.pth"
6 geoai.object_detection(
7     raster_path,
8     masks_path,
9     model_path,
10    window_size=512,
11    overlap=256,
12    confidence_threshold=0.5,
13    batch_size=4,
14    num_channels=3,
15 )
```



Surface Water Mapping

[notebook](#)

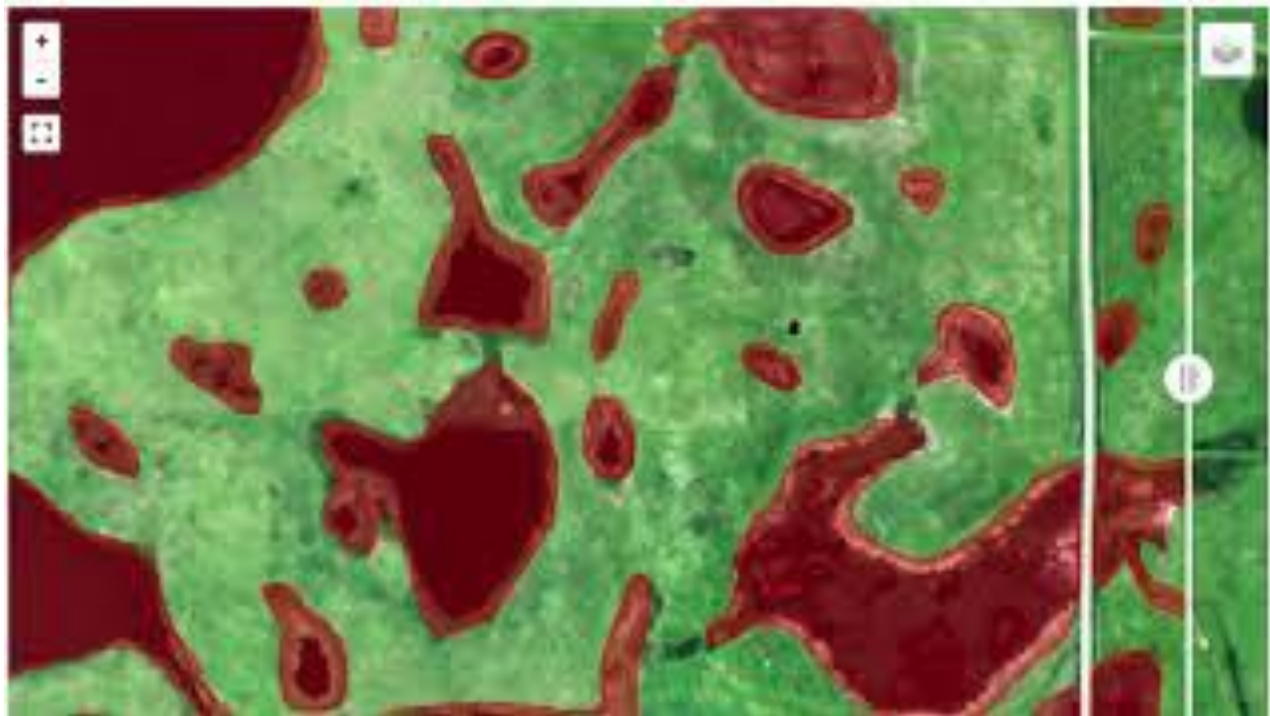
```
1 import geoai
2
3 raster_path = "/path/to/your/image.tif"
4 masks_path = "prediction.tif"
5 model_path = "water_detection.pth"
6 geoai.object_detection(
7     raster_path,
8     masks_path,
9     model_path,
10    window_size=512,
11    overlap=256,
12    confidence_threshold=0.5,
13    batch_size=4,
14    num_channels=3,
15 )
```



Wetland Mapping

[notebook](#)

```
1 import geoai
2
3 raster_path = "/path/to/your/image.tif"
4 masks_path = "prediction.tif"
5 model_path = "wetland_detection.pth"
