Week 7

Imagery Exif Information / Metadata Extraction and Features Visualization

Tuesday, February 20th 5:30 PM – 6:45 PM

DUE: Tuesday, February 27th

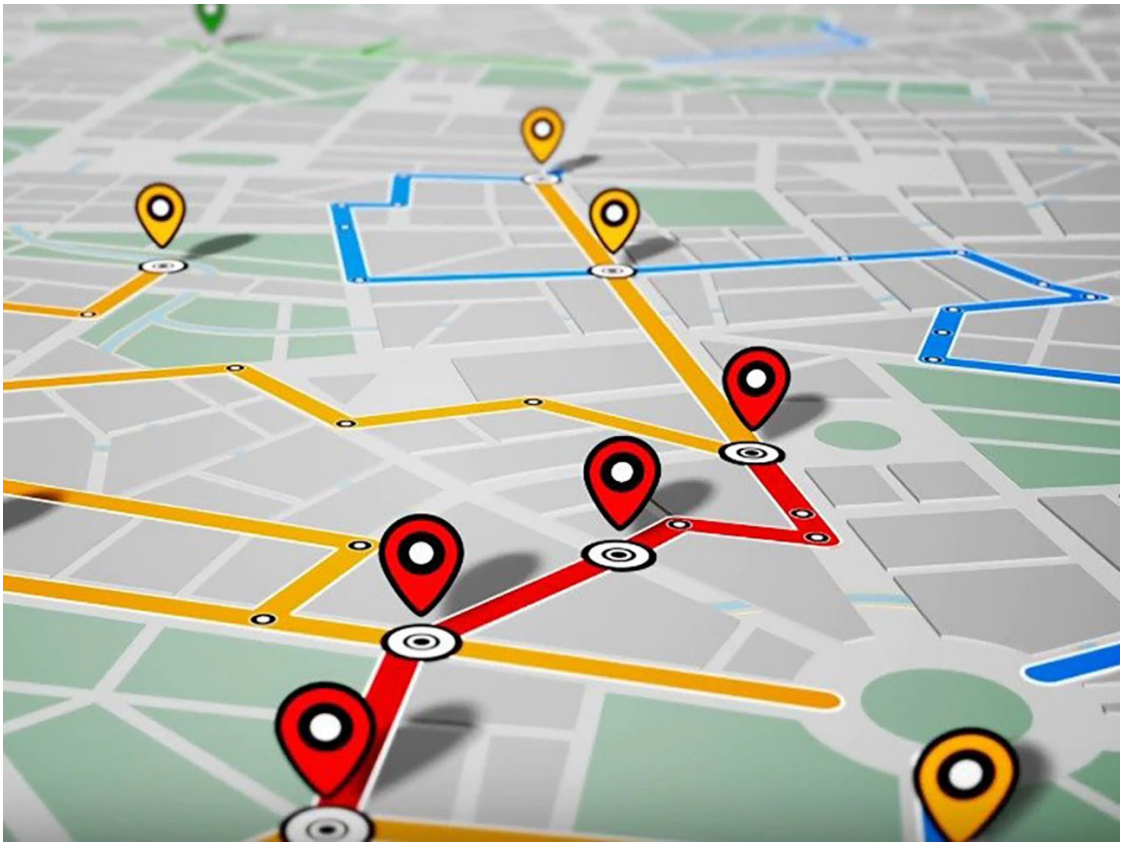


Figure 1. Geolocation information (https://tinyurl.com/47zvxtv2)

DESCRIPTION

Analyzing image metadata and extracting relevant information is an important component of various applications. Information corresponding to the image type, sensor used, data, time, geolocation, etc. are some examples of the information that can be obtained from image metadata. These can then be utilized for mapping and tracking purposes. In this lab, you will be using Python programming language to obtain image metadata from custom images that were acquired from an aerial platform. Using relevant libraries, you will then pinpoint the images onto a map and creating polygons by utilizing the geolocation information. Additionally, you will be using coordinates from Google for creating polygons on maps. Finally, you will be required to calculate the area and perimeter of the polygons that you create.

