Code Review

2024.03.28

1. **Libraries and Configuration Import:**

Imports necessary libraries: Gdal from Osgeo for reading and converting geospatial raster formats, Osr from Osgeo for spatial reference and transformation operations and Image from PIL (Python Imaging Library) for image processing;
Imports Pandas for data manipulation and saving the output in CSV format;
Configures GDAL to not use Python exceptions for error handling;
Removes the limit on the maximum number of pixels that can be loaded into a PIL image, preventing decompression bomb errors for large images.

Dependencies can be found at requirements.txt

2. **Opening the Raster Dataset:**

Opens a TIF file (Harmonized_DN_NTL_2021_simVIIRS.tif), which is a raster dataset;
Uses PIL to open the same file and converts it to an RGB image for pixel access.

3. **Setup for Geospatial Operations:**

Creates an empty DataFrame with columns for pixel coordinates, geographic coordinates (longitude and latitude) and brightness value;
Extracts the geotransform (gt) from the dataset, which contains coefficients for transforming pixel coordinates to geographic coordinates;
Prepares a spatial reference system (srs) from the dataset's projection information for coordinate transformation.

4. **Data Extraction Loop:**

Iterates over a subset of pixels in the raster dataset, specifically those in the ranges y = 0 to 4199 and x = 0 to 1405 for the first set, and then x = 23761 to the dataset's width for the second set. This subset selection suggests focusing on Russian regions;
For each pixel, calculates its geographic coordinates using the geotransform coefficients and the spatial reference system. This involves a coordinate transformation from the dataset's projection system to a geographic coordinate system (longitude and latitude);
Attempts to get the brightness value of the pixel from the RGB image. If the pixel is outside the bounds of the image (which could happen due to the subset selection), it assigns 'NaN' as the brightness value.

5. DataFrame Creation and CSV Output:

After processing, compiles the collected data into a pandas DataFrame. Each row in the DataFrame represents a pixel, with its x and y coordinates, calculated longitude and latitude, and brightness value;
Saves this DataFrame to a CSV file named output.csv.