6 geoai.object_detection(
7     raster_path,
8     masks_path,
9     model_path,
10    window_size=512,
11    overlap=256,
12    confidence_threshold=0.5,
13    batch_size=4,
14    num_channels=3,
15 )
16
```



Land Cover Classification

[notebook](#)



```
1 import geoai
2
3 geoai.semantic_segmentation(
4     input_path=test_raster_path,
5     output_path=masks_path,
6     model_path=model_path,
7     architecture="unet",
8     encoder_name="resnet34",
9     num_channels=4,
10    num_classes=13,
11    window_size=512,
12    overlap=256,
13    batch_size=4,
14 )
```



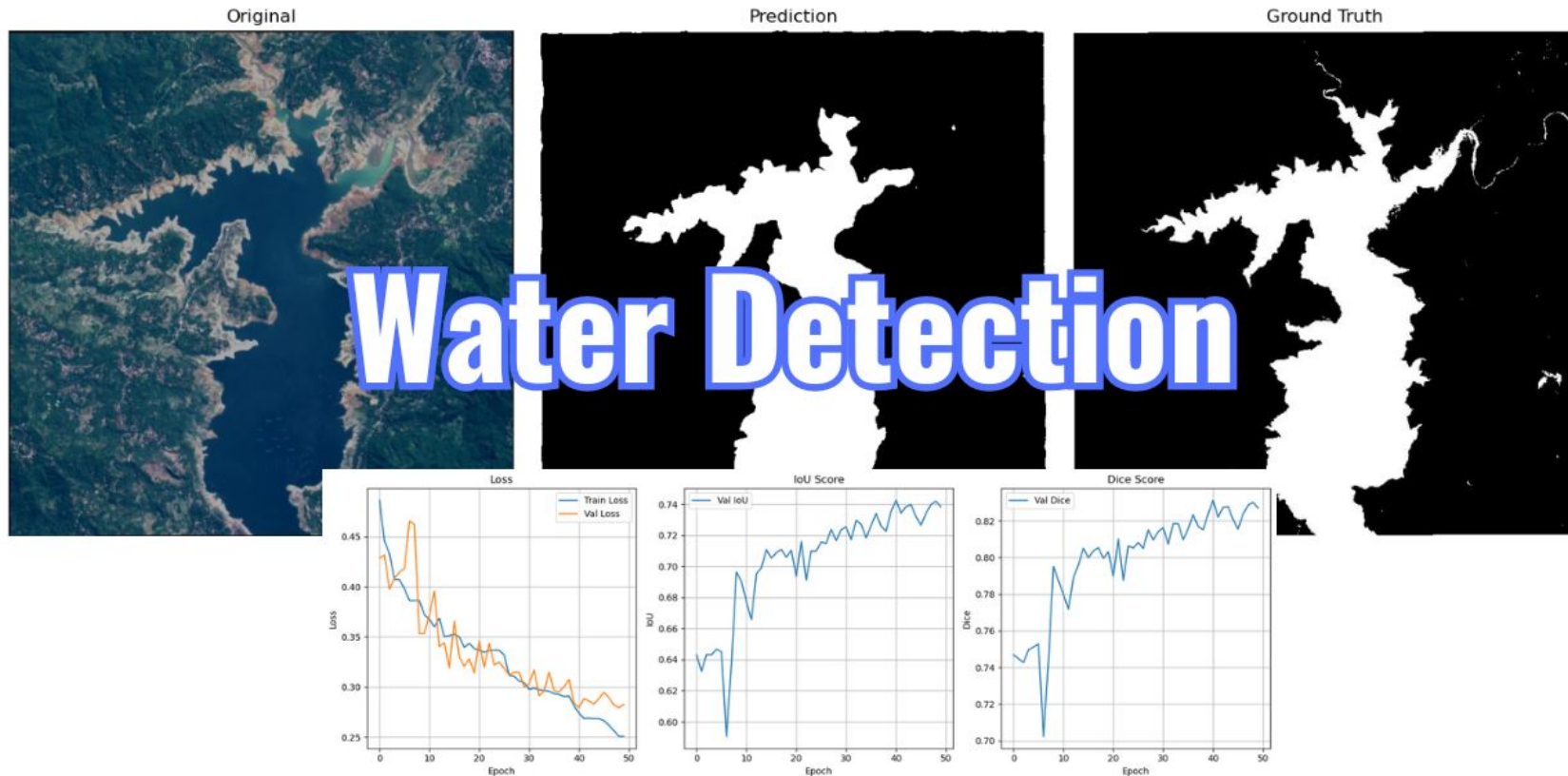
Create Image Chips from Multiple Images and Masks

[notebook](#)



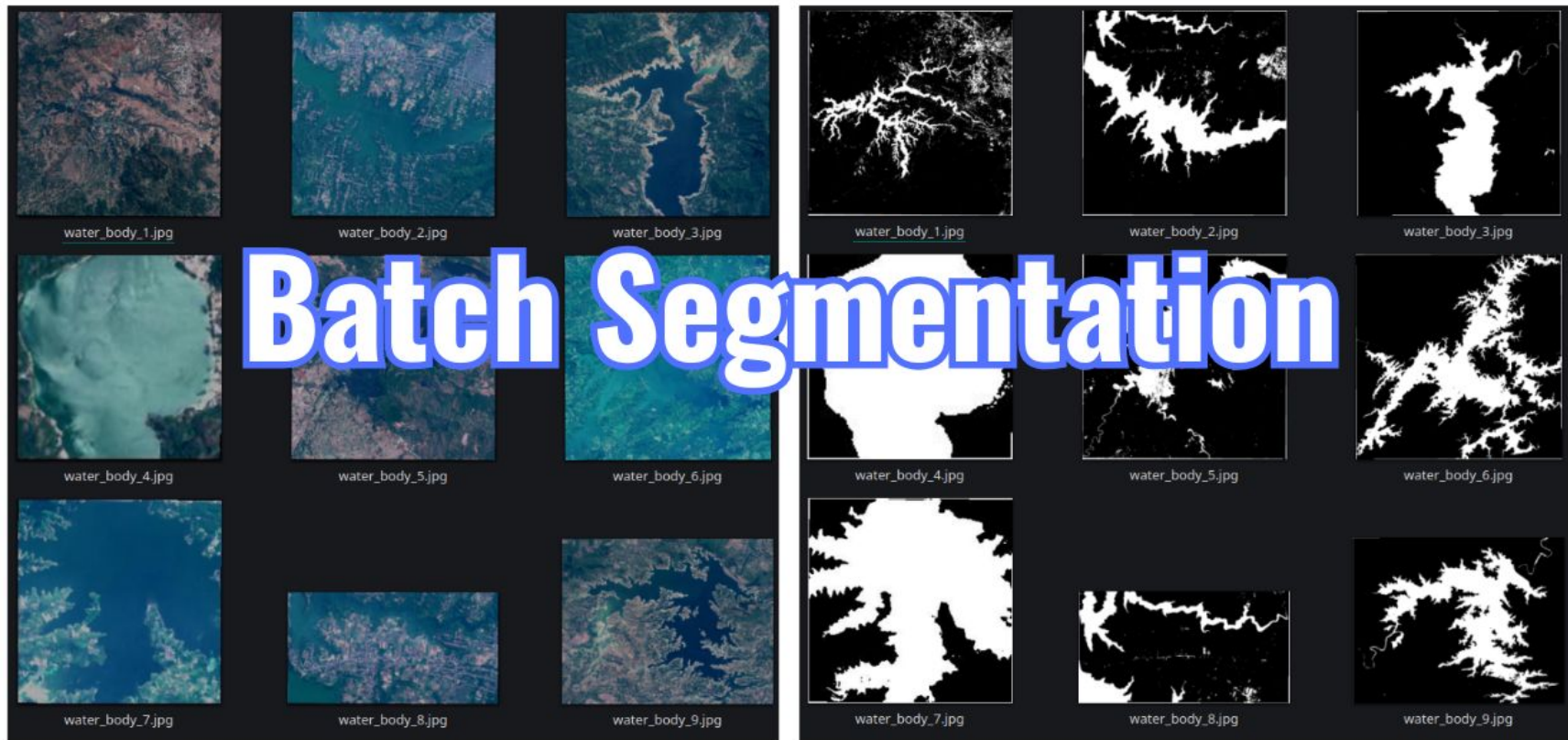
Using Non-geospatial Raster Formats (JPG/PNG)

