



GEO-LOCATING AN IMAGE USING EXIF DATA

Srivathsa Acharya R
Computer Science & Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga, Karnataka, India

Sudhan J
Computer Science & Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga, Karnataka, India

Suraj S
Computer Science & Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga, Karnataka, India

Vishakha S Aithal
Computer Science & Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga, Karnataka, India

Hiriyanna G S
Computer Science & Engineering
Jawaharlal Nehru New College of Engineering
Shivamogga, Karnataka, India

Abstract— GPS (Global Positioning System) information is a type of EXIF (Exchangeable Image file Format) metadata that's implanted in a computerized photo record. The project aims to create a system that takes user-uploaded pictures, extracts the metadata, shows the coordinates of the location information on a custom Google Maps embedded API, and displays the make and model of the gadget utilized to capture the picture. The system employs Python's `exif` and `Pillow` modules to extract the metadata from the uploaded image file. The metadata incorporates GPS information that contains the latitude and longitude coordinates of the area where the picture was taken. The extracted coordinates are then passed to the Google Maps API to show the area of the picture on map. This data is shown on the webpage, giving extra subtle elements almost the picture.

Keywords: Geo-location, EXIF, Maps, GPS.

I. INTRODUCTION

In the digital age, every action we take online, every message we send, and every image we capture, leaves a digital footprint. This digital footprint can be used to track individuals and their activities. One of the ways to track an individual's digital

footprint is by using images and the metadata associated with them. The EXIF metadata comprises a collection of details that are automatically recorded by contemporary smartphones and digital cameras at the moment of taking an image. This data contains the technical specifications including the type of camera utilized, timestamp, exposure time, lens opening, sensitivity rating, and most significantly, the geographical location of the photograph's origin. The precise whereabouts of the image capture can be ascertained through the extraction of GPS coordinates found within the EXIF metadata. The data could prove beneficial in multiple ways, such as tagging images with their geographic coordinates, sorting pictures based on their location, or for investigative purposes, particularly in the realm of criminal cases.

II. STRUCTURE OF EXIF

EXIF data refers to the metadata incorporated within a digital photograph file. The details provided in the photo include its capture date and time, the brand and model of the camera, the camera settings incorporated while clicking the picture such as aperture, ISO, focal length, and potentially the geographic coordinates of the photo location. EXIF data is usually automatically recorded in smartphones and digital cameras at the time of taking a photo.

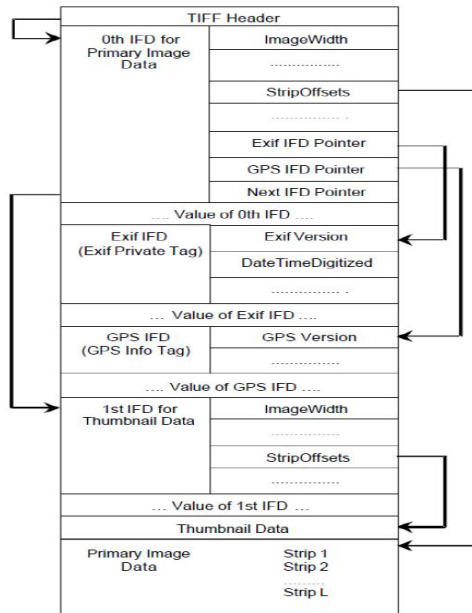


Fig.1 Structure of EXIF Data

The EXIF structure consists of a set of metadata tags that are stored in the image file. These tags hold crucial facts pertaining to the image, such as the precise moment it was captured, the camera device utilized, the configuration settings applied, as well as other significant technical particulars. The EXIF format contains a smaller replica of the primary depiction, known as a thumbnail image. The EXIF metadata markers are classified into several distinct categories or folders known as IFDs (Image File Directories). The primary categories comprise:

- Image IFD: Contains essential information about the image, including its dimensions, color representation, and other metadata.
- Thumbnail IFD: This data includes information on the dimensions, color space, and other details of the thumbnail image.
- GPS IFD: The metadata of a picture entails information about its GPS coordinates, including the latitude and longitude.
- Exif SubIFD: Includes information about the camera and its configuration utilized to capture the image, including the brand, model, length of exposure, and aperture.
- Interoperability IFD: Encapsulates information concerning the image's ability to interface with various devices or software.

As the photo file is saved to the device's storage, either the camera or smartphone captures and stores the EXIF data. Users have the capacity to alter or erase the EXIF data linked with a photo on certain smartphones and digital cameras. As an illustration, individuals might have the capability to modify the recorded date and time within the EXIF information, or

delete all EXIF data completely. There are numerous practical applications for utilizing EXIF data. Photographers could refer to the EXIF data to gather information on the specific camera settings employed to capture an image or to pinpoint the geographical coordinates of the photo's origin. Photo-sharing websites and software applications can utilize EXIF data to categorize and arrange photos according to different factors, including the camera and its configurations, or the precise time and date the photos were captured.

III. PROBLEM STATEMENT

Identifying the location of an image using EXIF metadata is becoming more important as there is a growing demand to geo-locate images for a range of uses, including travel, journalism, and research. The EXIF metadata of an image provides in-depth information such as the camera or device utilized to capture the image, technical specifications, and the geographic coordinates of the image's origin. Precise techniques are required to extract and decipher location information from EXIF metadata pertaining to diverse images and gadgets. Another aspect that requires attention is the issue of privacy surrounding the utilization of location data contained within images, necessitating the creation of protocols and recommendations to ensure the appropriate handling of this data.

IV. OBJECTIVES

- Design and develop a reliable system for geo-locating the images.
- To provide graphical view of the image location using Google maps.
- To provide additional information about the device used to take the photograph.

V. EXIF AND PILLOW MODULE

Python features a "exif" module which enables developers to retrieve as well as manage the EXIF metadata contained in image files. This particular module utilizes the widely-used Python Imaging Library (PIL) or Pillow module as its foundation, offering fundamental functionalities for image manipulation. Python's EXIF module facilitates effortless retrieval of metadata contained in image files, encompassing GPS location data, camera configurations, and other information.

The "Pillow" library, which is widely utilized for manipulating and processing images, is a favoured Python module. Pillow possesses a notable attribute in its capability to manage geo-tagged images, denoting images that encompass location information integrated into their metadata.

The Pillow library is a modified version of the Python Imaging Library (PIL) that enables users to load, edit, and store images in different file formats. Smart paraphrasing: Pillow works well with both Python 2 and 3 and offers an array of extra features



compared to PIL, such as the capability to handle geo-tagged images.

VI. SELENIUM

Selenium is an influential Python library that empowers programmers to execute browser automation. This versatile tool finds application in numerous areas, including web data extraction, quality assurance, and streamlining of repetitive tasks. The primary focus of this report is to explore the implementation of Selenium in web scraping activities for acquiring Google Maps' embedded links and downloading images of mobile models. The Selenium module possesses a crucial characteristic of being able to simulate user interactions and interact with web pages. It allows for the exploration of a website, insertion of search terms, and retrieval of information from the ensuing pages. Selenium has the capability to scan Google Maps integrated links to locate a specific destination, activate the "Share" tab on the Google Maps interface, and retrieve the embedded link from the "Embed a map" category.

The tool Selenium also has the capability of downloading images depicting various models of mobile phone. Numerous online platforms, including GSM Arena, offer intricate information regarding mobile phone models along with corresponding visuals. Selenium provides the means to streamline the task of accessing a website, using search to locate a specific phone model, and procuring its relevant images through automation.

