

Supplementary Material

SegDesicNet: Lightweight Semantic Segmentation in Remote Sensing with Geo-Coordinate Embeddings for Domain Adaptation

A. Encoding Geographic Coordinates for Potsdam Dataset

The Potsdam dataset contains pixel height and width, rotation terms for rows and columns, and latitude and longitude of the top-left pixel for each tile of size 6000×6000 . Because the origin is typically top-left, the pixel height is negative, and the rotation term for this data is zero here, we calculated the latitude and longitude of the top-left pixel of each image split of size 256×256 by following algorithm 1.

Algorithm 1 Calculate Geo-Coordinates of Image patch from Metadata of Image Tile

Require: $tile_{xcoord}$: longitude of top left pixel of given Image Tile, $tile_{ycoord}$: latitude of top left pixel of given Image Tile, $pixel_height$, $pixel_width$

- 1: **for** each Image Patch **do**
 - 2: $i, j \leftarrow$ obtain pixel location related to Tile)
 - 3: $patch_longitude_x \leftarrow x_{coord} + j \times pixel_width$
 - 4: $patch_latitude_y \leftarrow y_{coord} + i \times pixel_height$
 - 5: **Output:** $\{ patch_longitude_x, patch_latitude_y \}$
 - 6: **end for**
-

In addition, the Potsdam dataset uses the EPSG:32633 coordinate reference system (CRS), which corresponds to the WGS 84 / UTM zone 33N projection. Hence continuing with the method same as FLAIR #1, followed the following step

- Coordinate Centering,

$$\begin{aligned} C'_{lon} &= C_{lon} - 500000.0 \text{ m} \\ C'_{lat} &= C_{lat} - 4649776.22 \text{ m} \end{aligned}$$

- Transforming coordinates to the EPSG:4326 from EPSG:32633.

$$\lambda, \phi = f_{32633 \rightarrow 4326}(C'_{lon}, C'_{lat})$$

and the rest of the steps remain the same¹. As per the implementation of algorithm 1, for Potsdam we have not taken the

¹Center coordinates value (500000.0, 4649776.22) is taken from epsg.io

geolocation of centroid of the patch as in FLAIR#1, rather we took the coordinate of let top pixel of the patch.