[notebook](#)



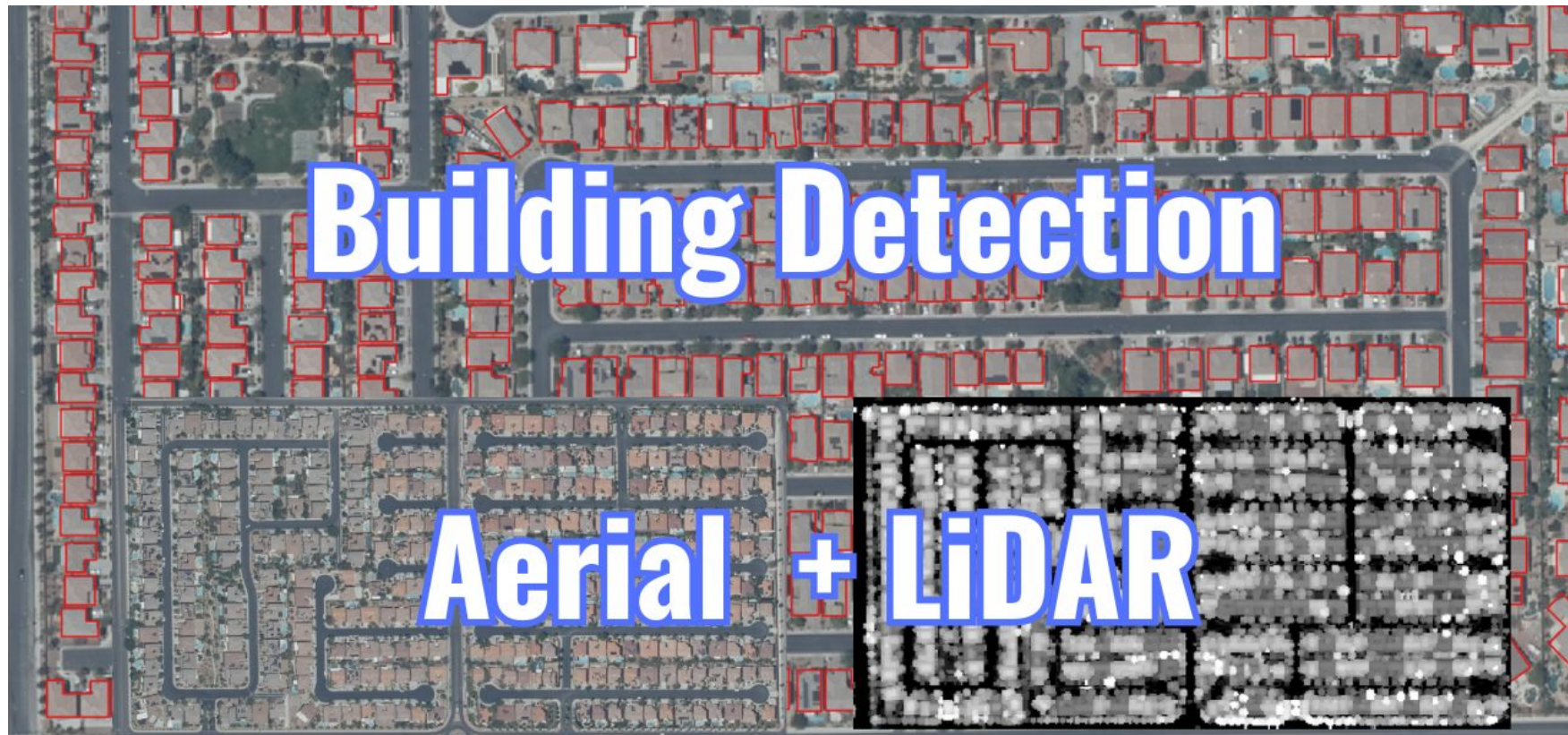
Batch Segmentation on Multiple Satellite Images

[notebook](#)



Building Detection with Aerial Imagery and LiDAR

[notebook](#)



Water Detection Using Sentinel-2 Satellite Imagery

[notebook](#)



Original



Prediction



Ground Truth



Additional Resources

YouTube



<https://youtube.com/@giswqs>



Open Geospatial Solutions

@giswqs · 49.3K subscribers · 860 videos

Dr. Qiusheng Wu is an Associate Professor and the Director of Graduate Studies in the ...more

twitter.com/giswqs and 5 more links

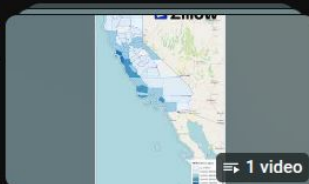
Subscribe



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



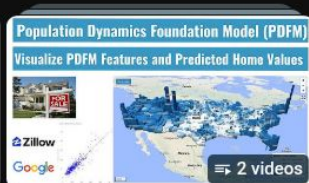
Manjaro Linux Tutorials
View full playlist



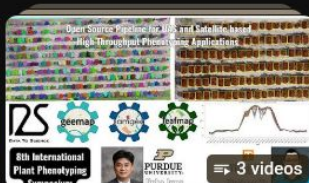
Deepseek and Open WebUI
Tutorials
View full playlist



Geographic Software Design
Updated 2 days ago
View full playlist



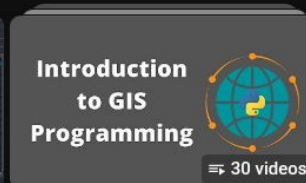
PDFM Tutorials
View full playlist



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>

The screenshot displays the YouTube channel page for 'GeoAI Tutorials' by Open Geospatial Solutions. The channel has 18 videos in its playlist, which are listed on the right side of the page. The videos cover various topics related to GeoAI, including wetland mapping, solar panel detection, data search and download, car detection, surface water mapping, labeled dataset creation, and editing existing labeled datasets. The channel also has a 'Sign in' button and a 'Play all' button for the playlist.

GeoAI Tutorials
by Open Geospatial Solutions
Playlist • 18 videos • 8,865 views
Explore the GitHub:
<https://github.com/opengeos/geosai>...more

GeoAI Tutorial 12: Wetland Mapping with Deep Learning and GeoAI
Open Geospatial Solutions • 2.1K views • 3 weeks ago

GeoAI Tutorial 13: Train a Deep Learning Model for Detecting Solar Panels
Open Geospatial Solutions • 677 views • 2 weeks ago

GeoAI Tutorial 14: Search and Download Data From Microsoft Planetary Computer
Open Geospatial Solutions • 1.4K views • 2 weeks ago

GeoAI Tutorial 15: Train a Deep Learning Model for Detecting Cars
Open Geospatial Solutions • 1.2K views • 2 weeks ago

GeoAI Tutorial 16: Train a Deep Learning Model for Mapping Surface Water
Open Geospatial Solutions • 2.2K views • 2 weeks ago

GeoAI Tutorial 17: Create a Labeled Dataset for Deep Learning
Open Geospatial Solutions • 722 views • 4 days ago

GeoAI Tutorial 18: Edit Existing Labeled Datasets for Deep Learning
Open Geospatial Solutions • 300 views • 3 days ago

Introduction to GIS Programming

<https://gispro.gishub.org/>



Introduction to GIS Programming

Welcome to the official course website for "Introduction to GIS Programming," offered at the University of Tennessee, Knoxville, during Fall 2024.

Course Overview

This course offers a comprehensive exploration of GIS programming, centered around the Python language. Throughout the semester, students will master the use of Python libraries and frameworks for processing, analyzing, and visualizing geospatial data.

What You'll Learn

- **Fundamentals of Python:** Start with the basics—variables, data types, data structures, string looping, and control statements. Progress to more advanced topics, including functions, class handling, and exception handling.
- **Geospatial Applications:** Delve into Python's application in the geospatial realm. Learn to create interactive web maps using Leafmap, visualize vector and raster data, and explore 3D mapping with MapLibre. You'll also get hands-on experience with a number of specialized geospatial libraries: GeoPandas, Rasterio, WhiteboxTools, Geemap, SAMGeo, HyperCoast, DuckDB, Xarray, and more.
- **Hands-On Experience:** Engage in practical labs and projects designed to reinforce the concepts in lectures. These activities culminate in a final project where you'll apply your skills to a real-world geospatial problem.

By the end of the course, you will have a robust understanding of Python programming, specifically for geospatial analysis and visualization, empowering you to confidently tackle various geospatial challenges.