PREREQUISITES

1. Complete Labs 1, 2, 3, and 4

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

2. Make sure the "dl" anaconda environment has been created from Lab 4

LEARNING OBJECTIVES

In this lab, students will:

1. Download files and install the relevant libraries on the edge device
2. Load aerial images into Jupyter notebooks and extract metadata
3. Extract and map coordinates corresponding to image acquisition location
4. Extract and map coordinates from Google for creating Polygons
5. Complete the homework assignment

ASSIGNMENT

Objective 1: Download files and install the relevant libraries on the edge device

1. **NOTE:** Please make sure that the "dl" anaconda environment from Lab 4 has been created.
2. Power ON the edge device
3. Launch the terminal
4. Navigate to the correct folder and conda environment (**figure 2**):
    a. Create a new directory called lab 5 by typing mkdir lab5 and press "enter"
    b. Navigate to the new directory by typing cd lab5 and press "enter"
    c. Now create activate the anaconda environment created in the last lab by typing conda activate dl and press "enter"

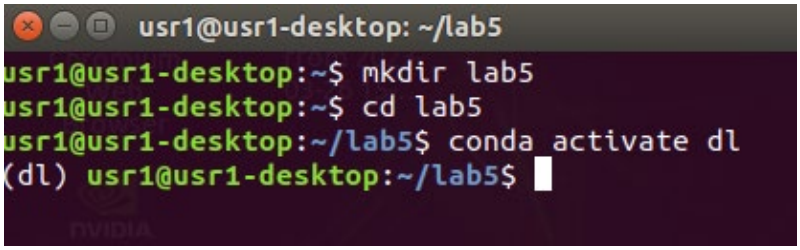


Figure 2. Create Lab 5 folder and enter the dl anaconda environment

5. Download the files from Brightspace:
    a. From Brightspace, download the "**CGT575_Edge_Device_Lab5_Skeleton.ipynb**" file.
    b. From Brightspace, download the image files:
        i. img.jpg
        ii. img1.jpg
        iii. img2.jpg
        iv. img3.jpg

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

v. img4.jpg

c. Save all the files in the lab5 folder that was created by navigating to the Home directory and then the lab5 directory

6. Install the relevant libraries:

a. Install the Pandas library for managing structured data by typing pip install pandas and press "enter"

b. Install the matplotlib library for viewing the images by typing pip install matplotlib and press "enter"

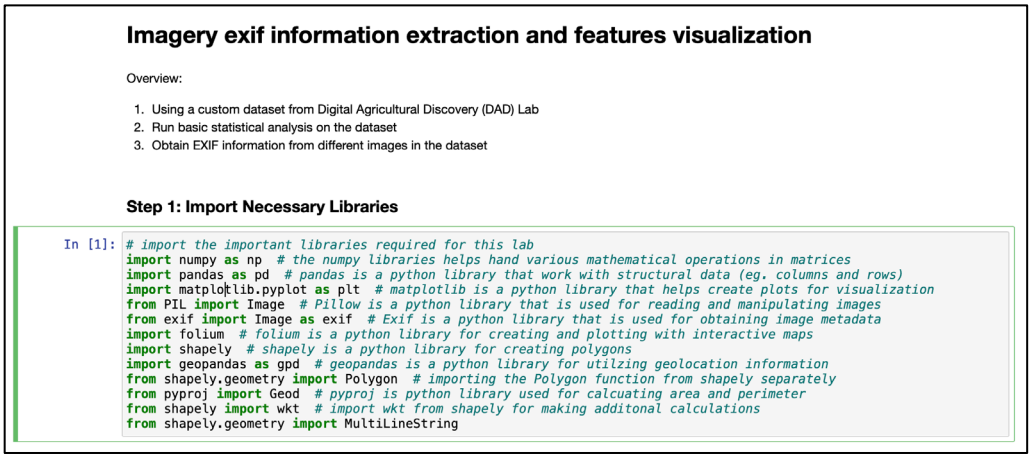
c. Install the Exif library for obtaining image metadata by typing pip install exif and press "enter"

d. Install the folium library for creating interactive maps by typing pip install folium and press "enter"

e. Install the shapely library for creating polygons by typing pip install shapely and press "enter"

f. Install the geopandas library for incorporating geolocation into Python files by typing conda install geopandas and press "enter"

