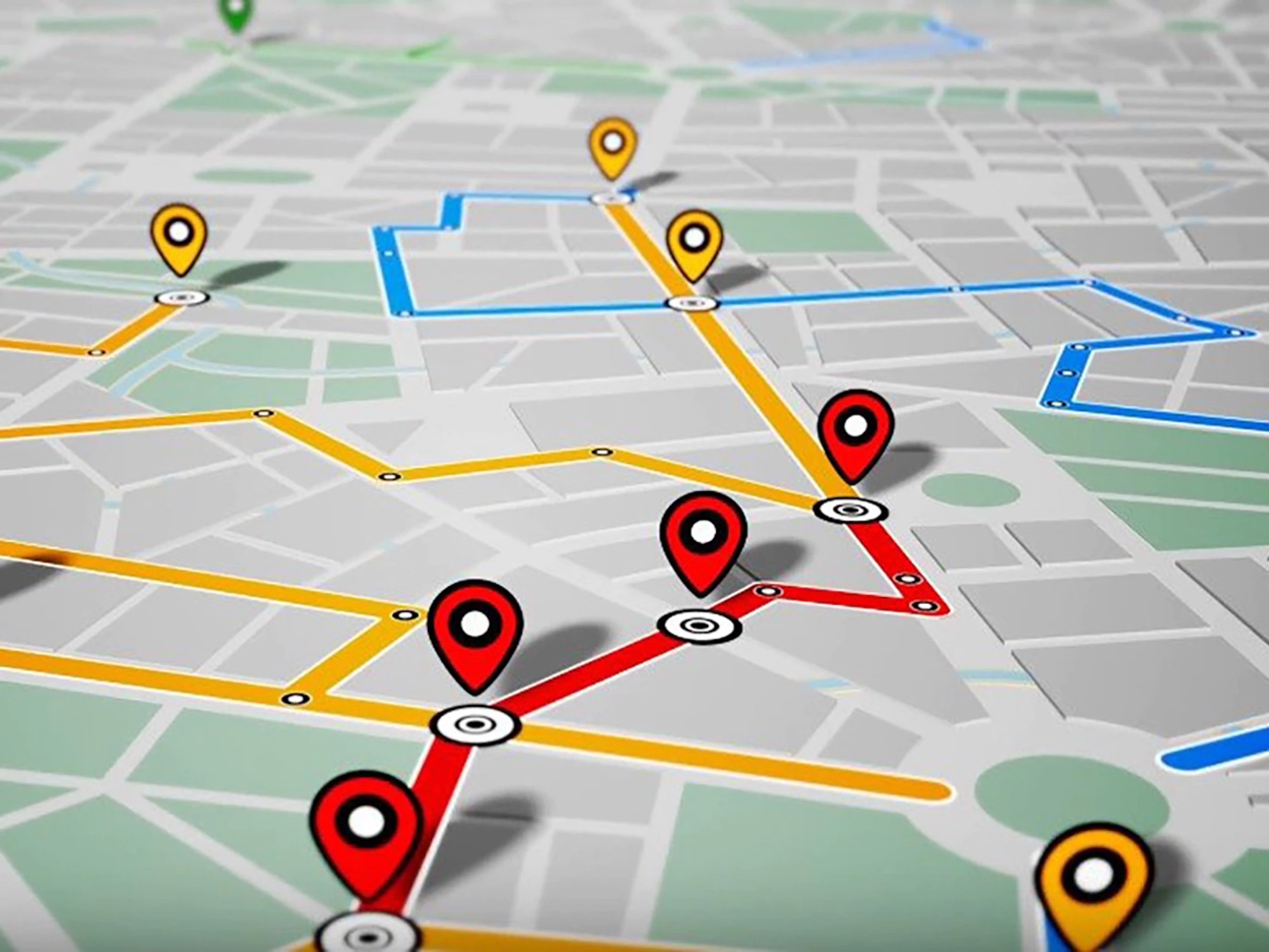
*Week 7*

*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**](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
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**):
   1. Create a new directory called lab 5 by typing mkdir lab5 and press “enter”
   2. Navigate to the new directory by typing cd lab5 and press “enter”
   3. Now create activate the anaconda environment created in the last lab by typing conda activate dl and press “enter”

