imGeo App
By Team Nightraid

Simplifying life since 2023

About Us :

- We are a team of 5 people known as Nightraid who are highly enthusiastic and determined to bring a better change to humanity

Our Team Members

Akshay P Sawai

21MI10005

Bhuneshwar Netam

21MI31026

Divyanshu Singh

21MI31012

Naina Sithara

21MI33013

Rupesh Raj

21MI10044

Table of Contents

01

Problem vs. Solution

02

ImGeo

03

Libraries used and code description

04

Benefits in Surveying

01

Problem vs. Solution

Problem

Many times we have the images of an specific place and we want to know where it was located so that we can have better understanding of that place. The problem with google map is:

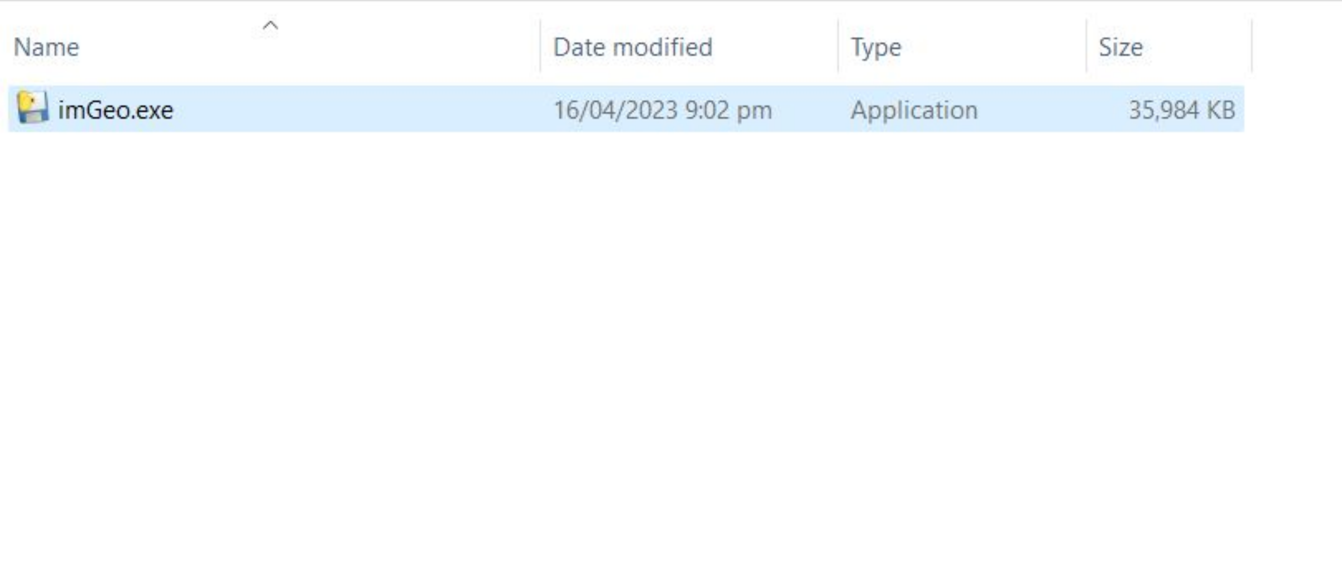
- The latest image updation takes place from 1 month to a year
- Only top views are available of that place if many people do not go that place regularly.

Solution

We have designed our app **imGeo** that takes the geotagged images and puts it on the map.

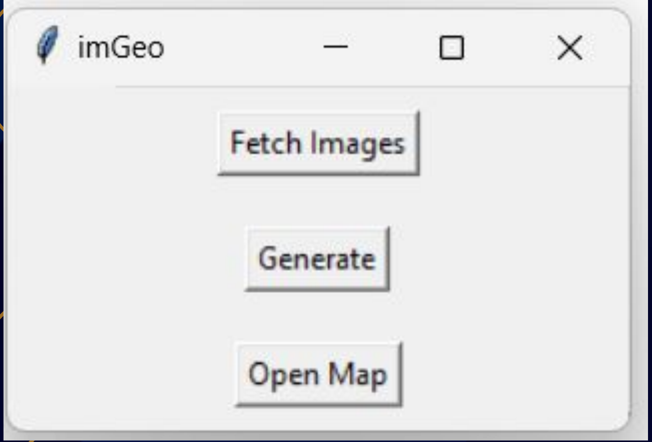
02

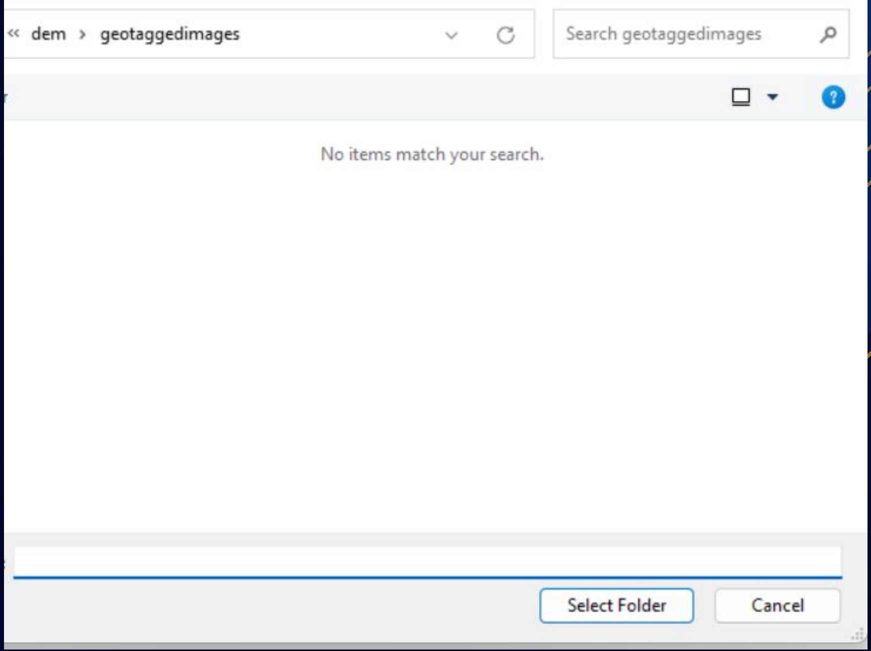
imGeo App



Name	Date modified	Type	Size
imGeo.exe	16/04/2023 9:02 pm	Application	35,984 KB

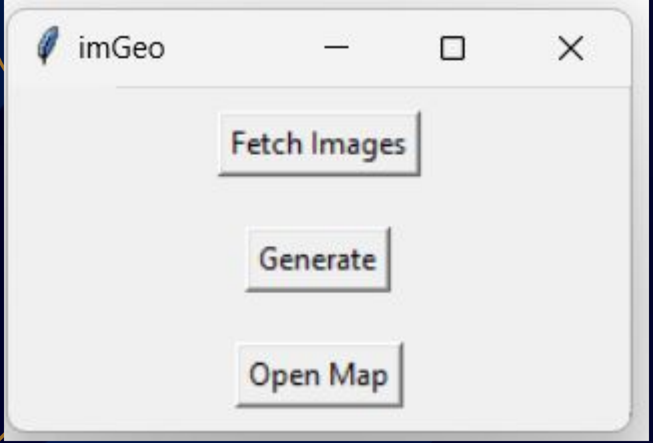
1. Double click and open the imGeo app

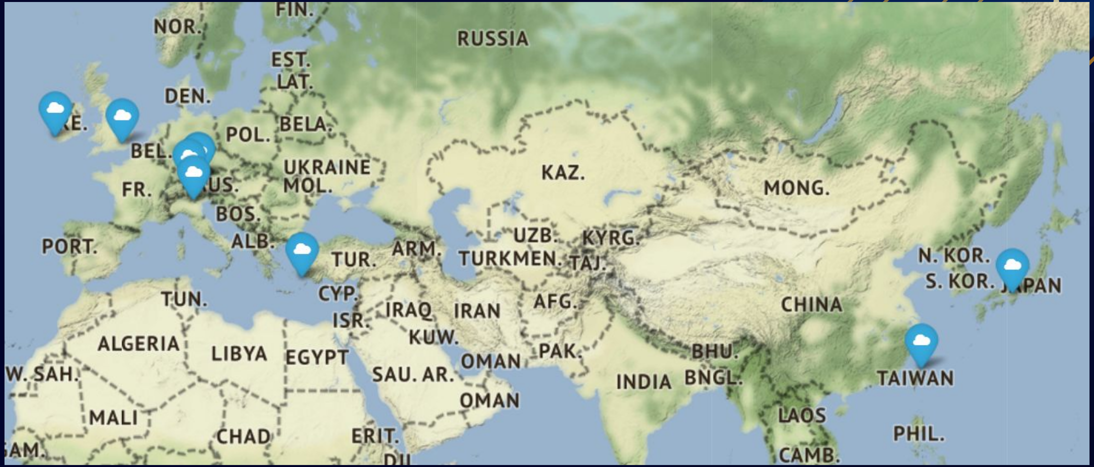




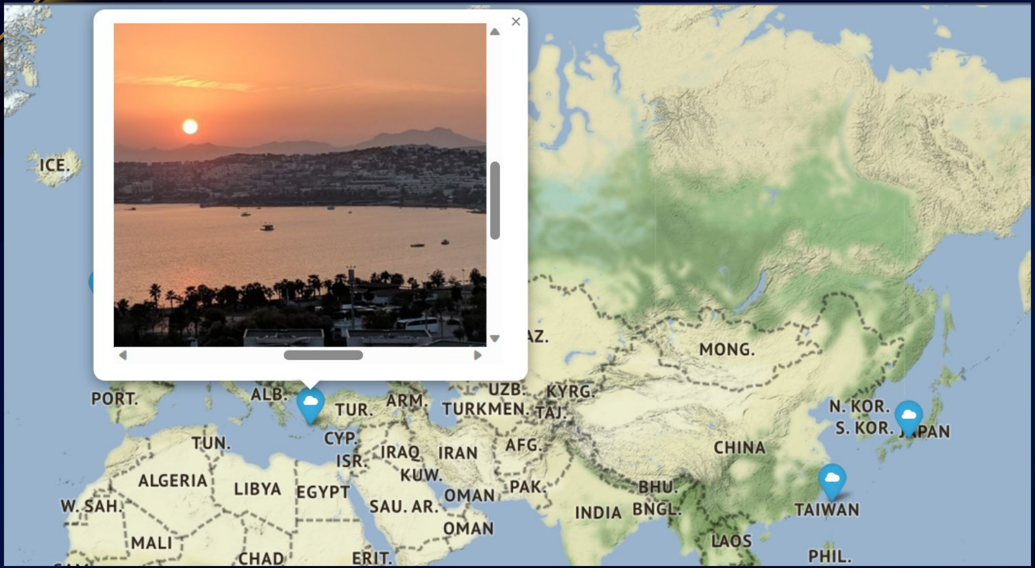
No items match your search.

2. Click on fetch image and select the folder containing geotagged images.





3. Now click on generate and then click on open Map, it will open the map in the default browser.



4. Navigate the map and click on the mark to see the image of that place.

03

Libraries used

sys: Provides access to some variables used or maintained by the interpreter and functions that interact strongly with the interpreter.

webbrowser: Provides a high-level interface to allow displaying Web-based documents to users.

exif: A library for parsing EXIF (Exchangeable Image File Format) data from JPEG and TIFF images.

os: Provides a way of using operating system dependent functionality like reading or writing to the file system.

folium: A powerful Python library used for visualizing geospatial data on interactive maps.

base64: Provides functions for encoding binary data to ASCII characters and decoding ASCII characters back to binary data.

tkinter: A standard Python interface to the Tk GUI toolkit.

