In [1]:

```
from PIL import Image
import requests
from io  import  BytesIO
```

In [2]:

```
#!/usr/bin/env python

from PIL import Image

def get_exif(filename):
    image = Image.open(filename)
    image.verify()
    return image._getexif()

exif = get_exif(r'C:\Users\rj100\OneDrive\Pictures\bird.jpg')
print(exif)
```

```
{36864: b'0220', 37377: 4.32, 37378: 1.85, 36867: '2016:11:10 19:33:22', 36868: '2016:11:10 19:33:22', 3
7379: -1.57, 37380: 0.0, 37381: 1.85, 37383: 3, 37385: 0, 37386: 4.3, 40961: 1, 40962: 5312, 41989: 28,
41990: 0, 256: 5312, 40963: 2988, 257: 2988, 271: 'samsung', 272: 'SM-G920F', 274: 1, 531: 1, 282: 72.0,
283: 72.0, 42016: 'A16LLIC08SM A16LLIL02GM\n\x00\x00\x00\x00\x00', 34850: 2, 34855: 640, 296: 2, 41986:
0, 40960: b'0100', 41987: 0, 305: 'G920FXXS4DPI4', 306: '2016:11:10 19:33:22', 34665: 226, 37500:
b'\x07\x00\x01\x00\x07\x00\x04\x00\x00\x000100\x02\x00\x04\x00\x01\x00\x00\x00\x00
\x01\x00\x0c\x00\x04\x00\x01\x00\x00\x00\x00\x00\x00\x00\x10\x00\x05\x00\x01\x00\x00\x00Z\x00\x00\x00@\x0
\x00\x01\x00\x00\x00\x00\x00\x00\x00P\x00\x04\x00\x01\x00\x00\x00\x01\x00\x00\x00\x00\x01\x03\x00\x01\x00
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00', 37510:
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00', 33434: 0.05,
33437: 1.9}
```

In [3]:

```
from PIL.ExifTags import TAGS

def get_labeled_exif(exif):
    labeled = {}
    for (key,val) in exif.items():
        labeled[TAGS.get(key)] = val

    return labeled

exif = get_exif(r'C:\Users\rj100\OneDrive\Pictures\bird.jpg')
labeled = get_labeled_exif(exif)
print(labeled)
```

```
{'ExifVersion': b'0220', 'ShutterSpeedValue': 4.32, 'ApertureValue': 1.85, 'DateTimeOriginal':
'2016:11:10 19:33:22', 'DateTimeDigitized': '2016:11:10 19:33:22', 'BrightnessValue': -1.57, 'ExposureBia
sValue': 0.0, 'MaxApertureValue': 1.85, 'MeteringMode': 3, 'Flash': 0, 'FocalLength': 4.3, 'ColorSpace':
1, 'ExifImageWidth': 5312, 'FocalLengthIn35mmFilm': 28, 'SceneCaptureType': 0, 'ImageWidth': 5312,
'ExifImageHeight': 2988, 'ImageLength': 2988, 'Make': 'samsung', 'Model': 'SM-G920F', 'Orientation': 1,
'YCbCrPositioning': 1, 'XResolution': 72.0, 'YResolution': 72.0, 'ImageUniqueID': 'A16LLIC08SM A16LLIL02G
M\n\x00\x00\x00\x00\x00', 'ExposureProgram': 2, 'ISOSpeedRatings': 640, 'ResolutionUnit': 2, 'ExposureMod
e': 0, 'FlashPixVersion': b'0100', 'WhiteBalance': 0, 'Software': 'G920FXXS4DPI4', 'DateTime':
'2016:11:10 19:33:22', 'ExifOffset': 226, 'MakerNote':
b'\x07\x00\x01\x00\x07\x00\x04\x00\x00\x000100\x02\x00\x04\x00\x01\x00\x00\x00\x00
\x01\x00\x0c\x00\x04\x00\x01\x00\x00\x00\x00\x00\x00\x00\x10\x00\x05\x00\x01\x00\x00\x00Z\x00\x00\x00@\x0
\x00\x01\x00\x00\x00\x00\x00\x00\x00P\x00\x04\x00\x01\x00\x00\x00\x01\x00\x00\x00\x00\x01\x03\x00\x01\x00
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00', 'UserComment':
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00', 'ExposureTime':
0.05, 'FNumber': 1.9}
```

In [4]:

```
from PIL.ExifTags import GPSTAGS

def get_geotagging(exif):
    if not exif:
        raise ValueError("No EXIF metadata found")

    geotagging = {}
    for (idx, tag) in TAGS.items():
        if tag == 'ExifVersion':
            if idx not in exif:
                raise ValueError("No EXIF geotagging found")
```

```
            for (key, val) in GPSTAGS.items():
                if key in exif[idx]:
                    geotagging[val] = exif[idx][key]

    return geotagging

exif = get_exif(r'C:\Users\rj100\OneDrive\Pictures\bird.jpg')
geotags = get_geotagging(exif)
print(geotags)
```

```
{}
```

In [ ]:

```
{

   'GPSLatitudeRef': 'N',
   'GPSLatitude': ((36, 1), (7, 1), (5263, 100)),
   'GPSLongitudeRef': 'W',
   'GPSLongitude': ((115, 1), (8, 1), (5789, 100)),
   'GPSTimeStamp': ((19, 1), (8, 1), (40, 1)),
   ...

}
```

If in geotags outcome was similarly like upper so we would be using downward code to calculate latitude & longitude.