VII. SYSTEM DESIGN AND IMPLEMENTATION

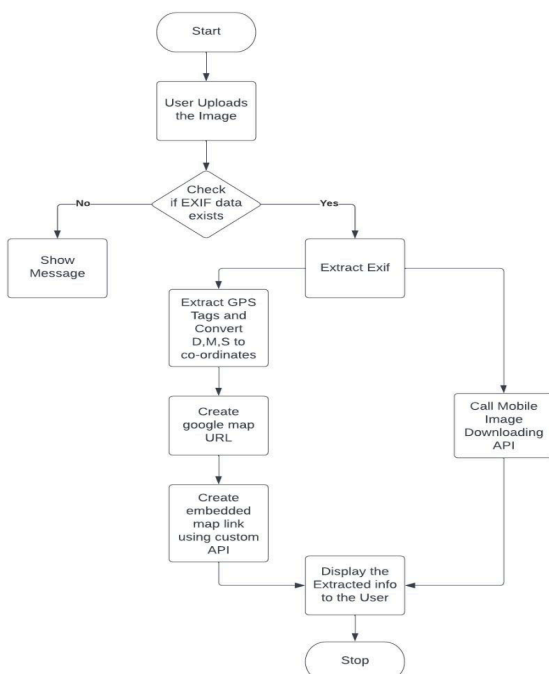


Fig.2 System Architecture

A. EXIF Extraction: When taking a photo with a digital device, supplementary data is saved as EXIF metadata. EXIF data includes camera type, f-stop, exposure duration, sensitivity, shooting date and time, and more. Python's exif module retrieves this data from images. This component simplifies accessing and controlling EXIF metadata. Use the open() function in Python to retrieve EXIF information from an image with the exif module. Acquire image data using read() method and transfer to exifread.process_file() for a dictionary of EXIF tags and values. After converting EXIF data to a dictionary, you can access specific tag values by their tag names. For example, the camera model can be retrieved with the Image Model tag, and the DateTimeOriginal tag provides the capture date and time. Enhance image processing, analysis, and organization by utilizing exif for valuable information extraction from image files.

B. GPS Tags Extraction from Exif Data: A GPS device adds geographic data to images using the EXIF file format. The data includes precise location and can be displayed in degrees, minutes, and seconds. Fetch GPS coordinates in degrees, minutes, and seconds with exifread module in Python. Access EXIF data easily. Use Python's open() function to access the image file. Then, use read() to access the visual data and transfer it to exifread.process_file(). This function generates a dictionary of EXIF tags and values. Use 'GPS GPS Latitude' and 'GPS GPS Longitude' to get GPS data from EXIF. Labels have degrees, minutes, and seconds. To convert to degrees, divide minutes by 60 and seconds by 3600, then add to degrees. Use decimal degrees for GPS coordinates and convert to degrees, minutes, and seconds format using Python. To convert decimal degrees to DMS format, multiply by 3600 and extract values using integer division and modulo operators.

C. DMS Conversion: To make GPS tags data from an image usable with Google Maps, it's important to convert the data from its original degrees, minutes, and seconds format. Google Maps utilizes the Decimal Degrees format to denote coordinates. One can use the formula given below in Python to convert GPS tags data from the format of degrees, minutes, and seconds to decimal degrees format.

$$\text{decimal degrees} = \text{degrees} + (\text{minutes} / 60) + (\text{seconds} / 3600)$$

To isolate the degrees, minutes, and seconds information within the GPS tags data, Python offers string manipulation techniques that can effectively split the tags into individual values. Following the availability of values for degrees, minutes, and seconds, the aforementioned equation can be utilized to transform them into decimal degrees. A fresh variable can be created to accumulate the decimal degrees values, or the current GPS tag data can be modified.

D. Google Maps Link Generator: It is possible to use programming in Python to convert the location's coordinates

into URLs that can be used in Google Maps. This transformation can create a link that, when clicked, will open Google Maps and center the location specified. Constructing a properly formatted string that includes the latitude and longitude values is necessary to convert the location coordinates into a Google Maps URL. The arrangement of characters in this particular sequence is as follows:
 "q=[latitude],[longitude]"
 Where [scope] and [longitude] are the decimal degrees values for the location's scope and longitude, individually.

Following, this string has to be added to the base Google Maps URL:
 "https://www.google.com/maps?q="

The complete URL would then be:
 "https://www.google.com/maps/search/?q=[latitude],[longitude]"

After generating the Google Maps link, it can either be viewed on a web browser or incorporated into a webpage or application through the Google Maps API, in order to exhibit the requested location.

E. Google Maps API: Using Selenium in Python, you can automate web scraping from websites with Google Maps. The first step is to install essential components. To use Selenium, use pip to install it and a web driver for your preferred browser, such as Chrome or Firefox. Once Selenium is installed, a web driver can launch a new browser for navigation to the webpage with integrated Google Maps. Selenium locates map component in HTML source and extracts element attributes via `get_attribute()` method. The Google Map's URL grants direct access to its data for customized applications. Automate website navigation and extract Google Map information seamlessly with Selenium for expedited data retrieval. Details may include location, marked points, directions, and other particulars. This is helpful for tasks requiring Google Maps like geocoding or cartography.

F. Base 64 Encoding: Base64 encodes binary information into a string format using ASCII characters. It is used to code and decode data, such as images, for internet transmission. Python's base64 module decodes Base64 strings to download images and has encoding and decoding functions. To download an image from a Base64 string, start by decoding it using `base64.B64 decode()`. This function takes an encoded string in Base64 and outputs binary data. Use Python's native file functions to record the output of decoded binary information in various image formats like .jpg, png, and .gif. Translating Base64 data in Python is useful for images obtained through web services. Decoded data can be easily stored and used in various programs.

G. Mobile Image Download API: With a Python program and Selenium, exploring and obtaining mobile model pictures on Google Images can be automated. Selenium Web Driver

launches and directs the browser to the Google Images website. Access the website, search for the mobile model, and automate loading additional images with Selenium. Identify scrollable element on website, use JS executor to reach bottom. Use Selenium to extract image URLs via `find_elements_by_xpath()`. To enhance search results, include extra keywords with the mobile model name when issuing a GET command to the URL to obtain pictures saved as a binary document. Refine search results by size, color, or file type.