Code Description

The code fetches images from a user-selected folder, reads their **EXIF** data to obtain GPS coordinates, and generates an interactive map using folium with markers representing the image locations.

The **decimal_coords()** function converts the GPS coordinates from degrees, minutes, and seconds format to decimal degrees format.

The **image_coordinates()** function reads the EXIF data from an image file and returns the GPS coordinates if available, or returns **None** if not.

The **GUI** (Graphical User Interface) for the application is created using tkinter library with buttons for fetching images, generating the map, and opening the map in a web browser.

The **fetch_images()** function opens a folder dialog to allow the user to select a folder containing the images.

The **generate()** function generates the folium map with markers for each image location, using the GPS coordinates and image file paths obtained earlier.

The **open_map()** function opens the generated map in a new tab of the default web browser.

The **on_closing()** function is called when the user closes the GUI window, and it destroys the window and exits the application.

The GUI window is created using tkinter with a **title** "imGeo" for the application.

Three buttons are created using **tk.Button()** with text labels "**Fetch Images**", "**Generate**", and "**Open Map**", respectively. Each button is associated with a command that specifies which function to call when the button is clicked (**fetch_images()**, **generate()**, and **open_map()** functions).

The **pack()** method is called on each button to add padding between the buttons using the pady parameter.

The **root.protocol()** method is used to bind the "WM_DELETE_WINDOW" event of the window to the **on_closing()** function, which is called when the user closes the window. This ensures that the application exits gracefully.

Finally, the **root.mainloop()** function is called to start the main event loop of the GUI, which handles user interactions and updates the GUI accordingly.

In short, this code uses various libraries to fetch images, extract GPS coordinates from their EXIF data, generate an interactive map using folium, and display the map in a GUI window using tkinter. It provides a simple way to visualize the geotagged images on a map for presentation or other purposes.

04

Benefit in Surveying

>> Geotagging Visualization: The app can help visualize the locations of geotagged images on a map, which can be useful in mine surveying for tracking the locations of geological features, mining equipment, and infrastructure.

>> Data Collection and Analysis: The app allows for fetching and processing of images with geotag information from a selected folder. This can be used in mine surveying for collecting data on rock formations, mineral deposits, or other features of interest, and further analysis can be done based on the geospatial data obtained.

>> Enhanced Reporting: The app generates an interactive map with markers that contain image popups, providing a visual and interactive way to present geotagged image data. This can be beneficial in mine surveying for creating detailed reports, presentations, or visual aids for communication with stakeholders or team members

>> Time and Cost Efficiency: The app automates the process of extracting geotag information from multiple images and generating a map with markers, saving time and effort compared to manual extraction and mapping. This can result in increased efficiency and cost savings in mine surveying workflows.

>> Flexibility and Customization: The app is built using Python and several popular libraries, making it highly customizable and adaptable to specific mine surveying needs. The code can be modified or extended to incorporate additional functionalities or integrate with other tools or workflows used in mine surveying.

>> Enhanced Reporting: The app generates an interactive map with markers that contain image popups, providing a visual and interactive way to present geotagged image data. This can be beneficial in mine surveying for creating detailed reports, presentations, or visual aids for communication with stakeholders or team members.

>> Time and Cost Efficiency: The app automates the process of extracting geotag information from multiple images and generating a map with markers, saving time and effort compared to manual extraction and mapping. This can result in increased efficiency and cost savings in mine surveying workflows.

Overall, the app is a valuable tool in mine surveying for visualizing, analyzing, and presenting geotagged image data, leading to improved data collection, analysis, and reporting processes.

Thanks!