Hemanta Pokharel

Dr. Lei Wang

GEOG-4057

May 6, 2025

Project-2

Introduction

Manually collecting elevation data for water boundaries is time-consuming. This project solves this problem by automating the process—fetching elevations from GEE, updating shapefiles, and providing an ArcGIS tool. It uses a CSV file containing water boundary locations taken from a GeoTIFF file named flood_2class.tif. The goal is to write a Python script that fetches elevation data for these points from Google Earth Engine and saves the values to a shapefile. Finally, an ArcGIS toolbox will be created to give users an easy way to interact with the tool.

Methods and Results

Step-1: Importing and Authenticating Google Earth Engine

This step imports the Earth Engine (ee) library, authenticates the user with the GEE by signing in via browser and finally initializes the GEE session with a specific project (vegetation—sav).

```
[23]: import ee

[24]: ee.Authenticate()

[24]: True

[25]: ee.Initialize(project='vegetation--sav')
```

Figure 1: Importing and authenticating Google Earth Engine

Step-2: Loading Elevation Data from GEE

This step loads the USGS 3DEP 10m resolution dataset. And retrieve the metadata about the dataset (e.g., coordinate system, resolution).

```
[42]: dem = ee.Image('USGS/3DEP/10m')
      dem.getInfo()
```

```
[42]: {'type': 'Image',
       'bands': [{'id': 'elevation',
         'data_type': {'type': 'PixelType', 'precision': 'float'},
         'dimensions': [3650412, 939612],
         'crs': 'EPSG:4269',
         'crs_transform': [9.259259259299957e-05,
          0,
          -174.0005555570324,
          0,
```

Figure 2: Loading Elevation Data from GEE

Step-3: Elevation Extraction

This code creates a geometry at coordinates [-91.0989573, 30.3529013], converts the point into a feature collection, and extracts the elevation value for the point using sampleRegions() and prints the result

```
[27]: geom = ee.Geometry.Point([-91.0989573,30.3529013])
      geom_col = ee.FeatureCollection([geom])
      geom_col
```

```
[27]: <ee.featurecollection.FeatureCollection at 0x1e7c4ff3690>
```

Figure 3: Elevation Extraction for a Single Point

Step-4: Reading Water Boundary Data from CSV

The code in this step reads the CSV file (boundary.csv) containing the water boundary points.

```
[29]: #read from the csv file
      import pandas as pd
      table = pd.read_csv(r"C:\Users\hpokha1\OneDrive - Louisiana State University\Documents\GitHub\geog4057\project\data\boundary.csv")
```

Figure 4: Converting CSV Data to a Shapefile

Step-5: Converting CSV Data to a Shapefile

This code converts the CSV data into a GeoDataFrame with point geometries. It sets the coordinate system that matches the raster and saves the point as a shapefile (boundary.shp).

```
[31]: ra1 = arcpy.Raster(r"C:\Users\hpokha1\OneDrive - Louisiana State University\Documents\GitHub\geog4057\project\data\flood_2class.tif")
      ra1.spatialReference.factoryCode

[31]: 32119

[32]: import geopandas
      gdf = geopandas.GeoDataFrame(table)
      gdf.set_geometry(
          geopandas.points_from_xy(gdf['X'], gdf['Y']),
          inplace=True, crs=f'EPSG:{ra1.spatialReference.factoryCode}')
```

Figure 5: Converting CSV to a Shapefile

Step-6: Adding an Elevation Field to the Shapefile

This code uses ArcPy to add a new field (elevation) to the shapefile to store elevation values.

```
[35]: #add field to the shapefile
      arcpy.management.AddField(shapefile,'elevation',"FLOAT")
```

[35]:

Messages

Start Time: Wednesday, April 30, 2025 12:48:58 PM
WARNING 000012: elevation already exists
Succeeded at Wednesday, April 30, 2025 12:48:58 PM (Elapsed Time: 0.03 seconds)

Figure 6: Adding an Elevation Field to the Shapefile

Step-7: Extracting Elevation Values from GEE and Updating the Shapefile

This code iterates over each point in the shapefile using a ArcPy UpdateCursor. For each point it extracts the (X,Y), queries GEE for the elevation at that point. Updates the elevation field in the shapefile with the retrieved value.

```
[41]: #read elevation fro gee and write values to the shapefile
with arcpy.da.UpdateCursor(shapefile,
                           ['SHAPE@XY','elevation'],
                           spatial_reference=4326) as cursor:
    for row in cursor:
        X,Y =row[0]
        geom = ee.Geometry.Point([X,Y])
        geom_col = ee.FeatureCollection([geom])
        elev = dem.sampleRegions(geom_col)
        elevation = elev.getInfo()['features'][0]['properties']['elevation']
        print(elevation)
        row[1]=elevation
        cursor.updateRow(row)
```

```
22.477031707763672
22.477031707763672
22.477031707763672
22.24553871154785
22.477031707763672
```

Figure 7: Extracting Elevation Values from GEE and Updating the Shapefile

Creating a Function and using it in Python Toolbox

Now, the above workflow is converted into a reusable function “get_elevation_from_gee” in the csv_gee.py. Then the python toolbox is created. This toolbox takes the CSV points and convert it to the shapefile, queries GEE for elevations in batch and updates the shapefile with elevation values. As a result, a shapefile with an elevation field is populated by a Google Earth Engine Data. See figure 8 and 9.

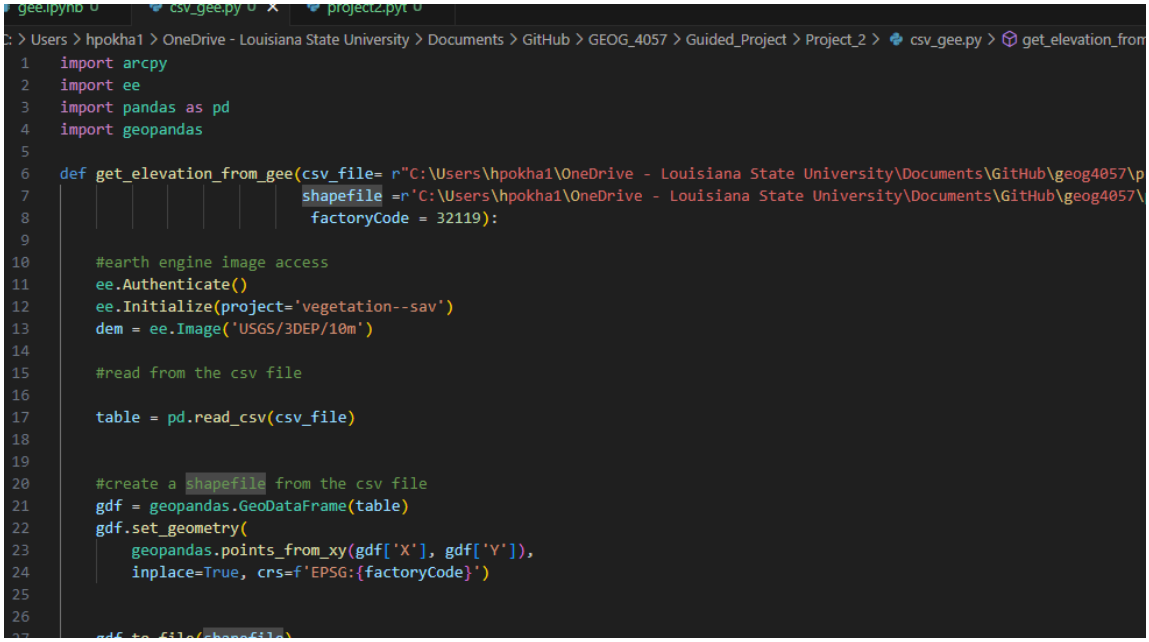

Figure 8: Creating a function

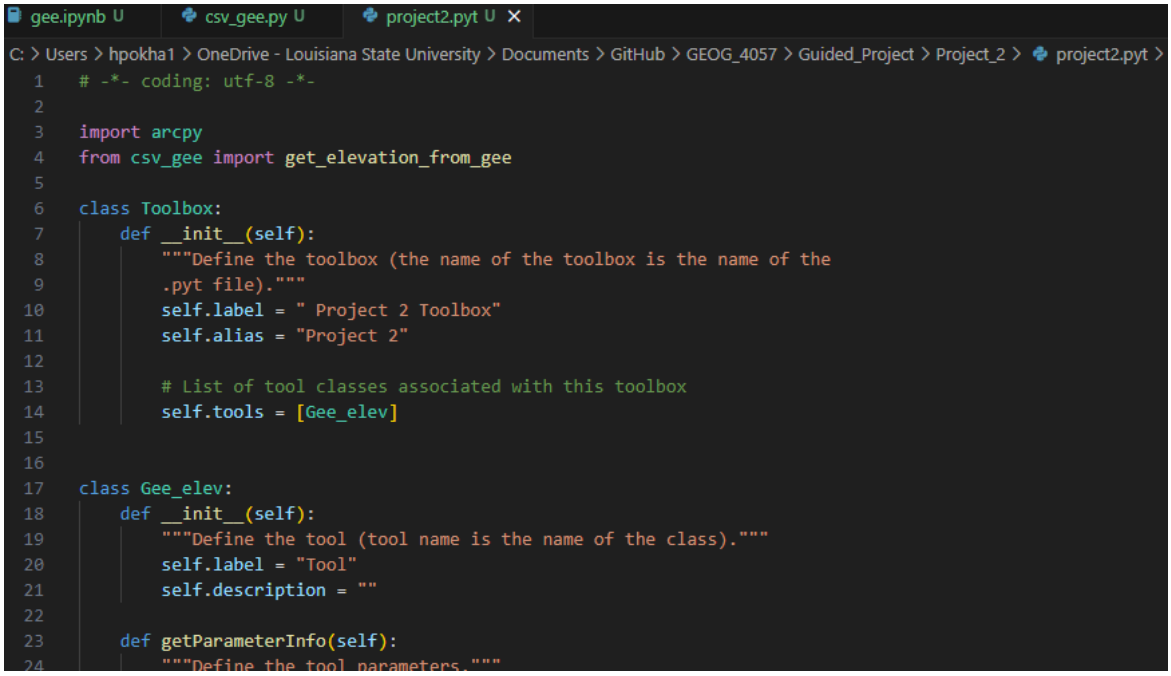

Figure 9: Creating a toolbox

Running a toolbox

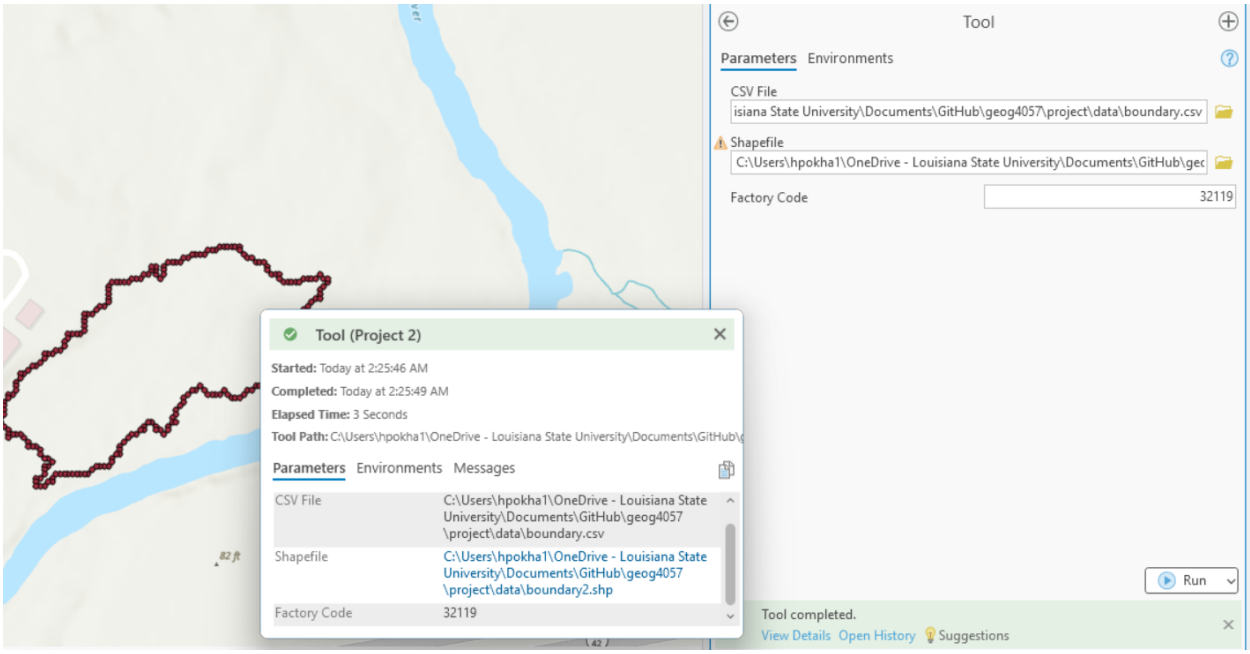

Figure 10: Reference for a successful run of the toolbox

boundary	4/28/2025 1:21 PM	Microsoft Excel C...	16 KB
boundary.dbf	5/5/2025 1:45 PM	DBF File	33 KB
boundary.prj	5/5/2025 1:45 PM	PRJ File	1 KB
boundary.shp	5/5/2025 1:45 PM	SHP File	10 KB
boundary.shp	5/5/2025 1:45 PM	Microsoft Edge H...	3 KB
boundary.shx	5/5/2025 1:45 PM	SHX File	3 KB
boundary1.cpg	5/5/2025 3:07 PM	CPG File	1 KB
boundary1.dbf	5/5/2025 3:07 PM	DBF File	33 KB
boundary1.prj	5/5/2025 3:07 PM	PRJ File	1 KB
boundary1.shp	5/5/2025 3:07 PM	SHP File	10 KB
boundary1.shp	5/5/2025 3:07 PM	Microsoft Edge H...	1 KB
boundary1.shx	5/5/2025 3:07 PM	SHX File	3 KB
boundary2.cpg	5/6/2025 2:25 AM	CPG File	1 KB
boundary2.dbf	5/6/2025 2:25 AM	DBF File	33 KB
boundary2.prj	5/6/2025 2:25 AM	PRJ File	1 KB
boundary2.shp	5/6/2025 2:25 AM	SHP File	10 KB
boundary2.shp	5/6/2025 2:25 AM	Microsoft Edge H...	1 KB
boundary2.shx	5/6/2025 2:25 AM	SHX File	3 KB
csv_gee	5/5/2025 2:45 PM	Python Source File	3 KB
csv_gee_function	5/2/2025 1:10 PM	Python Source File	2 KB