VIII. RESULT

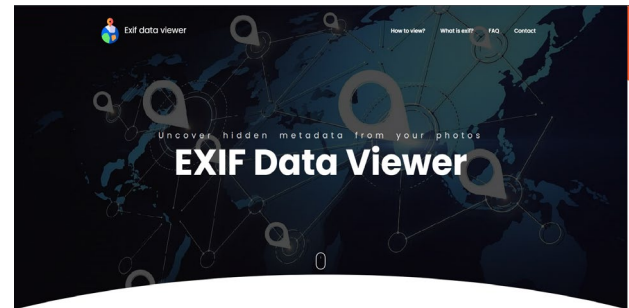


Fig.3 Home Page

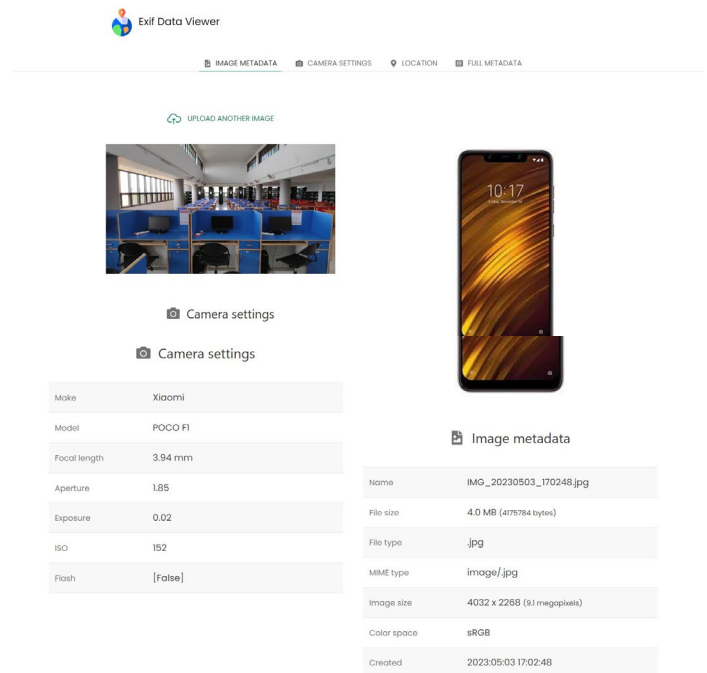


Fig.4 Device Information

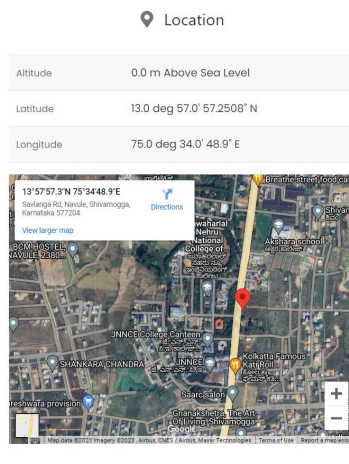


Fig.5 Location Information

IX. ACKNOWLEDGMENT

We are grateful to the Department of Computer Science and Engineering and our institution Jawaharlal Nehru New College of Engineering for providing us the facilities to carry out the research and for imparting us the knowledge with which we can do our best.

Finally, we also would like to thank the whole teaching and non-teaching staff of Computer Science and Engineering Department.

X. CONCLUSION

The utilization of EXIF metadata to geo-locate pictures has been an effective tool for various purposes. These may include conservation, disaster management, urban planning, crime investigation and other related endeavours. The new capability of gathering geographical information from digital photos has unleashed fresh opportunities for comprehending and controlling the society. As technology keeps advancing and becomes more accessible, one can anticipate an abundance of innovative applications to arise in the coming years. The use of EXIF metadata for geo-locating images brings forth numerous possibilities for revolutionizing and advancing sectors such as personalized travel suggestions and public safety and emergency response. Undoubtedly, it will continue to be a crucial aspect of research and advancement, with significant implications for numerous industries and domains.

XI. REFERENCES

- [1] Miroslav Ölvecký, Marián Host'ovecký (2021), "Digital image forensics using EXIF data of digital evidence", International Conference on Emerging eLearning Technologies and Applications (ICETA).
- [2] Hsiang-Cheh Huang, Wai-Chi Fang (2008), "Copyright Protection with EXIF Metadata and Error Control Codes", International Conference on Security Technology
- [3] Vanessa Teles da Mota, Catherine Pickering (2022), "Assessing the popularity of urban beaches using metadata from social media images as a rapid tool for coastal management". Vol 203.
- [4] Juzlinda Mohd Ghazali, Khan (2020), "Image Classification using EXIF Metadata", International Journal of Engineering Trends and Technology (IJETT).
- [5] D. P. Gangwar, Anju Pathania (2018), "Authentication of Digital Image using Exif Metadata and Decoding Properties", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol 3.
- [6] Sungjin Hong, Kyung-Hyune Rhee (2011), "An approach for the similar file detection with GPS information", First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering.
- [7] B Prasetyo, Alamsyah, M A, Subhan and N Baroroh (2020), "Automatic geotagging using GPS EXIF metadata of smartphone digital photos in tree planting location mapping", Journal of Physics: Conference Series, The 7th International Conference on Mathematics, Science, and Education 2020 (ICMSE) , Semarang, Indonesia, Vol 1918.
- [8] Ramesh Kumar P (2016), "Location Identification of the Individual based on Image Metadata", International Conference on Computational Modeling and Security.
- [9] Lionel Prat, Cheryl Baker (2015), "MapExif: an image scanning and mapping tool for investigators", International Journal of Digital Crime and Forensics .
- [10] Eswara Sai Prasad, Chunduru, Nagendar Rao Koppolu (2021), "Digital Crime Investigation, Unleashing the Secrets of Data by Attributing the Origination and Authorship". National Conference on Novel & Challenging Issues and Recent Innovations in Engineering and Information Sciences.
- [11] A L Sandoval Orozco, D M Arenas González, L J Garcíá Villalba (2014), "Analysis of Errors in EXIF Metadata on Mobile Devices", Springer.
- [12] <https://selenium-python.readthedocs.io/>
- [13] https://en.wikipedia.org/wiki/Web_scraping
- [14] <https://flask.palletsprojects.com/en/2.3.x/>