7. Launch the Jupyter Notebook:
   a. Within the terminal, type jupyter notebook and press "enter"
   b. Open the "**CGT575_Edge_Device_Lab5_Skeleton.ipynb**" that you had downloaded
   c. Run the first code block to ensure the libraries were correctly installed (**figure 3**):



Imagery exif information extraction and features visualization

Overview:

1. Using a custom dataset from Digital Agricultural Discovery (DAD) Lab
2. Run basic statistical analysis on the dataset
3. Obtain EXIF information from different images in the dataset

Step 1: Import Necessary Libraries

```
In [1]: # import the important libraries required for this lab
import numpy as np  # the numpy libraries helps hand various mathematical operations in matrices
import pandas as pd  # pandas is a python library that work with structural data (eg. columns and rows)
import matplotlib.pyplot as plt  # matplotlib is a python library that helps create plots for visualization
from PIL import Image  # Pillow is a python library that is used for reading and manipulating images
from exif import Image as exif  # Exif is a python library that is used for obtaining image metadata
import folium  # folium is a python library for creating and plotting with interactive maps
import shapely  # shapely is a python library for creating polygons
import geopandas as gpd  # geopandas is a python library for utilzing geolocation information
from shapely.geometry import Polygon  # importing the Polygon function from shapely separately
from pyproj import Geod  # pyproj is python library used for calcuating area and perimeter
from shapely import wkt  # import wkt from shapely for making additonal calculations
from shapely.geometry import MultiLineString
```

Figure 3. Import necessary libraries

Objective 2: Load aerial images into Jupyter notebooks and extract metadata

1. Read the image "img.jpg" as shown in **figure 4**:

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

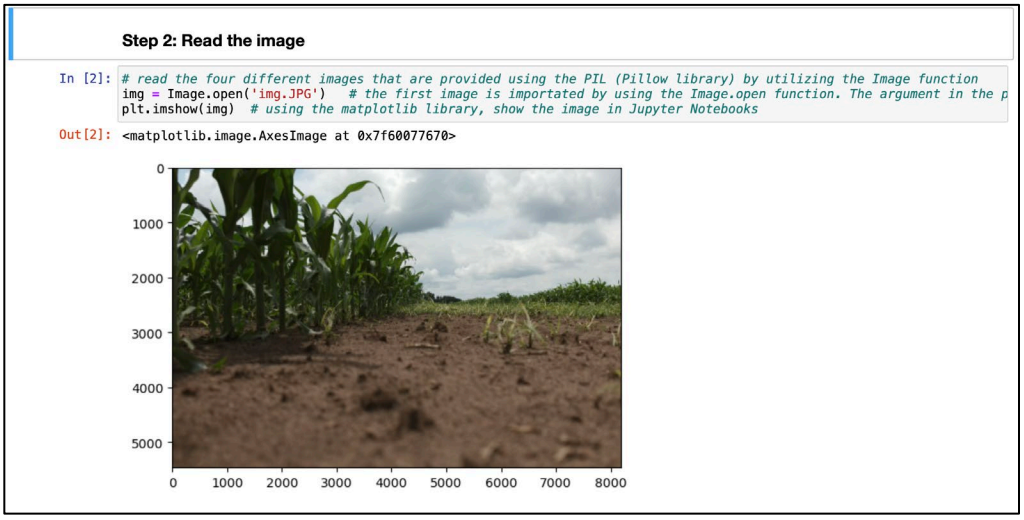


Figure 4. Reading the image

2. Extract the image metadata using the exif library as shown in **figure 5**. You will be able to see the different information that can be extracted from the image:

Figure 5. Extract metadata

3. Specific information can be extracted and printed as shown in **figure 6**:

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

```
In [9]: print(img.gps_altitude_ref, img.gps_altitude)
        GpsAltitudeRef.ABOVE_SEA_LEVEL 186.666

In [10]: print(img.gps_latitude_ref, img.gps_latitude)
         N (41.0, 27.0, 14.3428)

In [11]: print(img.gps_longitude_ref, img.gps_longitude)
         W (86.0, 56.0, 18.9067)
```

Figure 6. Extract specific information from metadata

Objective 3: Extract and map coordinates corresponding to image acquisition location

1. Obtain the geolocation information associated with images in the correct format. The functions for doing this have been created for you as shown in **figure 7**. An example is also shown.

Step 4: Convert the coordinates into decimal coordinates in the correct format

```
In [12]: def decimal_coords(coords, ref):
             decimal_degrees = coords[0] + coords[1] / 60 + coords[2] / 3600
             if ref == "S" or ref == "W":
                 decimal_degrees = -decimal_degrees
             return decimal_degrees

In [13]: def image_coordinates(image_path):
             with open(img_path, 'rb') as src:
                 img = exif(src)

             if img.has_exif:
                 coords = (decimal_coords(img.gps_latitude,img.gps_latitude_ref), decimal_coords(img.gps_longitude,img.gps_lon
             else:
                 print('The image has no EXIF information')

             print(f"Image {src.name}, OS Version:{img.get('software', 'Not Known')} -----")
             print(f"was taken: {img.datetime_original}")
             print(f"Coordiantes: {coords}")

             return coords

In [14]: coordinates = image_coordinates(img_path)
         Image img.JPG, OS Version:02.04.01.02 -----
         was taken: 2022:07:27 13:07:58
         Coordiantes: (41.45398411111111, -86.93858519444444)
```

Figure 7. Convert geocoordinates in the correct format

2. Now use folium to obtain a map from the location corresponding to where the image was acquired as shown in **figure 8**. You will use the folium.Map() function:

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

Figure 8. Create a map using folium library

3. Then use the folium.Marker() function to drop a pinpoint corresponding to the exact location of the image as shown in **figure 9**:

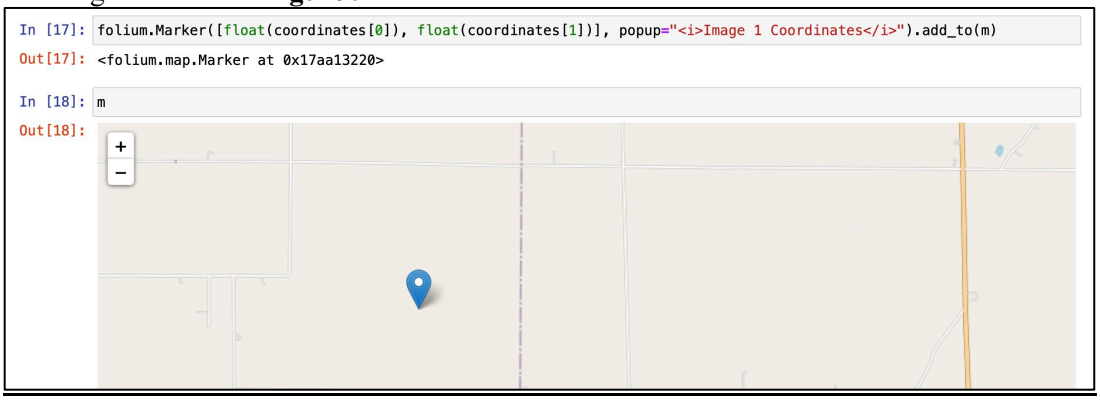


Figure 9. Create a pinpoint / marker on the map

4. Now repeat the steps for the remaining 4 images that were downloaded: "img1.jpg", "img2.jpg", "img3.jpg", and "img4.jpg" as shown in **figure 10**:

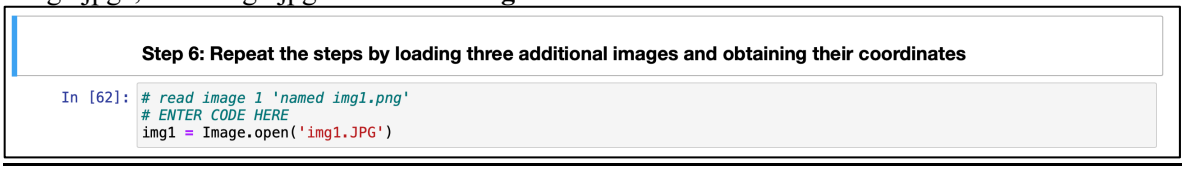


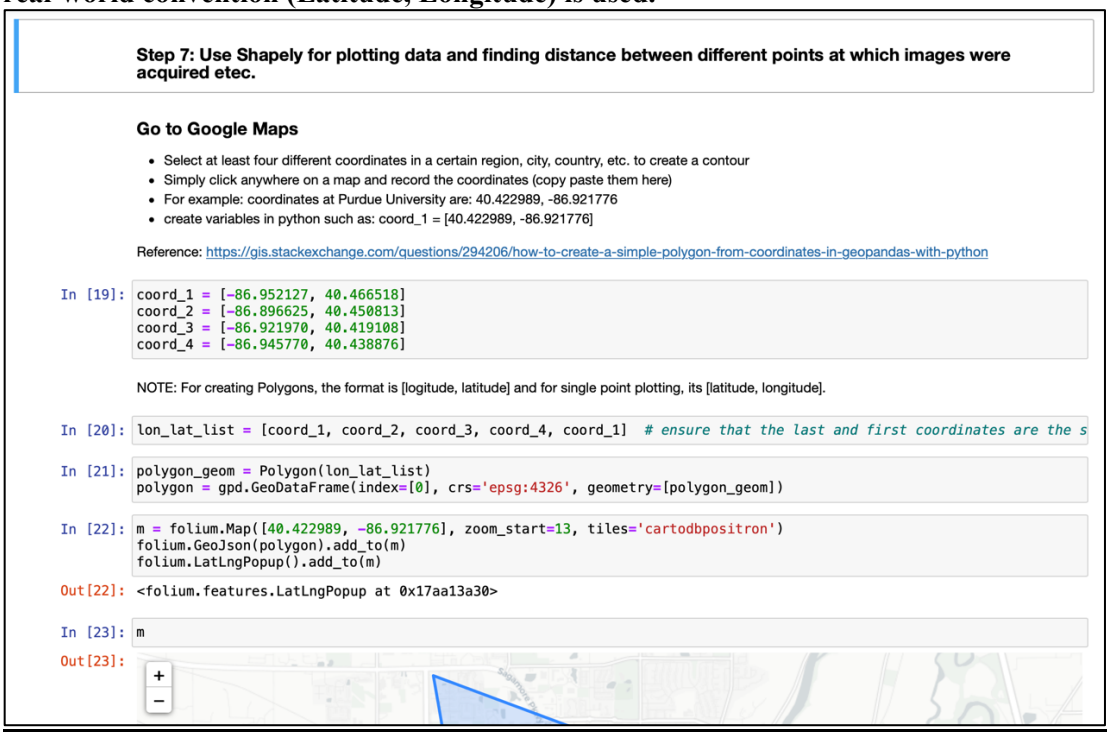
Figure 10. Repeat the mapping for the four additional images that were downloaded

Objective 4: Extract and map coordinates from Google for creating Polygons

1. Follow the steps as shown in **figure 11** for creating polygons:

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

a. Define the points using coordinates from Google (these coordinates are from West Lafayette around Purdue University)
b. Create a Latitude Longitude list called lat_lon_list
c. Add the coordinates to the Polygon() function
d. Once again create a map using folium centered at Purdue University using the folium.Map() function
e. **NOTE: The real-world convention for using geocoordinates is (Latitude, Longitude). However, in this lab as per library requirements, the convention for creating Polygons is (Longitude, Latitude). And, when plotting a single point in folium, the real-world convention (Latitude, Longitude) is used.**



Step 7: Use Shapely for plotting data and finding distance between different points at which images were acquired etec.