Text

Description automatically generated

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

1. Download the files from Brightspace:
   1. From Brightspace, download the “**CGT575\_Edge\_Device\_Lab5\_Skeleton.ipynb**” file.
   2. From Brightspace, download the image files:
      1. img.jpg
      2. img1.jpg
      3. img2.jpg
      4. img3.jpg
      5. img4.jpg
   3. Save all the files in the lab5 folder that was created by navigating to the Home directory and then the lab5 directory
2. Install the relevant libraries:
   1. Install the Pandas library for managing structured data by typing pip install pandas and press “enter”
   2. Install the matplotlib library for viewing the images by typing pip install matplotlib and press “enter”
   3. Install the Exif library for obtaining image metadata by typing pip install exif and press “enter”
   4. Install the folium library for creating interactive maps by typing pip install folium and press “enter”
   5. Install the shapely library for creating polygons by typing pip install shapely and press “enter”
   6. Install the geopandas library for incorporating geolocation into Python files by typing conda install geopandas and press “enter”
3. Launch the Jupyter Notebook:
   1. Within the terminal, type jupyter notebook and press “enter”
   2. Open the “**CGT575\_Edge\_Device\_Lab5\_Skeleton.ipynb**” that you had downloaded
   3. Run the first code block to ensure the libraries were correctly installed (**figure 3**):

Text, letter

Description automatically generated

**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**:

Graphical user interface, application, PowerPoint

Description automatically generated

**Figure 4. Reading the image**

1. 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:

Graphical user interface, text, application, email

Description automatically generated

**Figure 5. Extract metadata**

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

Graphical user interface, application

Description automatically generated with medium confidence

**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.

**Graphical user interface, text, application, email

Description automatically generated**

**Figure 7. Convert geocoordinates in the correct format**

1. 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:

**A picture containing chart

Description automatically generated**

**Figure 8. Create a map using folium library**

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

**Graphical user interface

Description automatically generated with low confidence**

**Figure 9. Create a pinpoint / marker on the map**

1. 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**:

**Graphical user interface, text, application

Description automatically generated**

**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:
   1. Define the points using coordinates from Google (these coordinates are from West Lafayette around Purdue University)
   2. Create a Latitude Longitude list called lat\_lon\_list
   3. Add the coordinates to the Polygon() function
   4. Once again create a map using folium centered at Purdue University using the folium.Map() function
   5. **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.**

**Graphical user interface, text, application, email

Description automatically generated**

**Figure 11. Creating polygons using coordinates obtained from Google**

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

Graphical user interface, text, application, email

Description automatically generated

**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**.
   1. To obtain coordinates from Google Maps, first open Google Maps in the browser
   2. To get the coordinates for a specific point, click on the map where you would like as shown in **figure 13**:

Map

Description automatically generated

**Figure 13. Google Maps click on desired location**

* 1. Now click the coordinates circled in **RED** in **figure 13**
  2. Now you will see the side window as shown in **figure 14**

Application

Description automatically generated with low confidence

**Figure 14. Obtain coordinates**

* 1. Copy the coordinates circled in **RED** in **figure 14**
  2. Input the coordinates by following the steps from **objective 4** above.

1. 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.

Graphical user interface, text, application, email

Description automatically generated

**Figure 15. Homework training custom CNN using Fashion MNIST dataset**

1. Finally, submit the following on Brightspace:
   1. “**CGT575\_Edge\_Device\_Lab5\_Skeleton.ipynb**”

*REFERENCES / ADDITIONAL RESOURCES*