Landslide identification using deep learning-based change detection and the DeepESDL collaborative cloud platform

Master Thesis developed at the Politecnico di Milano by Julia Leonardi under the supervision of Prof. Maria Antonia Brovelli and co-supervision of Dr. Vasil Yordanov and Dr. Valerio Marsocci

Objectives of the project

- Investigate the possibilities of **deep learning-based change detection** workflows for **landslide identification**
  - *Create a **global landslide change detection training dataset** with optical **Sentinel-2 data***
  - *Apply different **change detection workflows** to compare their performances and explore the capabilities of **Deep Learning-based** workflows*

- Assess the **usefulness of the DeepESDL** cloud collaborative platform for **machine** and **deep learning** applied to **earth observation research** use cases.

What tools were critical to the project

- **The Sentinel Hub xcube data store cloud API**. Retrieving the relevant patches of Sentinel-2 imagery by subsetting using the intuitive Python API request syntax was a great advantage when creating the dataset.
- **The possibility of storage of the retrieved data**. Thanks to which there was no need of downloading the heavy files to the PC, speeding up and easing the research process.
- **Jupyter Lab environment with dependencies**. Providing an intuitive data manipulation platform with all the important tools in place.

The outputs of the project

- **Training dataset** created within the DeepESDL platform and **using the SH xcube data store cloud API** (code snippet on the next slide). The dataset is **published** on **Zenodo**: https://zenodo.org/records/10800338
- **Master Thesis** with the result of the **assessment of three change detection workflows** (Differential Thresholding, Bitemporal Image Transformer Change Detection [BIT-CD], and Change Detection based on image Reconstruction Loss [CDRL]). The text of the thesis is publicly available: https://www.politesi.polimi.it/handle/10589/218042
- Code repository with the Notebooks developed on GitHub: https://github.com/leonardijulia/MSc

```python
landslide_list = []
ident = 0
x = 0.01

for index, row in landslide_inventory.iterrows():
    event_date = row.event_date
    bbox = [row.longitude - x, row.latitude - x, row.longitude + x, row.latitude + x]
    poly = box(bbox[0], bbox[1], bbox[2], bbox[3])
    time_range_pre = (row.pre2, row.pre2)
    time_range_post = (row.post1, row.post1)

    dataset_pre = store.open_data('S2L2A',
                          variable_names=['B01', 'B02', 'B03', 'B04', 'B05', 'B06', 'B07', 'B08', 'B8A', 'B09', 'B11', 'B12', 'C
                          bbox=bbox,
                          spatial_res=0.00001,
                          time_range=time_range_pre,
                          time_period='1D',
                          tile_size= [1024, 1024])

    dataset_post = store.open_data('S2L2A',
                          variable_names=['B01', 'B02', 'B03', 'B04', 'B05', 'B06', 'B07', 'B08', 'B8A',
                          bbox=bbox,
                          spatial_res=0.00001,
                          time_range=time_range_post,
                          time_period='1D',
                          tile_size= [1024, 1024])

    landslide_list.append(
        {
            'T1':dataset_pre,
            'T2':dataset_post,
        }
    )
    ident = ident + 1
```

[3]:
```python
landslide_inventory
```

[3]:

	event_date	accuracy	category	latitude	longitude	source	pre1	pre2	post1
0	2022-11-26	exact	mudslide	40.737080	13.901270	EMS	2022-04-05	2022-04-12	2023-03-28
1	2017-04-22	exact	landslide	3.055990	101.781120	GLC	2017-01-14	2017-02-03	2018-02-13
2	2017-05-20	exact	landslide	35.864314	-121.430499	GLC	2017-04-20	2017-05-17	2017-05-27
3	2017-05-26	exact	landslide	6.514224	80.169071	GLC	2017-01-31	2017-03-02	2017-07-15
4	2017-06-19	exact	mudslide	51.976862	-122.279271	GLC	2017-05-29	2017-06-05	2017-06-25
...	...	...	...	...	...	...	...	...	...
169	2023-02-06	exact	landslide	36.750054	36.297924	Tukey	2022-09-07	2022-09-17	2023-05-25
170	2023-02-06	exact	landslide	36.842000	36.323500	Tukey	2022-09-07	2022-09-17	2023-05-25
171	2023-02-06	exact	landslide	37.004370	36.590910	Tukey	2022-09-07	2022-09-17	2023-02-24
172	2017-05-11	exact	landslide	5.851690	-75.921240	UT	2017-02-16	2017-03-28	2017-06-16
173	2018-02-15	exact	landslide	5.534380	-75.225990	UT	2017-05-24	2017-07-13	2018-08-07

174 rows × 9 columns

[26]:
```python
landslide_df = pd.DataFrame(landslide_list)
```

[27]:
```python
landslide_df
```

[27]:

	T1	T2
0	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
1	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
2	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
3	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
4	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
...	...	...
169	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
170	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
171	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
172	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...
173	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...	[B01, B02, B03, B04, B05, B06, B07, B08, B09, ...

174 rows × 2 columns

Additional Comments

- An issue that I found during the work using the DeepESDL platform was that there was **limited flexibility in terms of customizing the users' dependencies**. I believe that giving more freedom and **simplifying the process of installing and updating libraries** would benefit the research process.
- A more **in depth documentation of examples using the xcube** toolkit would also be a big advantage. I believe that xcube could be a state of the art framework for earth observation data used in machine and deep learning research, however still not much information and examples are available to propagate this technology.