Go to Google Maps

- Select at least four different coordinates in a certain region, city, country, etc. to create a contour
- Simply click anywhere on a map and record the coordinates (copy paste them here)
- For example: coordinates at Purdue University are: 40.422989, -86.921776
- create variables in python such as: coord_1 = [40.422989, -86.921776]

Reference: https://gis.stackexchange.com/questions/294206/how-to-create-a-simple-polygon-from-coordinates-in-geopandas-with-python

```
In [19]: coord_1 = [-86.952127, 40.466518]
         coord_2 = [-86.896625, 40.450813]
         coord_3 = [-86.921970, 40.419108]
         coord_4 = [-86.945770, 40.438876]
```

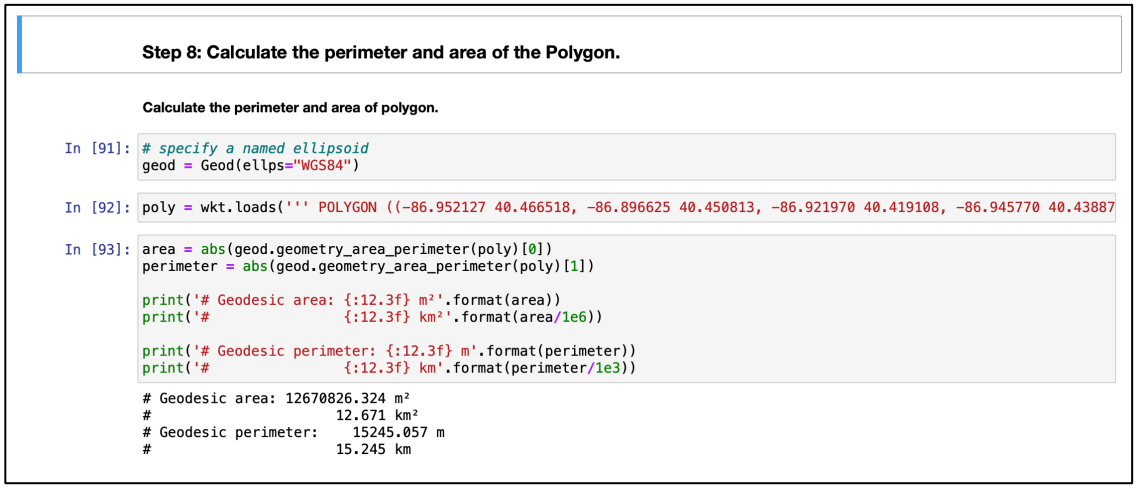
NOTE: For creating Polygons, the format is [logitude, latitude] and for single point plotting, its [latitude, longitude].

```
In [20]: lon_lat_list = [coord_1, coord_2, coord_3, coord_4, coord_1]  # ensure that the last and first coordinates are the s

In [21]: polygon_geom = Polygon(lon_lat_list)
         polygon = gpd.GeoDataFrame(index=[0], crs='epsg:4326', geometry=[polygon_geom])

In [22]: m = folium.Map([40.422989, -86.921776], zoom_start=13, tiles='cartodbpositron')
         folium.GeoJson(polygon).add_to(m)
         folium.LatLngPopup().add_to(m)

Out[22]: <folium.features.LatLngPopup at 0x17aa13a30>

In [23]: m

Out[23]:
```

Figure 11. Creating polygons using coordinates obtained from Google

2. Finally, calculate the perimeter and area using the code shown in **figure 12** below:

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)



Step 8: Calculate the perimeter and area of the Polygon.

Calculate the perimeter and area of polygon.

```
In [91]: # specify a named ellipsoid
         geod = Geod(ellps="WGS84")

In [92]: poly = wkt.loads(''' POLYGON ((-86.952127 40.466518, -86.896625 40.450813, -86.921970 40.419108, -86.945770 40.43887

In [93]: area = abs(geod.geometry_area_perimeter(poly)[0])
         perimeter = abs(geod.geometry_area_perimeter(poly)[1])

         print('# Geodesic area: {:12.3f} m²'.format(area))
         print('#                {:12.3f} km²'.format(area/1e6))

         print('# Geodesic perimeter: {:12.3f} m'.format(perimeter))
         print('#                {:12.3f} km'.format(perimeter/1e3))

         # Geodesic area: 12670826.324 m²
         #                12.671 km²
         # Geodesic perimeter:    15245.057 m
         #                15.245 km
```

Figure 12. Calculating the perimeter and area

Objective 5: Homework Tasks

1. Repeat **Objective 4** by selecting 3 or more coordinates anywhere in the world and create a polygon. Additionally, calculate the area and the perimeter of the polygon as was done in **Objective 4**.
    a. To obtain coordinates from Google Maps, first open Google Maps in the browser
    b. To get the coordinates for a specific point, click on the map where you would like as shown in **figure 13**:

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

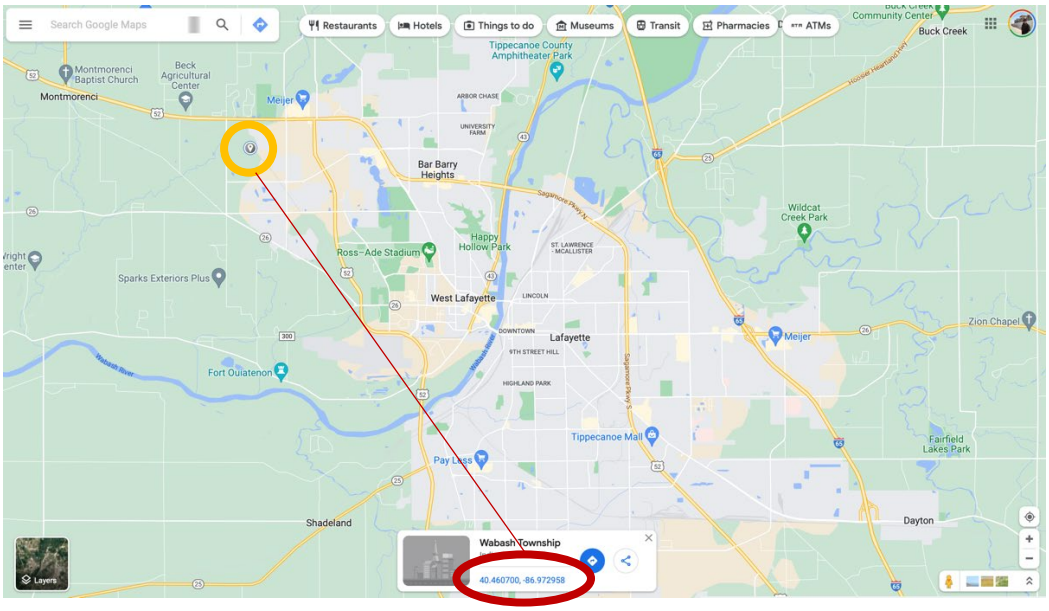


Figure 13. Google Maps click on desired location

c. Now click the coordinates circled in **RED** in **figure 13**

d. Now you will see the side window as shown in **figure 14**

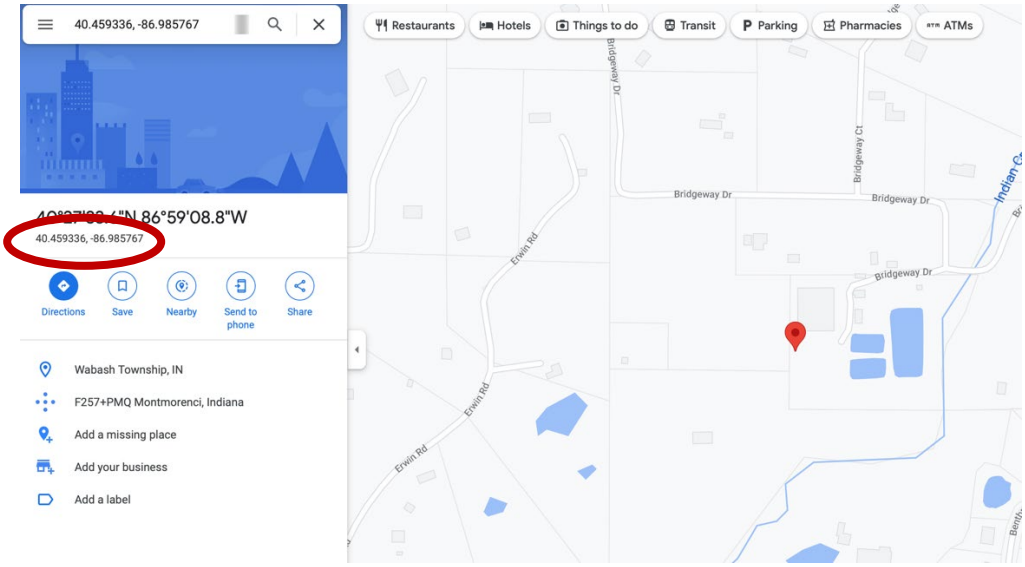


Figure 14. Obtain coordinates

e. Copy the coordinates circled in **RED** in **figure 14**

f. Input the coordinates by following the steps from **objective 4** above.

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)

2. Using the 4 images that were uploaded in **objective 2**, create a polygon by extracting their coordinates as was done in **objective 3**. Additionally, calculate the area and the perimeter of the polygon as was done in **Objective** 4.

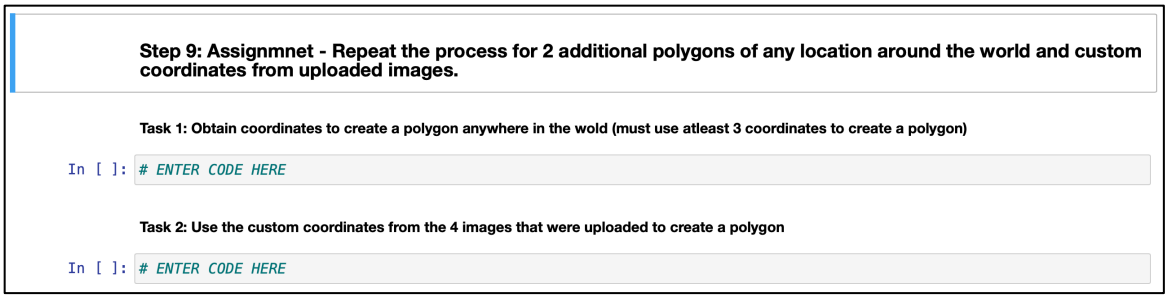


Figure 15. Homework training custom CNN using Fashion MNIST dataset

3. Finally, submit the following on Brightspace:
    a. "**CGT575_Edge_Device_Lab5_Skeleton.ipynb**"

REFERENCES / ADDITIONAL RESOURCES

1.

Instructors:
Dr. Vetria Byrd (vlbyrd@purdue.edu)
Dr. Dharmendra Saraswat (saraswat@purdue.edu)
Dr. Aanis Ahmad (ahmad31@purdue.edu)
Temitope Ibrahim Amosa (tamosa@purdue.edu)