About

- Instructor
- Syllabus
- Schedule
- Textbooks
- Videos
- Acknowledgments

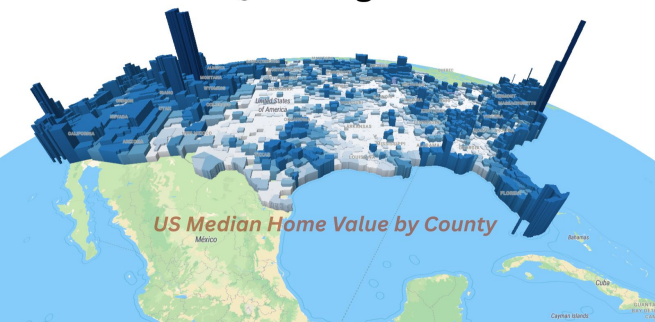
Software

- Overview
- Miniconda
- Visual Studio Code
- Git
- Google Colab
- JupyterLab

Introduction to GIS Programming

A Practical Python Guide to Open Source Geospatial Tools

Qiusheng Wu



Introduction to GIS Programming

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

Introduction to GIS Programming

by Open Geospatial Solutions

Playlist • 23 videos • 20,652 views

This is a course I am teaching at the University of Tennessee in Fall 2024. ...more

1 **Introduction to GIS Programming** 6:17

Intro to GIS Programming | Week 1: Course Introduction

Open Geospatial Solutions • 12K views • 2 months ago

2 **Install VS Code, Git, Miniconda** 11:05

Intro to GIS Programming | Week 1: Install Miniconda, VS Code, and Git

Open Geospatial Solutions • 4K views • 2 months ago

3 **Intro to Conda, Mamba, VS Code, & Git** 1:16:14

Intro to GIS Programming | Week 1: Introduction to Conda, Mamba, VS Code, and Git

Open Geospatial Solutions • 4.5K views • 2 months ago

4 **Intro to Colab, Variables, and Data Types** 1:16:01

Intro to GIS Programming | Week 2: Introduction to Colab, Python Variables, and Data Types

Open Geospatial Solutions • 2.2K views • 2 months ago

5 **Instructions for Submitting Lab Assignments** 11:21

Intro to GIS Programming | Week 2: Instructions for Submitting Lab Assignments

Open Geospatial Solutions • 1K views • 1 month ago

Geographic Software Design

<https://geog-510.qishub.org>



Q Search

Ctrl + K

About

Instructor

Syllabus

Schedule

Textbooks

Videos

Acknowledgments

Software Tools

Overview

Visual Studio Code

Git

Miniconda



Geographic Software Design

Welcome to the official course website for **Geographic Software Design**, offered by the University of Tennessee, Knoxville. This course provides an immersive journey into the world of developing open-source Python packages for geospatial analysis and interactive mapping.

Course Overview

This course offers a comprehensive introduction to creating and designing open-source Python packages for geospatial analysis and interactive mapping. Students will gain hands-on experience with popular Python development environments, including Visual Studio Code, Jupyter Notebook, and Google Colab, to streamline the development, packaging, and distribution of their code. The course emphasizes leveraging existing geospatial libraries, such as [geemap](#) and [leafmap](#), as models for best practices, while fostering contributions to the open-source community.

Throughout the course, students will learn how to design a fully functional Python package, complete with an accompanying documentation website that can be freely hosted on GitHub. The curriculum includes essential topics such as version control with Git, writing clean and reusable code, generating documentation, and sharing projects with collaborators and the broader Python ecosystem.

By the end of the course, students will have developed their own Python package, gaining practical skills highly valued in technical roles such as GIS analyst, data scientist, and software developer. These skills will not only enhance their resumes but also position them competitively in the job market for cutting-edge geospatial and software development roles.



Contents

Course Overview

Learning Outcomes

Course Resources

Enroll for Certification

Lecture Videos

Geographic Software Design

<https://geog-510.qishub.org>

The image shows a YouTube interface with a dark theme. On the left, there is a sidebar with navigation icons for Home, Shorts, Subscriptions, You, and History. The main content area displays a playlist titled 'Geographic Software Design' by 'Open Geospatial Solutions'. The playlist has 29 videos and 5,211 views. Below the title, there is a description: 'Welcome to the official course website for Geographic Software Design, offered by the University of Tennessee...more'. A 'Play all' button is visible. The video list on the right includes the following titles and durations:

- 1. Geographic Software Design (10:01)
- 2. Installing VS Code, Git, & Miniconda (49:58)
- 3. Installing Python Packages (56:01)
- 4. Introducing conda, uv, and git (50:42)
- 5. Using Git, GitHub, and Markdown (52:41)
- 6. Introducing JupyterLab & Google Colab (50:45)

Each video entry also shows the channel name 'Open Geospatial Solutions' and the number of views and time ago (e.g., '3.2K views • 2 months ago').

Demo

https://opengeoai.org/workshops/GeoAI_Workshop_2025

Thank you!

Any questions



Qiusheng Wu
<https://gishub.org>