Figure 11: Files created from the toolbox.

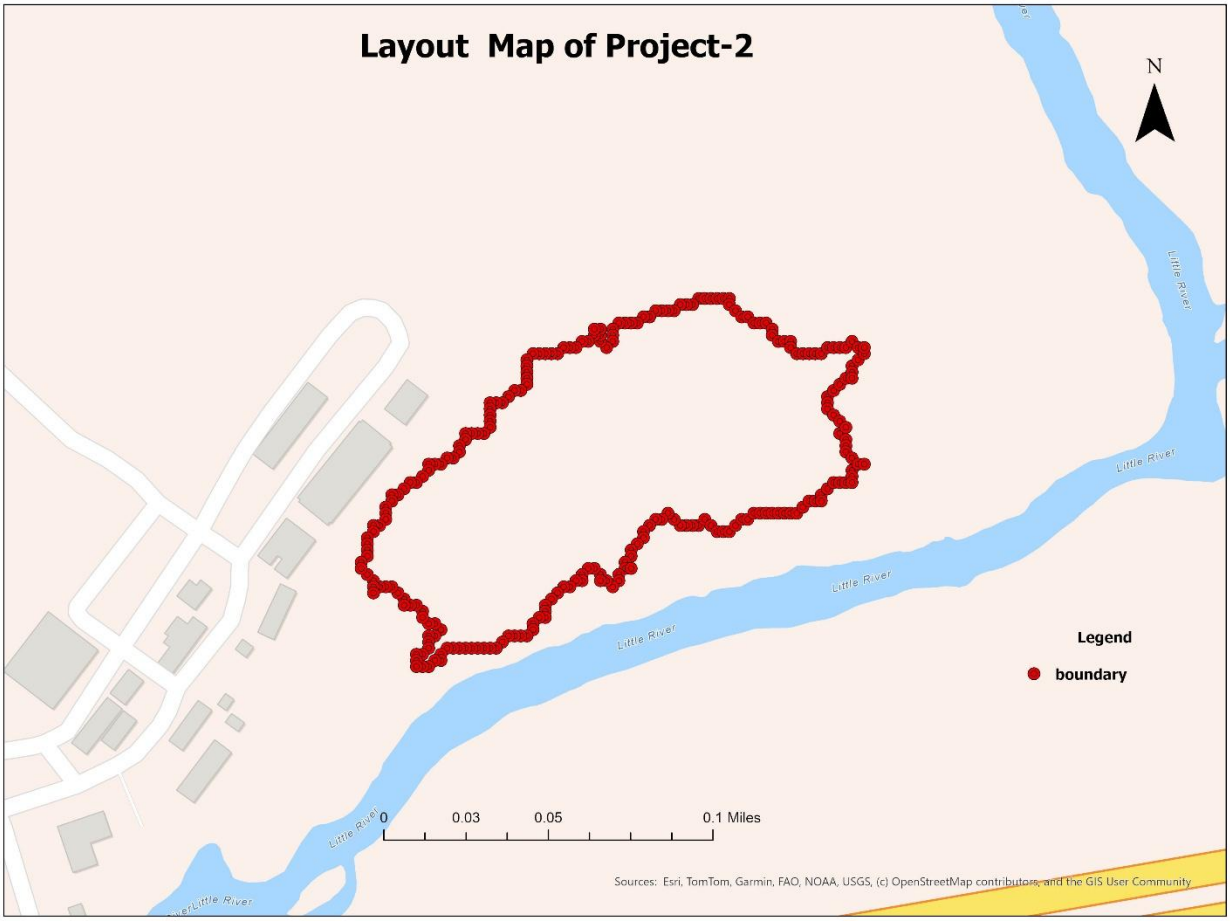

Figure 12: Layout map of the output of the Project-2

Conclusion

The project successfully developed a python toolbox that automates the elevation data extraction for the boundary using Google Earth Engine and ArcGIS. The workflow converts the CSV data to shapefile, retrieves USGS 3DEP elevations, and packages the process into an ArcGIS toolbox.