In [ ]:

```
def get_decimal_from_dms(dms, ref):

    degrees = dms[0][0] / dms[0][1]
    minutes = dms[1][0] / dms[1][1] / 60.0
    seconds = dms[2][0] / dms[2][1] / 3600.0

    if ref in ['S', 'W']:
        degrees = -degrees
        minutes = -minutes
        seconds = -seconds

    return round(degrees + minutes + seconds, 5)

def get_coordinates(geotags):
    lat = get_decimal_from_dms(geotags['GPSLatitude'], geotags['GPSLatitudeRef'])

    lon = get_decimal_from_dms(geotags['GPSLongitude'], geotags['GPSLongitudeRef'])

    return (lat,lon)


exif = get_exif(r'C:\Users\rj100\OneDrive\Pictures\bird.jpg')
geotags = get_geotagging(exif)
print(get_coordinates(geotags))
```

For GeoReverse Coding

In [ ]:

```
import os
import requests

def get_location(geotags):
    coords = get_coordinates(geotags)

    uri = 'https://revgeocode.search.hereapi.com/v1/revgeocode'
    headers = {}
    params = {
```

```
        'apiKey': os.environ['API_KEY'],
        'at': "%s,%s" % coords,
        'lang': 'en-US',
        'limit': 1,
    }

    response = requests.get(uri, headers=headers, params=params)
    try:
        response.raise_for_status()
        return response.json()

    except requests.exceptions.HTTPError as e:
        print(str(e))
        return {}

exif = get_exif('image.jpg')
geotags = get_geotagging(exif)
location = get_location(geotags)

print(location['items'][0]['address']['label'])
```

CHecking

In [6]:

```
from PIL import Image
from PIL.ExifTags import TAGS, GPSTAGS

def get_exif_data(image):
    """Returns a dictionary from the exif data of an PIL Image item. Also converts the GPS Tags"""
    exif_data = {}
    info = image._getexif()
    if info:
        for tag, value in info.items():
            decoded = TAGS.get(tag, tag)
            if decoded == "GPSInfo":
                gps_data = {}
                for t in value:
                    sub_decoded = GPSTAGS.get(t, t)
                    gps_data[sub_decoded] = value[t]

                exif_data[decoded] = gps_data
            else:
                exif_data[decoded] = value

    return exif_data

def _get_if_exist(data, key):
    if key in data:
        return data[key]

    return None

def _convert_to_degress(value):
    """Helper function to convert the GPS coordinates stored in the EXIF to degress in float format"""
    d0 = value[0][0]
    d1 = value[0][1]
    d = float(d0) / float(d1)

    m0 = value[1][0]
    m1 = value[1][1]
    m = float(m0) / float(m1)

    s0 = value[2][0]
    s1 = value[2][1]
    s = float(s0) / float(s1)

    return d + (m / 60.0) + (s / 3600.0)

def get_lat_lon(exif_data):
    """Returns the latitude and longitude, if available, from the provided exif_data (obtained through g
    lat = None
    lon = None

    if "GPSInfo" in exif_data:
```

```
        gps_info = exif_data["GPSInfo"]

        gps_latitude = _get_if_exist(gps_info, "GPSLatitude")
        gps_latitude_ref = _get_if_exist(gps_info, 'GPSLatitudeRef')
        gps_longitude = _get_if_exist(gps_info, 'GPSLongitude')
        gps_longitude_ref = _get_if_exist(gps_info, 'GPSLongitudeRef')

        if gps_latitude and gps_latitude_ref and gps_longitude and gps_longitude_ref:
            lat = _convert_to_degress(gps_latitude)
            if gps_latitude_ref != "N":
                lat = 0 - lat

            lon = _convert_to_degress(gps_longitude)
            if gps_longitude_ref != "E":
                lon = 0 - lon

    return lat, lon


################
# Example ######
################
if __name__ == "__main__":
    image = Image.open(r'C:\Users\rj100\OneDrive\Pictures\bird.jpg')
    exif_data = get_exif_data(image)
    print(get_lat_lon(exif_data))
```

```
(None, None)
```

In [ ]: