AI4Copernicus tools and methods for bridging AI and EO

A half-day tutorial at Big Data from Space (BiDS) 2023

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 101016798.

Table of Contents

1 Introduction

Artificial Intelligence (AI) represents a collection of tools and methodologies that have the potential of transforming virtually all aspects of human activity. Earth observation (EO) data, including satellite and in-situ, are essential for a number of applications, covering high-impact domains as diverse as security, agriculture and health. The H2020 AI4Copernicus project delivers a technological framework for developers and businesses to combine AI-infused tools and EO data and services in order to create high impact applications. This is further facilitated by providing a bridge between DIAS platforms and the European AI-on-Demand platform.

This tutorial will present the main technological assets AI4Copernicus brings to the table as well as its methodology and tools for linking the DIAS and AI-on-Demand platforms, through appropriately selected use-cases during a hands-on session. Indicatively, technological assets AI4Copernicus contribute to the community include the following: Sentinel-1 and Sentinel-2 pre-processing chains, Deep network for pixel-level classification of S2 patches, Probabilistic downscaling of CAMS air quality model data, and many others. In addition, AI4Copernicus contributed semantics-based tools for visualisation and discovery of complex EO data. More information can be found in the AI4Copernicus Technical Documentation.

The choice of the use cases for the hands-on applications comes from the AI4Copernicus project's rich collaborations with 3rd parties who specialise in EO applications in the fields of Security, Health, Agriculture, Climate and others. This tutorial targets EO and AI specialists and will require little IT technical background. The hands-on session takes place on the CREODIAS infrastructure and involves the application of AI techniques on available datasets.

2 On-site Instructors & Contributors

On-site instructors

- Antonis Troumpoukis, NCSR-Demokritos
- Despina-Athanasia Pantazi, National and Kapodistrian University of Athens
- Omar Barrilero, European Union Satellite Centre
- Giulio Weikmann, University of Trento
- Iraklis Klampanos, NCSR-Demokritos

Contributors

- Jacek Tokarski, CloudFerro
- Lorenzo Bruzzone, University of Trento
- Michele Lazzarini, European Union Satellite Centre
- Mohanad Albughdadi, ECMWF
- Vasileios Baousis, ECMWF
- George Stamoulis, National and Kapodistrian University of Athens
- Manolis Koubarakis, National and Kapodistrian University of Athens

More information about the tutorial can be found at https://ai4copernicus-bids2023.github.io/

3 Exercise 1: Assessment of Impact of Natural Hazards

3.1 Summary

In this exercise, it will be estimated the area affected by a flooding that took place in Australia in March 2021. For that, in the first part, they will be explored the capabilities of Sentinel-1 and Sentinel-2 imagery for the delineation of the flood and later, in the second part, they will be exploited the linked data tools in order to make the produced data more valuable.

3.2 Background

"Extreme rainfall on the east coast of Australia beginning on 18 March 2021 led to widespread flooding in New South Wales, affecting regions from the North Coast to the Sydney metropolitan area in the south. Suburbs of Sydney experienced the worst flooding in 60 years, and the events were described by NSW Premier Gladys Berejiklian as "one in 100-year" flooding. Far-southeast communities in Queensland were also affected by flooding and heavy rainfall, though to a lesser extent than those in New South Wales.

The Australian government declared many parts of the east coast a natural disaster zone after the flooding rains forced 18,000 people to evacuate, in addition to over 1,000 flood rescues. Described as a "prolonged event" by Berejiklian and "dangerous and threatening" by the Bureau of Meteorology, the floods extended from the coastal towns of Taree and Kempsey on Thursday, 18 March, to the populated suburbs of western Sydney by Friday and Saturday.

The floods occurred less than 18 months after Australia was affected by the Black Summer bushfires, impacting many towns still recovering from that disaster." Source: https://en.wikipedia.org/wiki/2021_eastern_Australia_floods

3.3 Area of Interest

Figure 1: Area of interest considered in the tutorial for the first exercise.

In this tutorial we are going to focus on an area on the North-West of Sydney. The polygon, represented in WKT format and EPSG:4326 is POLYGON((150.6266 -33.49, 150.952 -33.49, 150.952 -33.795, 150.6266 -33.795, 150.6266 -33.49)).

3.4 Theoretical basis

3.4.1 Change detection with Sentinel-1

In Sentinel-1 imagery (SAR) the pixel values represent the energy reflected back to the radar and it depends on the amount of energy the SAR sensor transmitted, the properties and the shape of the object and the angle from which the object is viewed. Regions of calm water (e.g. flooding) have low pixel values when compared with rough surfaces or man-made structures.

Exploiting this difference in the backscatter that flooding presents, it can be used the Amplitude Change Detection (ACD) to detect flood areas. A typical representation of ACD is:

R: Backscatter day 1

G: Backscatter day 2

B: Backscatter day 2

With this representation, some significant changes are highlighted in the ACD composite. For example:

- New buildings: Before the construction low backscatter, after the construction high backscatter -> CYAN
- Flooding: decrease of backscatter because of very low backscatter of water bodies -> RED

It is also possible to exploit the coherence to perform the Multi-temporal Coherence (MTC) for change detection with Sentinel-1. The coherence is the amplitude of the complex correlation coefficient between two images. A low coherence represents changes between two acquisitions. The MTC composite is generated as follows:

R: Backscatter day 1

G: Backscatter day 2

B: Coherence

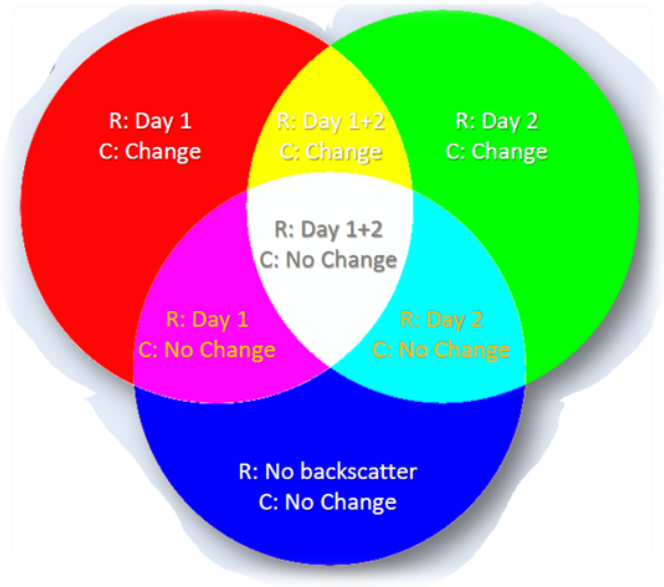


Figure 2: Color interpretation of MTC product

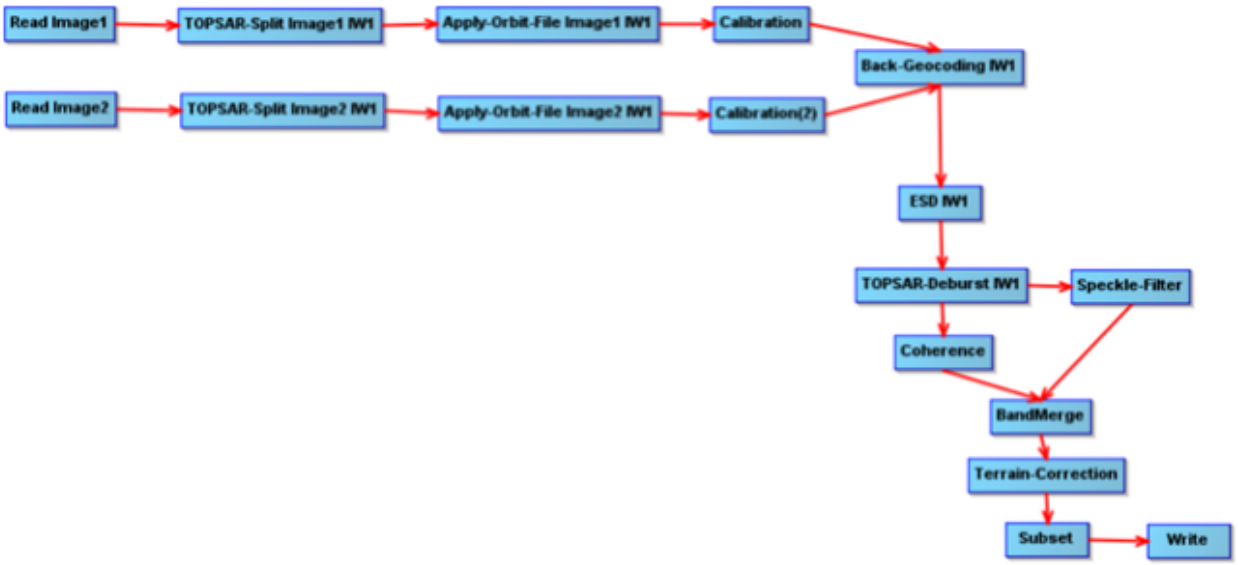


Figure 3: Processing pipeline to generate MTC with SNAP.

3.4.2 Change detection with Sentinel-2

Sentinel-2 images contain information of 13 different spectral bands and their differences between two acquisitions can be used to identify relevant changes.

The Change Vector Analysis (CVA) algorithm computes the difference vector of the selected bands. This vector has an amplitude, that can be used to determine if there is a relevant change or not (if

the value is higher than a certain threshold), and a direction, that can be used to classify the change.

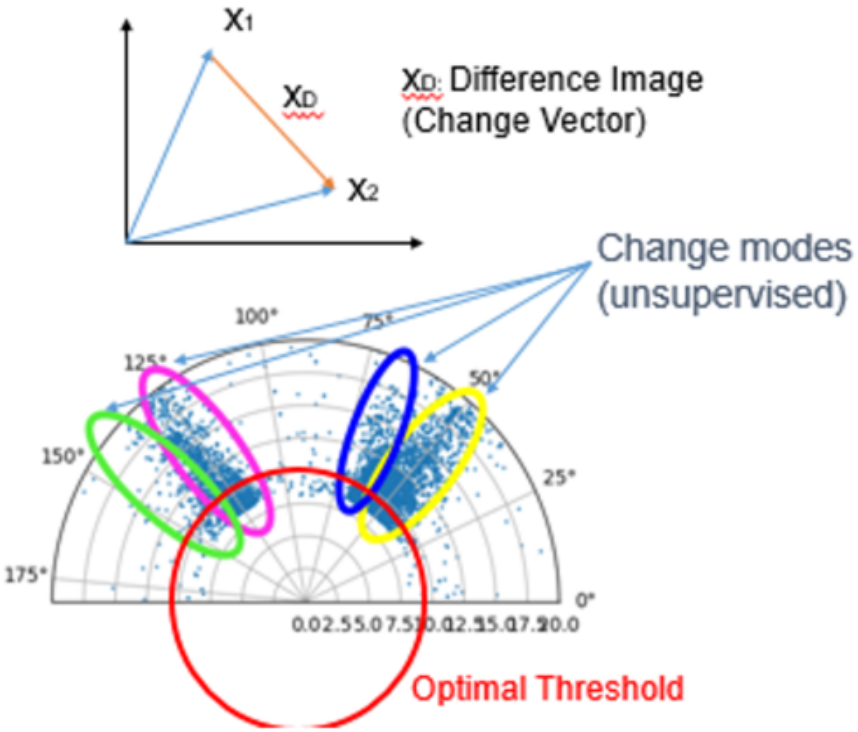


Figure 4. Change Vector analysis example

3.5 Exercise setup and VM configuration

3.5.1 First part: generation of flood mask

For the **first part** of this exercise, we will compute the changes between images acquired before and after the flooding to generate a flood mask. For that, we will connect to the VM and will execute some of the bootstrapping services that are available for AI4Copernicus project.

Step-by-step setup:

1. Connect to your VM

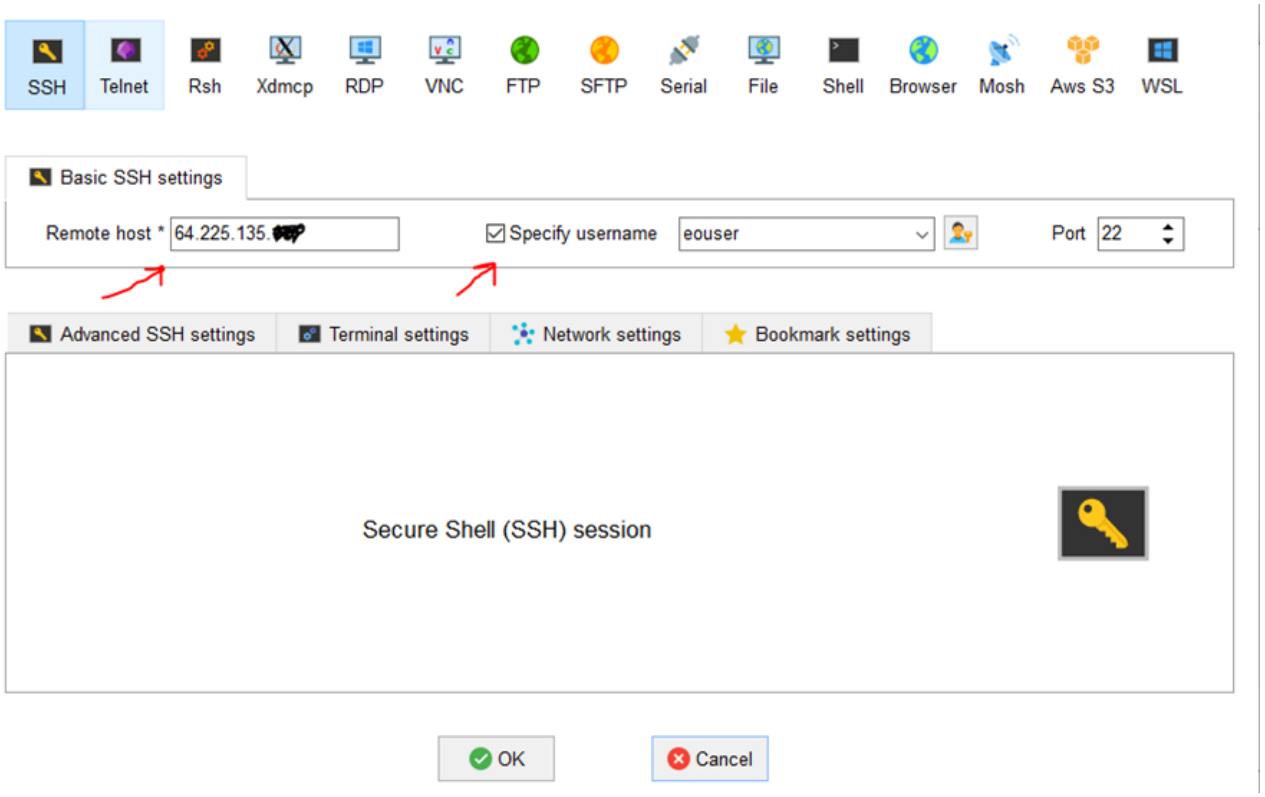
We will use MobaXterm, that is available for Windows, but this can be also done using any other software or command line that allows ssh connection (and ideally support X11-Forwarding for visualising some results, although this is not mandatory for the execution of the exercise)

1.a. Download MobaXterm (Portable edition)

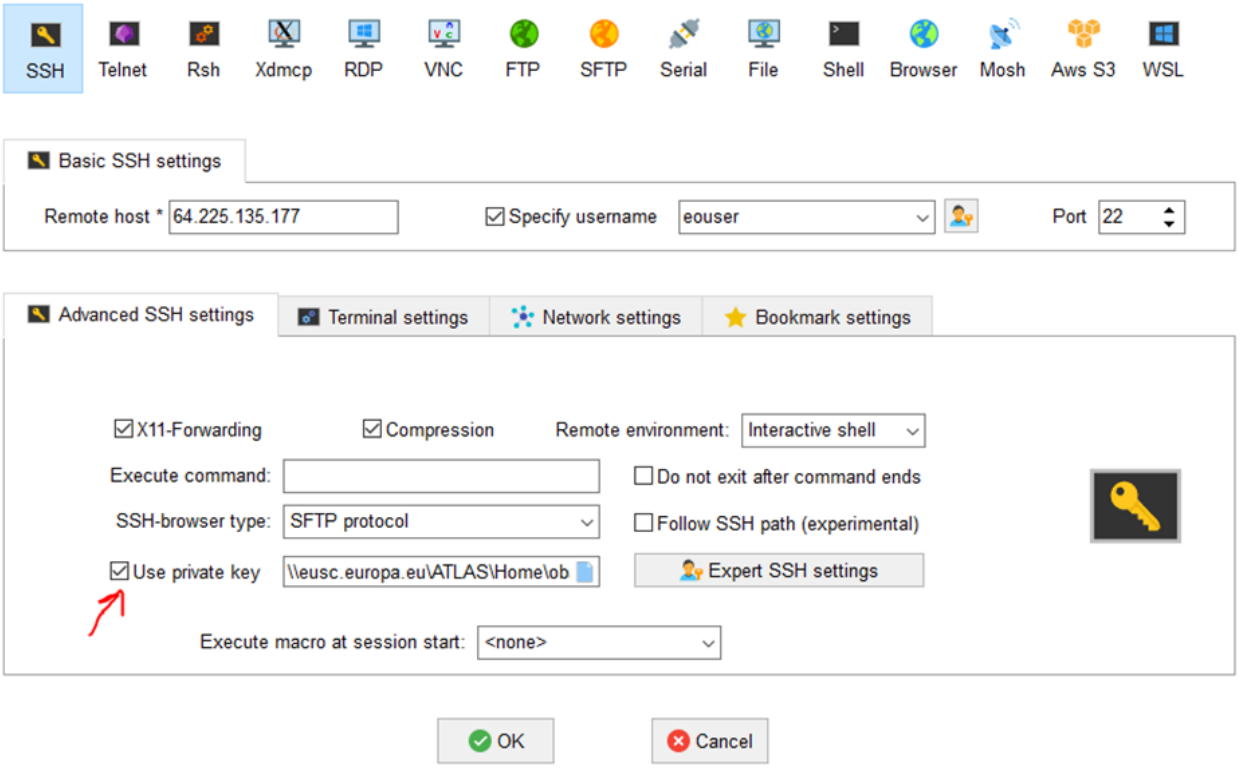
1.b. Run MobaXterm_Personal_23.3.exe

1.c. Create new session:

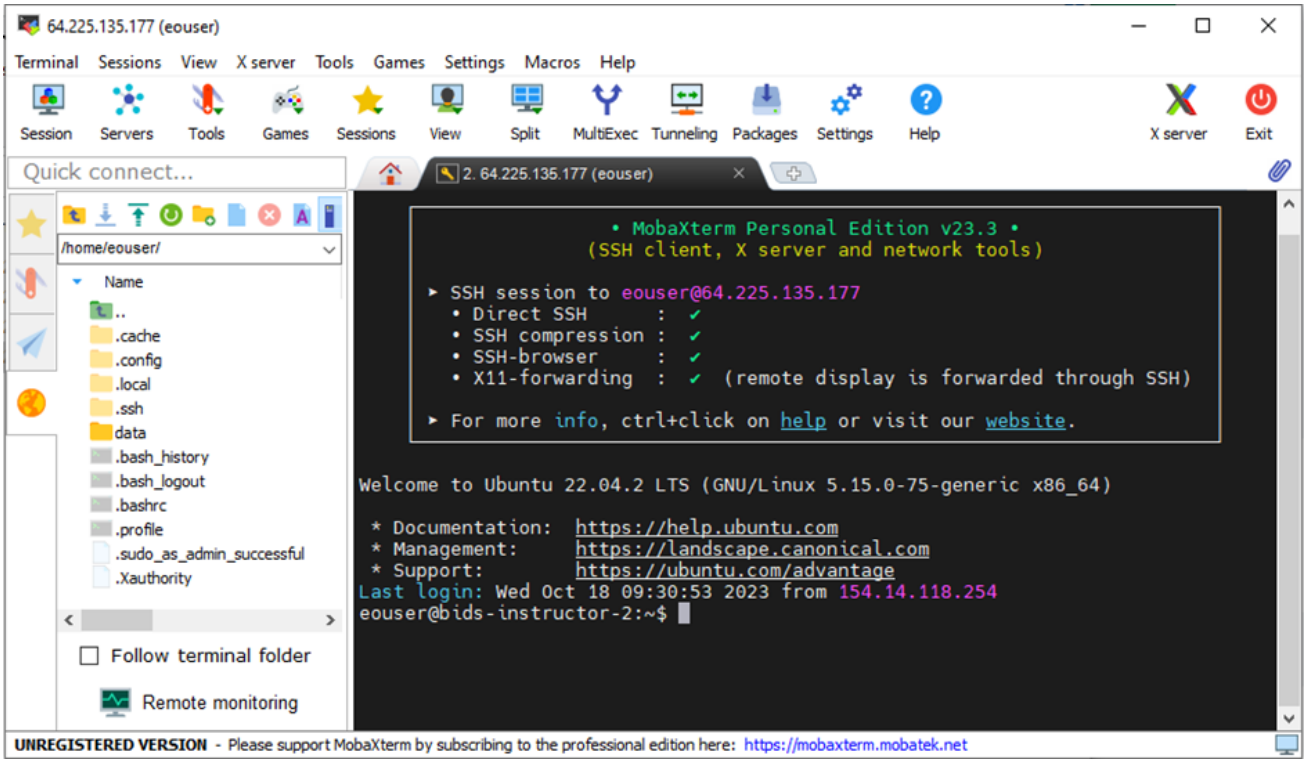
a. Select SSH and define Remote host, Specify username (eouser)



b. Define advanced settings indicating the path to the private key provided



c. Double-click on the session to start it



2. Check that inputs are available

Check files in /bids_tutorial. You should have:

S2A_MSIL2A_20210315T000241_N0214_R030_T56HKH_20210315T020346

S2A_MSIL2A_20210325T000241_N0214_R030_T56HKH_20210325T022649

S1A_IW_SLC__1SDV_20210312T191528_20210312T191555_036969_045983_7048

S1A_IW_SLC__1SDV_20210324T191529_20210324T191555_037144_045F93_9AB1

3. Run docker container with security services

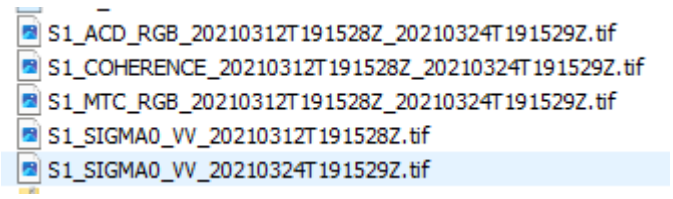
The security services, that include the Sentinel-1 and Sentinel-2 change detection pipelines are available as python scripts in the docker image that is already available in the VM. We will execute the bash command in it to have access to all the pipelines. Note that we are mounting the volume with the inputs as a volume in the container.

```
sudo docker run -it -v /bids_tutorial:/output
harborai4c.cloudferro.com/ai4copernicuswp5/security_services:1.2.
0 bash
```

4. Run the S1-CD algorithm

```
S1-CD -i1
/output/S1A_IW_SLC__1SDV_20210312T191528_20210312T191555_036969_0
45983_7048.zip -i2
/output/S1A_IW_SLC__1SDV_20210324T191529_20210324T191555_037144_0
45F93_9AB1.zip -r 20 -p "POLYGON((150.6266 -33.49, 150.952
-33.49, 150.952 -33.795, 150.6266 -33.795, 150.6266 -33.49))"
-outdir /output/
```

This command will generate the following results:



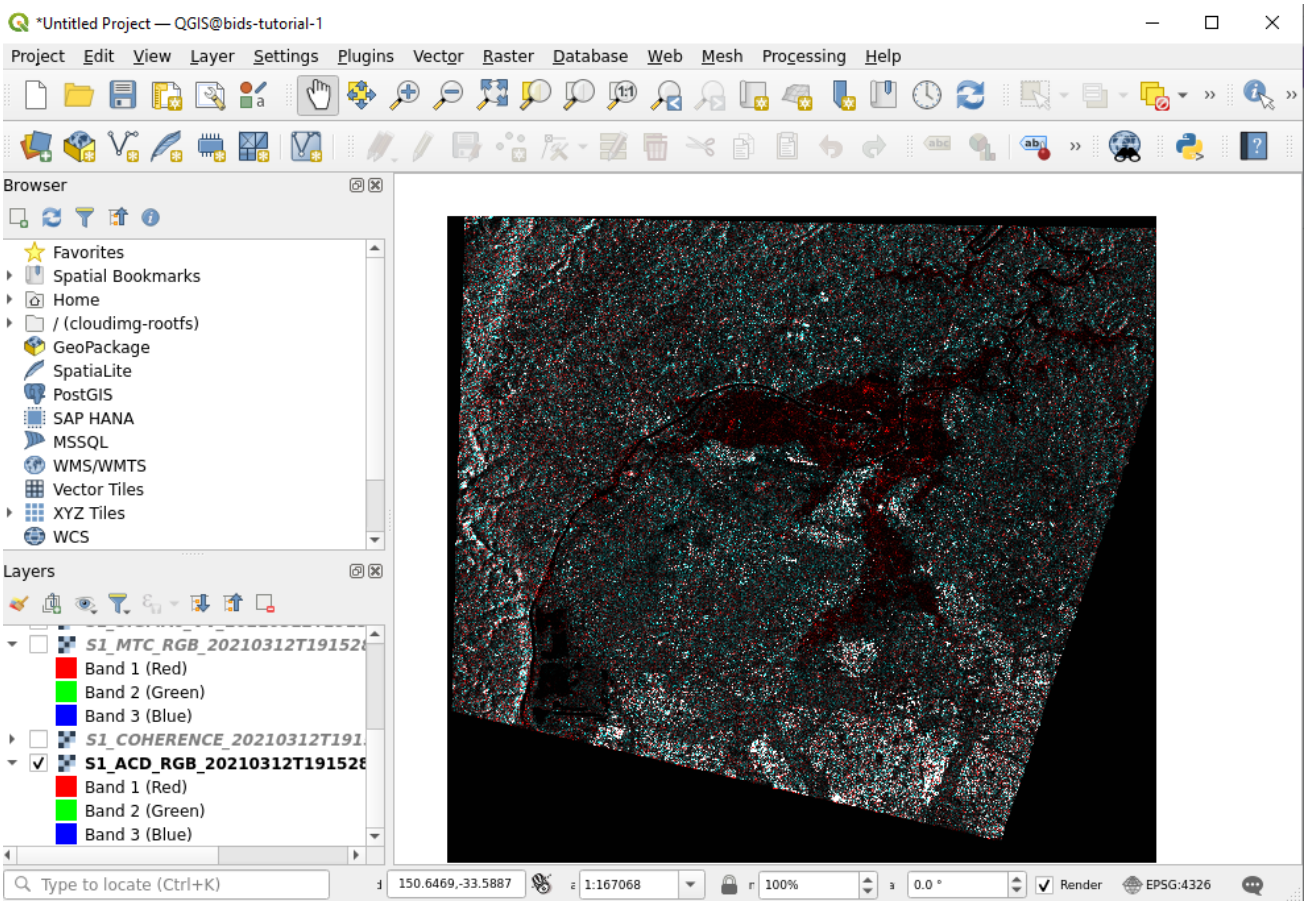
where:

- Coherence: GeoTiff image that represents the amplitude of correlation between the images. The pixel type is float32.
- S1_SIGMA0: 2 GeoTiff products (one for each of the input images) with one float32 band representing the calibrated backscatter.

- ACD: a RGB composite of the backscatter of the input images.
  - GeoTiff with three bands. R: backscatter of image 1; G : backscatter of image 2; B: backscatter of image 2
  - Pixel type is Byte, where byte value is computed by data conversion of float values using a linear interpolation taking as min and max values the percentiles 2.5 and 7.5.
- MTC (Multi-Temporal Coherence): a RGB composite of the backscatters and the coherence
  - GeoTiff with three bands. R: backscatter of image 1; G : backscatter of image 2; B: coherence
  - Pixel type is Byte, where byte value is computed by data conversion of float values using a linear interpolation taking as min and max values the percentiles 2.5 and 7.5 for backscatter band. For the coherence band the linear conversion is (0,1)->(0,255)

5. Check S1-CD results in qgis

Open QGIS (with the command "qgis") (from VM directly, not docker container) and open results. (Menu-> Layer->Add Layer… -> Add Raster layer…). Results are available in /bids_tutorial folder.

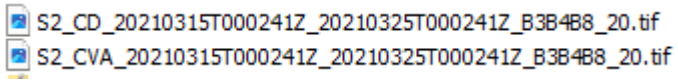


6. Run the S2-CD algorithm

We will run the Sentinel-2 Change Detection algorithm using the bands B3, B4 and B8 in the area of interest.

```
S2-CD -i1
/output/S2A_MSIL2A_20210315T000241_N0214_R030_T56HKH_20210315T020
346.zip -i2
/output/S2A_MSIL2A_20210325T000241_N0214_R030_T56HKH_20210325T022
649.zip -b B3,B4,B8 -r 20 -p "POLYGON((150.6266 -33.49, 150.952
-33.49, 150.952 -33.795, 150.6266 -33.795, 150.6266 -33.49))"
-outdir /output/
```

This command will generate the following outputs:

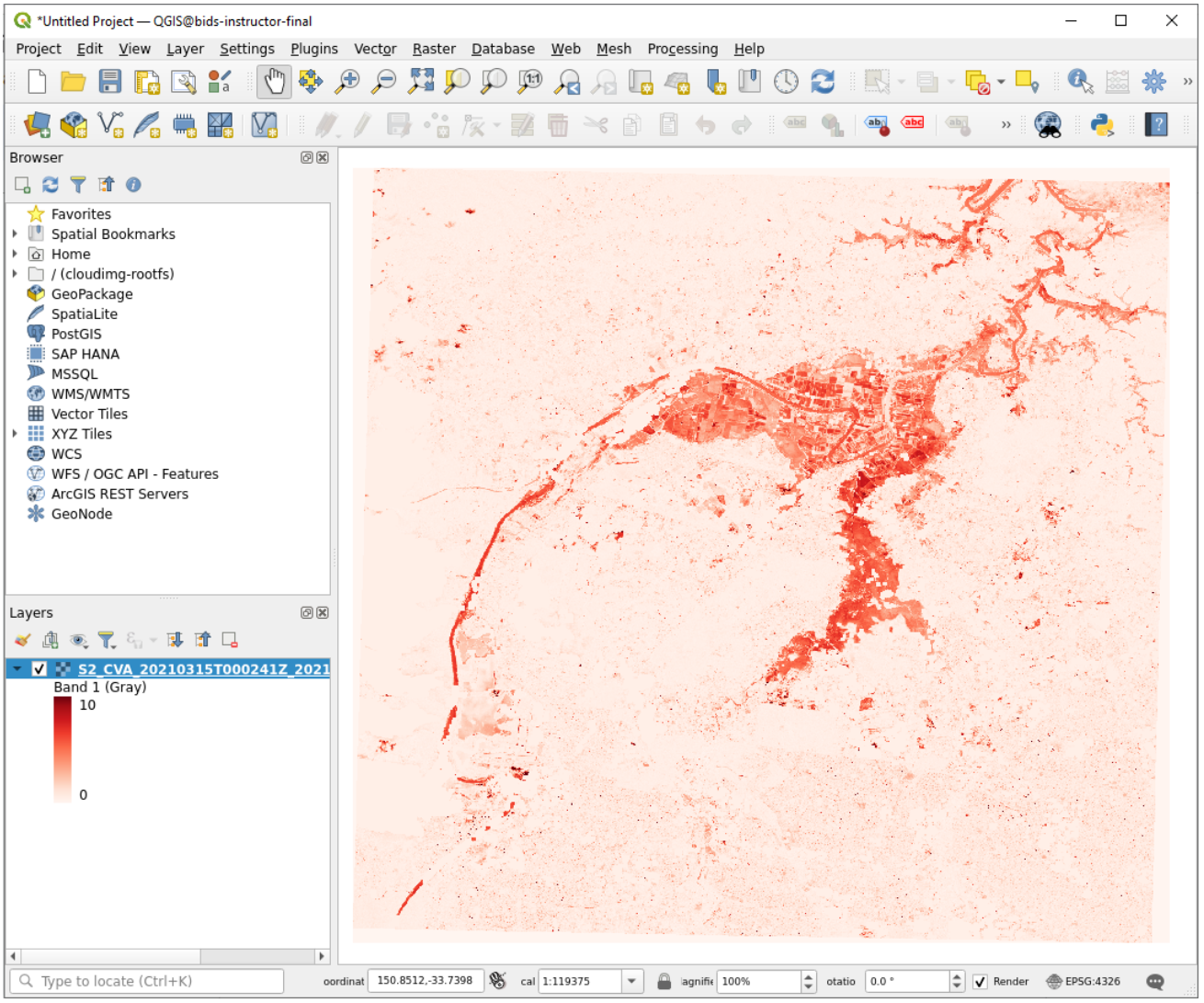


where:

- CVA: GeoTiff image with two bands. The first band is the amplitude of the change and the second band is the angle with respect to the reference vector.
- S2-CD: GeoTiff image with one band with pixel type Byte. It represents the classes of the detected changes.

7. Check the S2-CD results

Open QGIS (with the command “qgis”) (from VM directly, not docker container) and open results.



8. <u>Post process the results to create a vector mask</u>

Now, we are going to transform the S2-CD results to generate a flood mask in shapefile format using some gdal tools:

- gdal_sieve: to remove raster polygons smaller than a provided threshold size

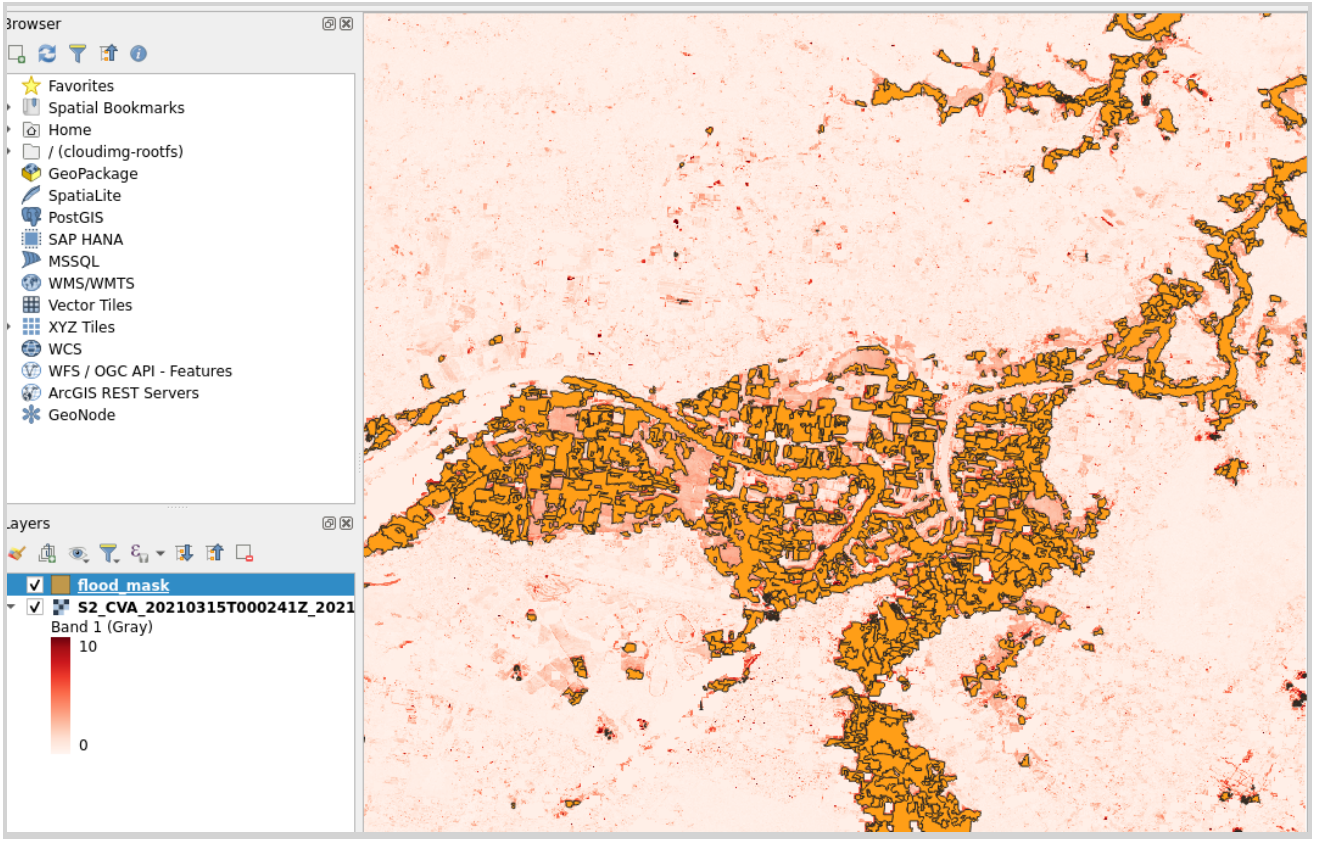
```
python3 /usr/lib/python3/dist-packages/osgeo_utils/gdal_sieve.py
-st 10
/output/S2_CVA_20210315T000241Z_20210325T000241Z_B3B4B8_20.tif
/output/sieve.tif
```

- gdal_calc: to apply a basic threshold to classify as flooding or not (when amplitude of difference is higher than certain value)

```
python3 /usr/bin/gdal_calc.py -A /output/sieve.tif
--calc="(A>=4)*A" --NoDataValue=0 --outfile
/output/sieve_calc.tif
```

- gdal_polygonize: to vectorize the raster to generate the final floodins mask in shapefile format

```
python3 /usr/bin/gdal_polygonize.py /output/sieve_calc.tif
/output/flood_mask.shp flood_mask floodLevel
```



3.5.2 Second part: exploitation of linked data tools

For the **second part** of this exercise, we will use the linked data tools in order to make the produced data more valuable.

I. Transformation of the produced data of Part 1 into linked data using GeoTriples

II. Storing of the linked data in the spatiotemporal RDF store Strabon

III. Example queries using the produced data and external data

IV. Visualisation of the results using Sextant

Step-by-step setup:

- Connect to your VM

1. Open terminal and connect via ssh to your VM (skip this step if you are already connected):

```
ssh -i /path/to/your/private-key-file eouser@remote-server-ip
```

2. Check, using ls, that files are in `/exercise1/part2`

You should have the folders:

```
FloodMaskLevels docker
```

3. Run the docker container.

```
sudo docker run -name bids-container -p 9999:8080 -v .:/inout bids
```

This script will also forward the port 8080 of the VM. In this way, you will be able to connect from your personal laptop/your pc. In another terminal window on your personal laptop/PC, connect to the VM and run the following command:

```
sudo docker exec -it bids-container /bin/bash
```

4. During this step we will transform the produced file of part 1, which is the shapefile included in the folder FloodMaskLevels, to linked data, using the tool GeoTriples.

Firstly, we need to create the mapping file, using the shapefile and the following code:

```
./geotriples-core-1.1.6-SNAPSHOT/bin/geotriples-all
generate_mapping -o /inout/levels_mapping_file.ttl -b
http://ai.di.uoa.gr/flood/ /inout/FloodMaskLevels/flood_levels.shp
```

Then, using the mapping file levels_mapping_file.ttl and the shapefile, we will create the linked data of the shapefile:

```
./geotriples-core-1.1.6-SNAPSHOT/bin/geotriples-all dump_rdf -o
inout/floodlevels_sm_v1.nt -b http://ai.di.uoa.gr/flood -sh
/inout/FloodMaskLevels/flood_levels.shp
/inout/levels_mapping_file.ttl
```

5. During this step, we will store in a Strabon endpoint our 2 datasets. The first dataset is the floodlevels_sm.nt, which was created during the previous step, and the GADM dataset, which includes information about the administrative divisions of our area of interest.

```
cd strabon/runtime

./strabon-cmd -db "endpoint" store
../../../inout/docker/floodlevels_sm.nt
```

```
./strabon-cmd -db "endpoint" store
../../../inout/docker/AUS_level2.nt

cd ~/..

./usr/local/bin/conf.sh
```

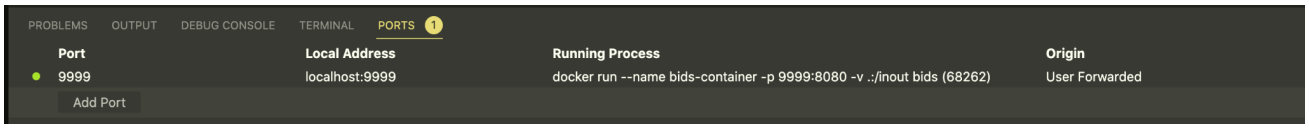
5. In order to connect to our local machine, we need to open two ports in the VM. Firstly, we install the ufw package:

```
sudo apt-get install ufw
```

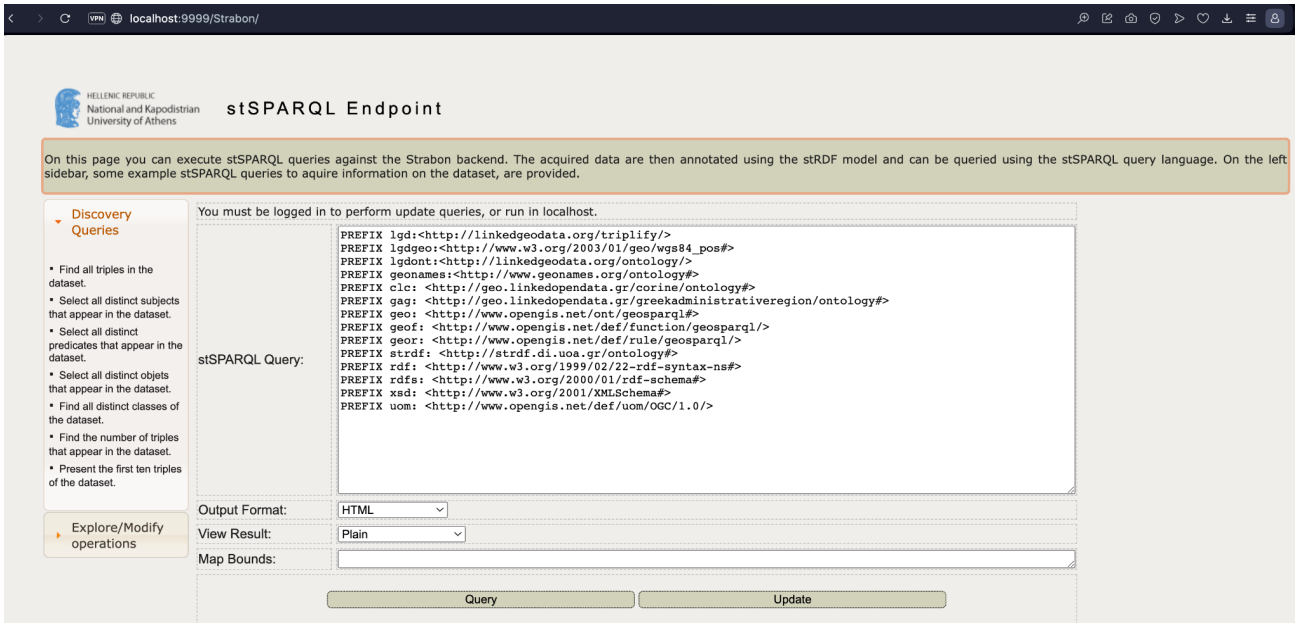
Then we need to open the ports 8080 and 9999 using the following commands:

```
sudo ufw allow 8080
sudo ufw allow 9999
```

In our local machine, we launch Visual Studio Code and we connect to the VM. Using the SSH extension, we forward traffic from a port on your local machine to a port on a remote server. We will select the port 9999.



6. We open a browser and we go to the url http://localhost:9999/Strabon/:



During this step, we will pose some queries using the datasets we stored in the Strabon endpoint.

- **Query 1**: Retrieve the amount of areas that have each flooding level.

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX flo: <http://ai.di.uoa.gr/flood/ontology#>
PREFIX gadmr: <http://www.app-lab.eu/gadm/>
PREFIX gadmo: <http://www.app-lab.eu/gadm/ontology/>

SELECT ?floodLevel (count(?x) as ?amountoflareas) WHERE{

?x flo:has_floodLevel ?floodLevel.
?x geo:hasGeometry ?xgeo.
?xgeo geo:asWKT ?geo.

} GROUP BY ?floodLevel
ORDER BY DESC(?floodLevel)
```

For the second query, we have two subqueries, which will then project to the map using the tool Sextant.

- **Query 2a**: Retrieve all the administrative divisions of level 3 of Australia, which intersect our flooding area of interest.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX flo: <http://ai.di.uoa.gr/flood/ontology#>
PREFIX gadmr: <http://www.app-lab.eu/gadm/>
PREFIX gadmo: <http://www.app-lab.eu/gadm/ontology/>

SELECT DISTINCT ?name ?wgeo WHERE{

?x flo:has_floodLevel ?floodLevel.
?x geo:hasGeometry ?xgeo.
?xgeo geo:asWKT ?geo.

?admP rdf:type gadmr:AdministrativeUnit.
?admP gadmo:hasName ?name.
?admP gadmo:hasNationalLevel "3rdOrder"^^<http://www.w3.org/2001/XMLSchema#string>.
?admP geo:hasGeometry ?admgeo.
?admgeo geo:asWKT ?wgeo

FILTER(geof:sfIntersects(?geo, ?wgeo))
}
```

The results of this query include the following 9 entries:

name	wgeo
"Baulkham Hills - Central"@en	"POLYGON ((151.026931762695 -33.7327537536621, 151.023986816406 ...
"Baulkham Hills - North"@en	"POLYGON ((151.006332397461 -33.5973930358886, 151.006332397461 ...
"Blacktown - South-East"@en	"POLYGON ((150.865661621094 -33.7412910461425, 150.866241455078 ...
"Blacktown - North"@en	"POLYGON ((150.865661621094 -33.7412910461425, 150.864974975586...
"Blacktown - South-West"@en	"POLYGON ((150.811096191406 -33.788215637207, 150.810943603516 ...
"Blue Mountains"@en	"POLYGON ((150.416107177734 -33.8085212707519, 150.415985107422 ...
"Hawkesbury"@en	"POLYGON ((150.508407592773 -33.416919708252, 150.509140014648 ...
"Penrith - East"@en	"POLYGON ((150.704010009766 -33.65128326416, 150.704162597656 -33.6510772705078 ...
"Penrith - West"@en	"POLYGON ((150.704010009766 -33.65128326416, 150.703918457031 -33.6515922546387...

- **Query 2b**: Retrieve 50 flooded areas that are included in our area of interest, and print their flooding level.

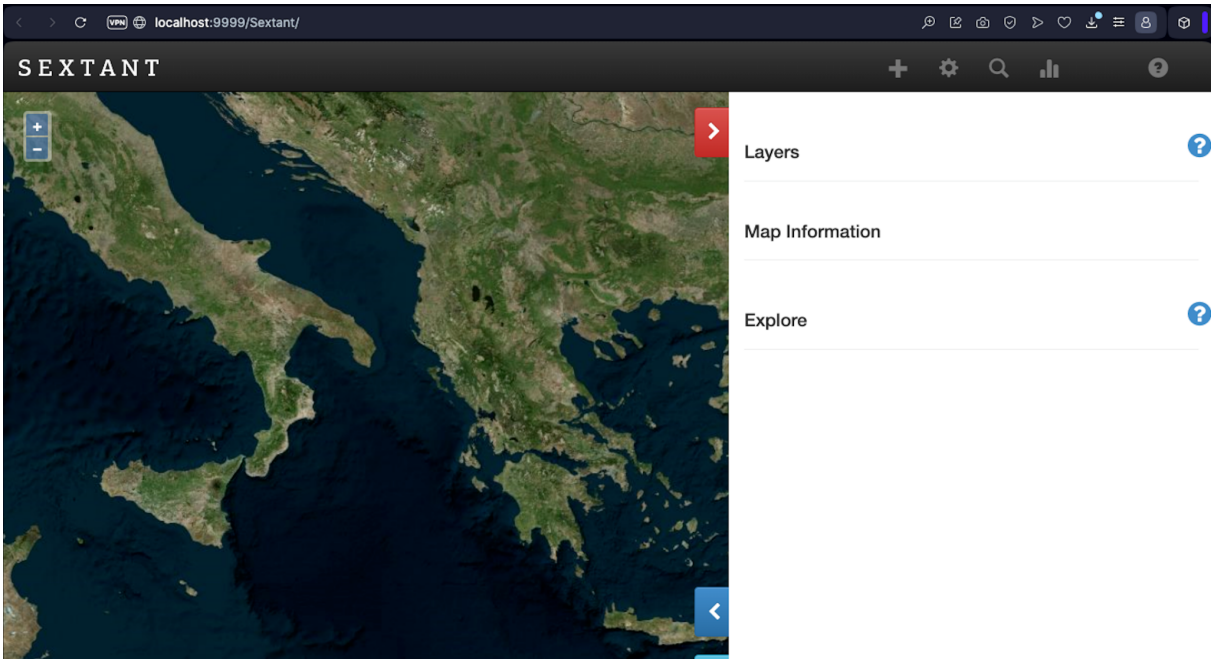
```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX flo: <http://ai.di.uoa.gr/flood/ontology#>

SELECT ?floodLevel ?geo WHERE{

?x flo:has_floodLevel ?floodLevel.
?x geo:hasGeometry ?xgeo.
?xgeo geo:asWKT ?geo.

}LIMIT 50
```

The following are the first 10 results of the query:

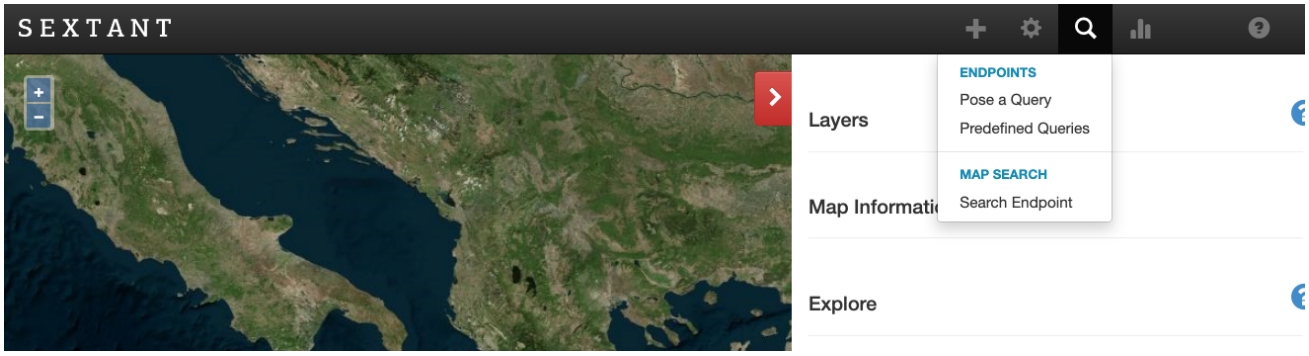
floodLevel	geo
"7"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326>

	MULTIPOLYGON (((150.82204547307123 …
"4"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.8218280852877…
"4"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.8852648726747…
"4"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.885370253615 …
"20"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.62850563929095 …
"16"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.628503181909 …
"5"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.89686533426 …
"8"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.8529289924099 …
"8"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.85378957594668 …
"8"^^<http://www.w3.org/2001/XMLSchema#integer>	"<http://www.opengis.net/def/crs/EPSG/0/4326> MULTIPOLYGON (((150.8545403632589 …

Now, we open a browser and we go to the url http://localhost:9999/Sextant/:

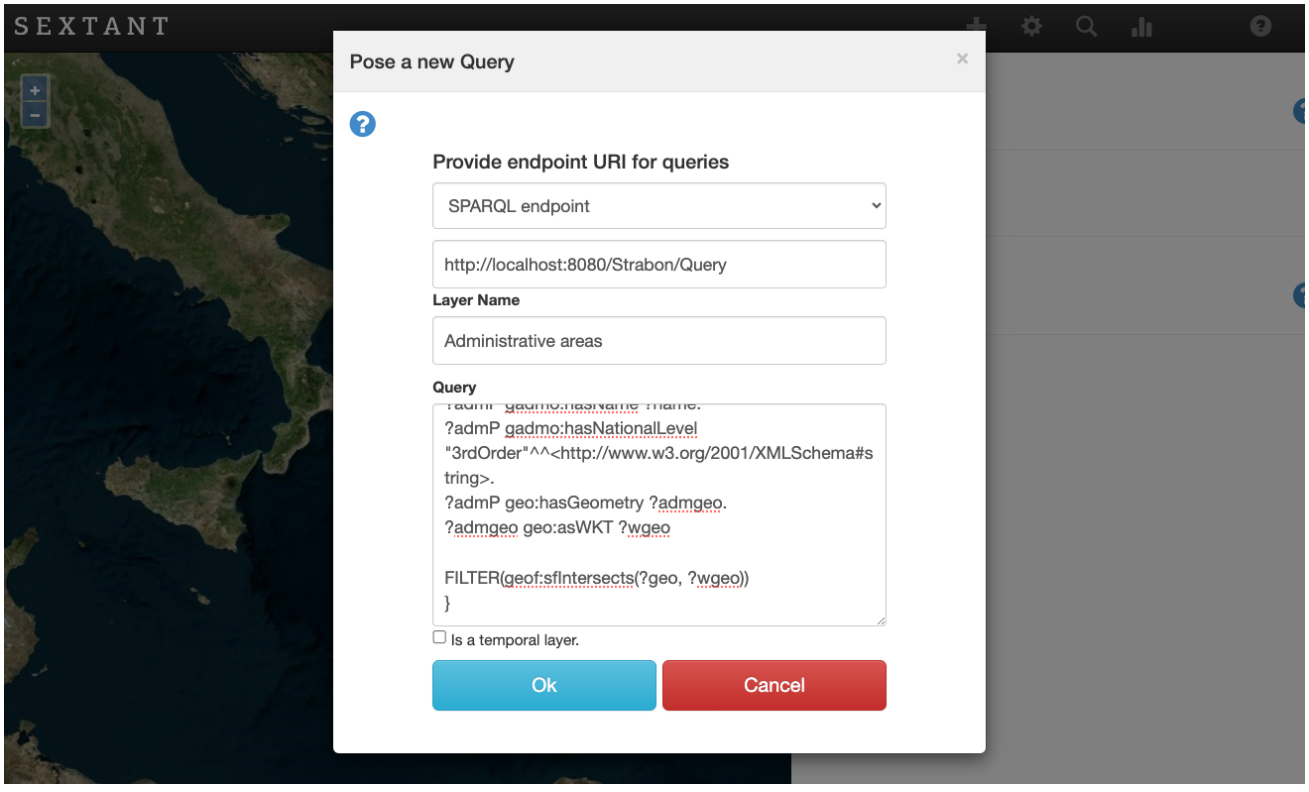


We will create 2 layers for the subqueries 2a and 2b. Firstly, we select the button to pose a query.

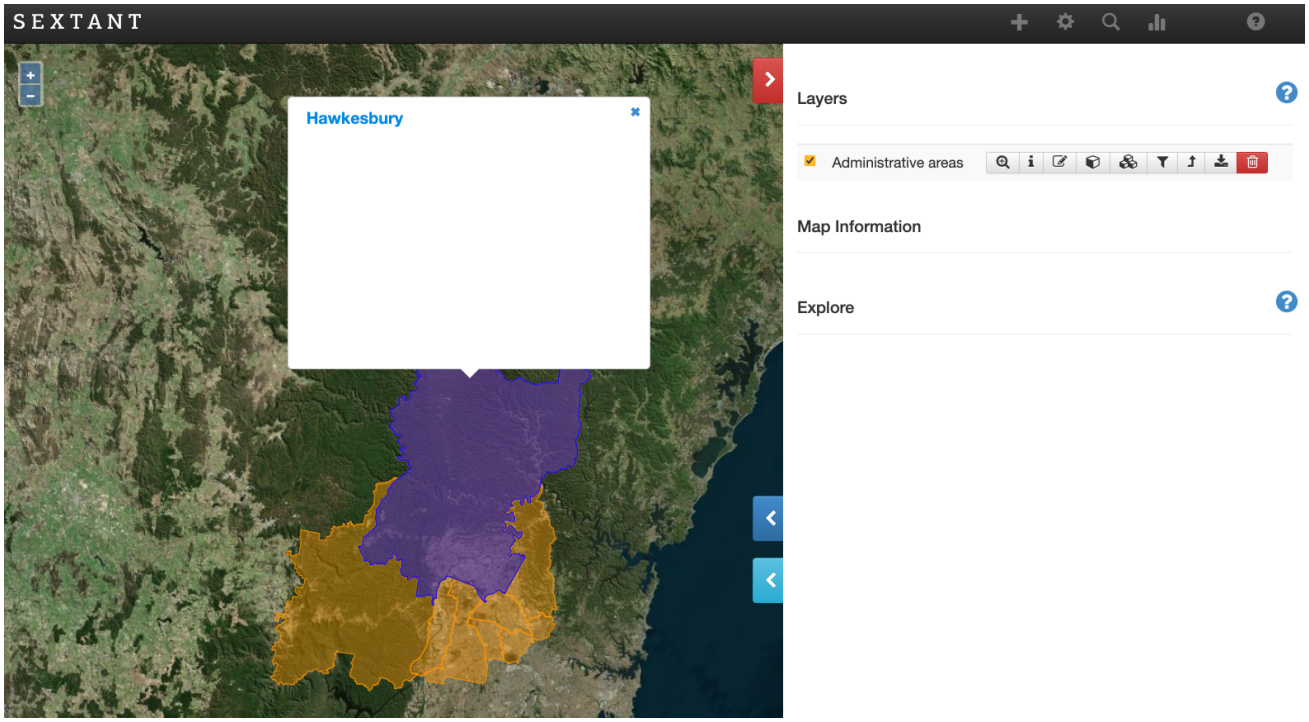


Then we pose query 2a and the other fields as shown below:

URI: http://localhost:8080/Strabon/Query
Layer Name: Administrative areas
Query: Query 2a

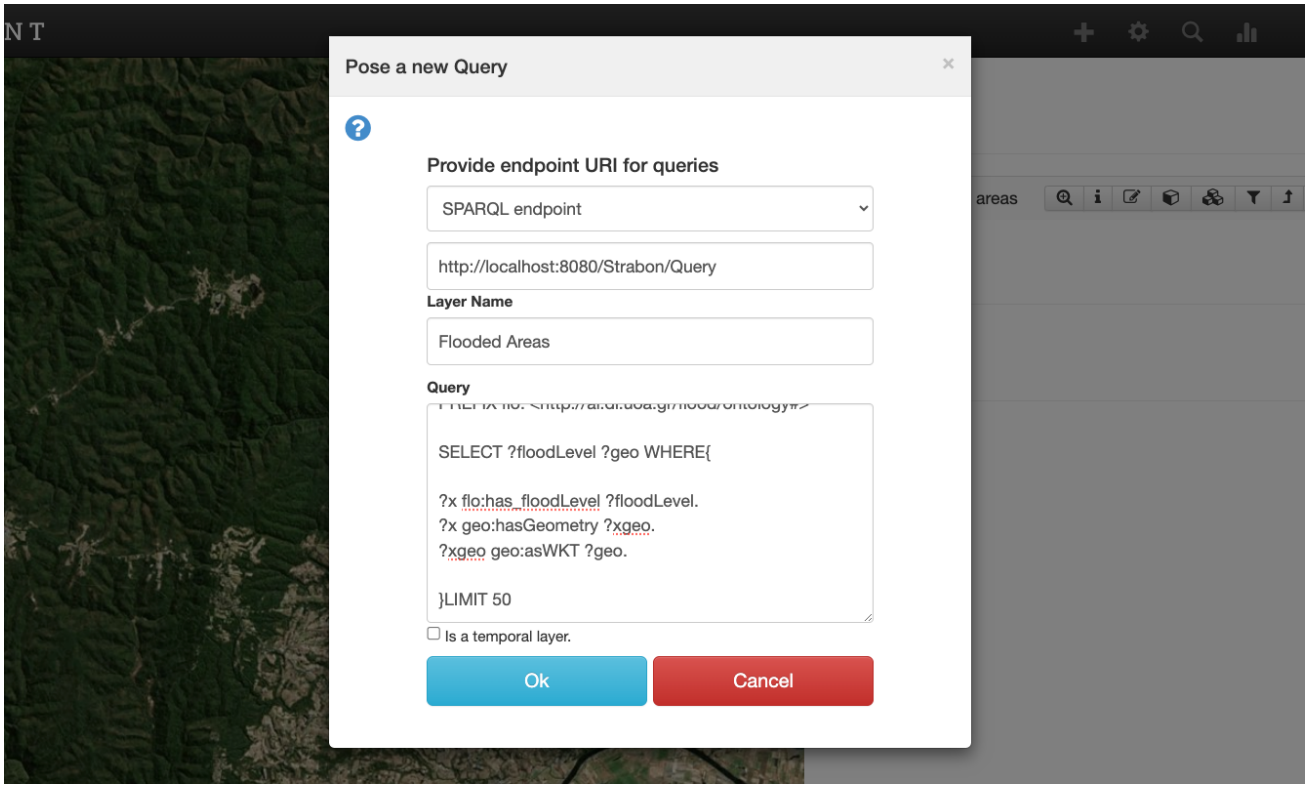


By selecting OK, we will see the created layer, and the 9 administrative areas shown on the map.

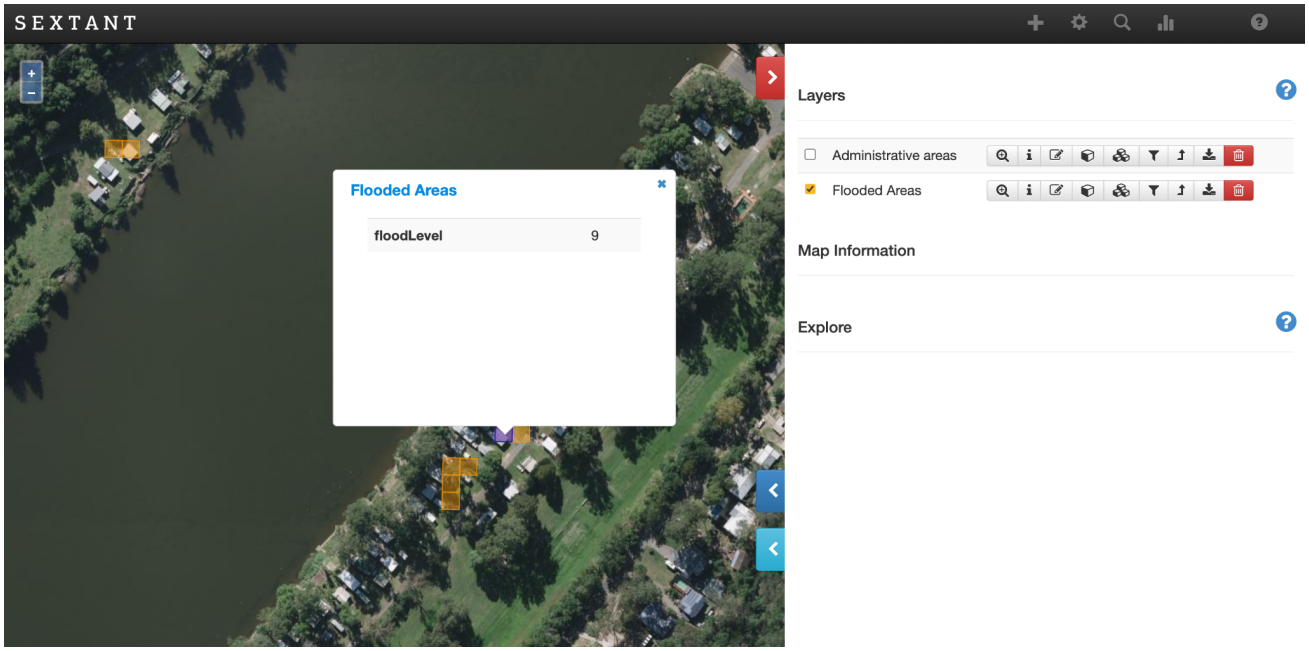


Then we create the second layer by posing query 2b and the other fields as shown below:

URI: http://localhost:8080/Strabon/Query
Layer Name: Flooded areas
Query: Query 2b



By selecting OK, we will see the created layer, and the 50 flooded areas shown on the map. The following screenshot includes some of these areas if we zoom in.



4 Exercise 2: Crop type classification

4.1 Summary

In this exercise, we will explore an example of a crop type mapping pipeline, which covers the steps from the pre-processing of Sentinel-2 satellite images to the final classification and production of detailed crop type maps. Crop type mapping is a crucial aspect of agricultural monitoring, land management, and precision agriculture, and this exercise will guide you through the entire process.

4.2 Background

Crop type mapping is a crucial aspect of agricultural monitoring and land management. This task involves the identification and classification of several types of crops grown in a specific area or region. This information provides valuable information for a variety of stakeholders, including farmers, agricultural organisations, government agencies, and researchers.

Crop type mapping is essential for implementing precision agriculture practices, allowing farmers to tailor their specific farming practices to the specific needs of each crop through irrigation, fertilisation, and pest control. Moreover, it allows the monitoring and management of agricultural production, ensuring a stable food supply. From the land manager's point of view, understanding the distribution of different crop types is crucial for making decisions on land use, zoning, and environmental conservation. Additionally, in case of natural disasters, crop type maps can be used to assess the impact on agriculture and plain disaster response efforts. Finally, market analysis and forecasting rely on such maps to predict crop yields and market prices, allowing stakeholders to make better informed decisions regarding trade and distribution.

Satellite imagery, such as data acquired from the Sentinel-2 satellites, is a primary source for crop type mapping. These images allow the analysis of crops based on their spectral signature and the phenological evolution of the crop during the agronomical year. Multispectral and hyperspectral data can be used to distinguish between different crops based on their unique spectral signatures. Deep neural networks allow the extraction of complex spatial and spectral relationships that are often challenging to extract using traditional image processing techniques. As a result, they play a vital role in enabling precision agriculture and land management. Nevertheless, high-quality labelled training data are required to effectively train a neural network from scratch.

Such applications, however, come with several challenges. Different crop types exhibit similar spectral signatures, especially during certain growth stages, making it difficult to distinguish between them based on single satellite images. Moreover, crop rotation is a common agricultural practice, which adds to the challenge of producing up-to-date crop type maps and determining the period of transition. Mapping the temporal variability is essential since the crop appearance changes drastically throughout the growing season. In addition, the cloud coverage leads to missing or incomplete data for a particular location, thus affecting the mapping accuracy.

4.3 Area of Interest

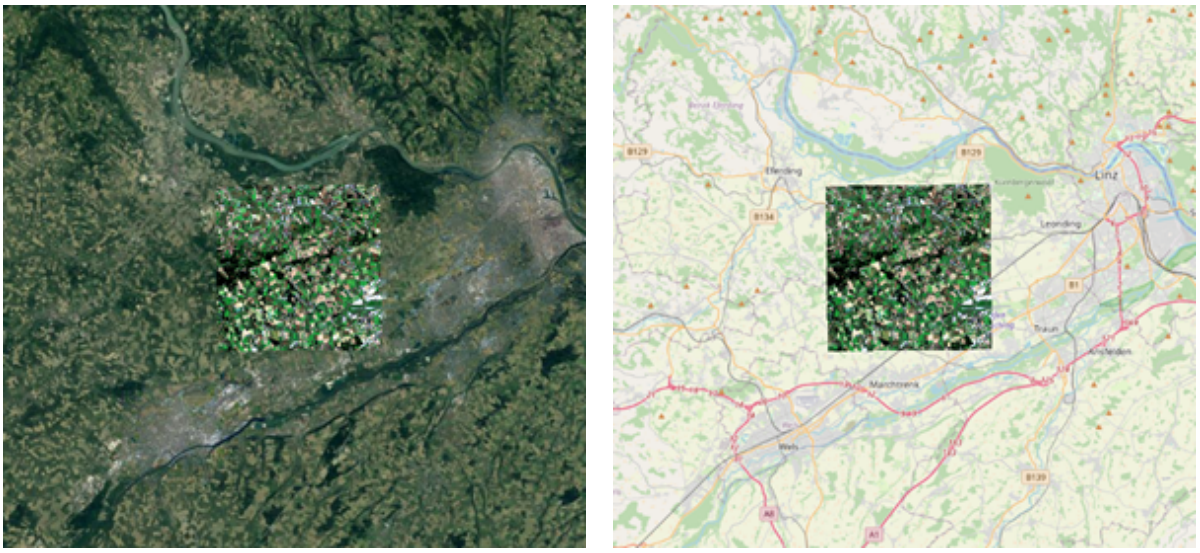


Figure 2: Area of interest considered in the tutorial for the second exercise.

In this tutorial we are going to focus on a sub-portion of tile 33UVP (MGRS) in the north of Austria, close to the city of Linz. The selection of this area was driven by several key factors, since it includes a wide range of crops and is a mainly cultivated area, allowing us to focus only on the crop type mapping task without considering the presence of other distinct classes.

4.4 Theoretical basis

Remote Sensing is the science of obtaining information about targets from a distance, without being in direct physical contact with it. This is typically achieved using sensors aboard platforms, typically satellites, aircrafts, or drones.

For this tutorial, we are going to use the Sentinel-2 constellation, which is a pair of Earth-observing satellites launched by the European Space Agency (ESA) as part of the Copernicus program. The twin satellites are in the same sun-synchronous orbit, phased at 180° to each other. Their wide swath width and high revisit time make them a perfect candidate for the crop type mapping task we are showcasing. These satellites are equipped with multispectral sensors that can capture high-resolution optical images of the Earth's surface. In particular, Sentinel-2 images show 13 different spectral bands in the visible, near-infrared, and shortwave-infrared regions. The resolution of the final product is 10m for the high-resolution bands, while it also includes 20m and 60m resolution bands. Moreover, the data acquired by the constellation is completely freely accessible to the public, making it an important resource for various applications.

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture, designed to address the vanishing gradient problem typical of RNNs. These architectures are widely used in various tasks, such as natural language processing, speech recognition, and time series analysis.

They are mostly suitable for tasks that involve sequences of data since they are able to capture long-term dependencies. Differently from RNNs, LSTMs consist of multiple gates that control the flow of information through the network. These gates let retaining or forgetting the information from previous time steps. An internal cell state is calculated based on the current input and the previous hidden state, which is updated with the information coming from the forget gate and the input gate. The output gate determines what part of the cell state will be used to produce the output for the current time step. The final hidden state is finally used for the prediction (or passed to subsequent LSTM cells).

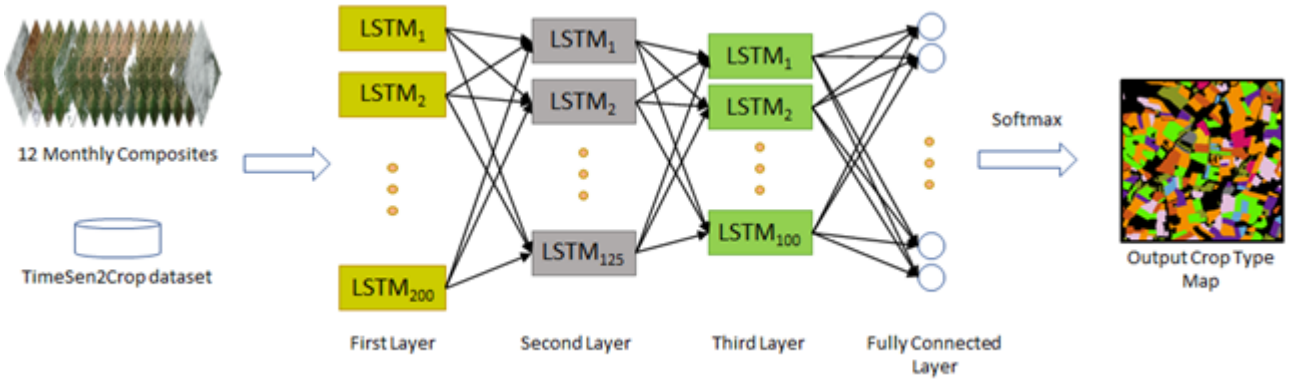


Figure 2: LSTM trained on TimeSen2Crop used in the inference step of the tutorial.

While deep neural networks are capable of learning complex patterns and representations from the input data, such complexity requires enough data available to generalise well. Large training sets help the model to generalise to a wide variety of crop types, growth stages, and environmental conditions. In addition to its other benefits, large training data can help reduce overfitting, learn the diversity of crop types and environmental conditions, address imbalanced class distributions, enhance model robustness, and mitigate the effects of data quality problems.

4.5 Exercise setup and VM configuration

1. Introduction and visualisation of images.
2. Cloud detection.
3. Calculate monthly composites.
4. Classification and post-processing.

Step-by-step setup:

- Connect to your VM

1. Open terminal and connect via ssh to your VM:

```
ssh -i /path/to/your/private-key-file eouser@remote-server-ip
```

2. Check, using ls, that files are in `/bids_tutorial/exercise2/`

You should have:

```
Dockerfile input_data notebooks output
```

3. Run docker container with agriculture services.

```
sudo docker run --name agriculture_tutorial -h agri_tutorial -it
--rm -p 8888:8888 agriculture_tutorial:latest
```

This script will also forward the port 8888 of the VM. In this way, you will be able to connect to the Jupyter notebook from your personal laptop/your pc.

4. In another terminal window on your personal laptop/PC, run the following command:

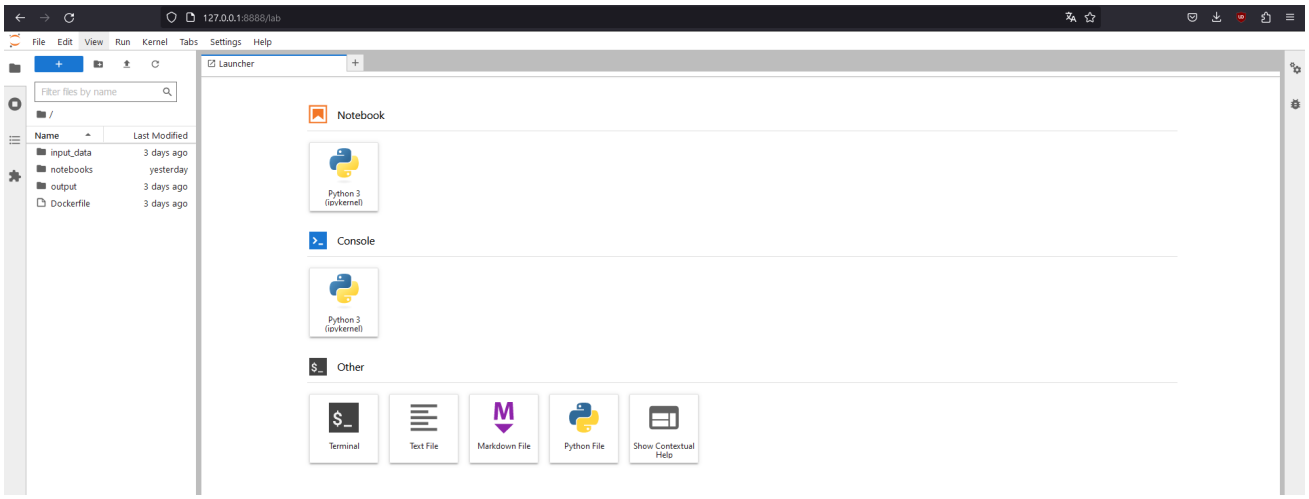
```
ssh -i /path/to/your/private-key-file -N -L
localhost:8888:localhost:8888 eouser@remote-server-ip
```

This command sets up a secure tunnel via ssh to forward traffic from a port on your local machine to a port on a remote server. If the command returns a permission denied error, it means the port you specified is already in use. Just change the first 8888 to 8889. If the terminal seems frozen, the command has been run correctly.

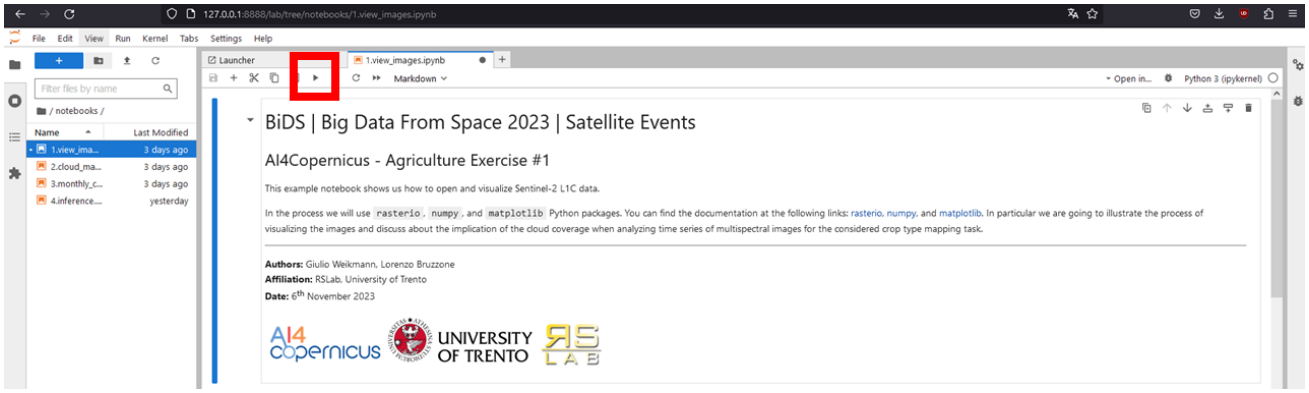
5. You can now copy the token from the jupyter terminal (or open the link 127.0.0.1[…]) and paste it into your browser. Otherwise, open your preferred browser and navigate to:

```
localhost:8888/
```

Or the port you specified if you encountered the error. The Jupyter server will ask for the token printed in 4. Copy-paste it and hit login. Now, you can open and run the notebooks with the exercises. After entering the Notebook at the specified link, the following webpage will show.



From there, open the notebooks folder on the left panel and open the first exercise.



To execute the code cells and follow the exercises, you can run the notebook using the play button highlighted in the image above.

5 Further reading and resources

- Antonis Troumpoukis, Iraklis Klampanos, Despina-Athanasia Pantazi, Eleni Tsalapati, Mohanad Albughdadi, Mihai Alexe, Vasileios Baousis, Omar Barrilero, Bryce Billière, Alexandra Bojor, Pedro Branco, Lorenzo Bruzzone, Andreina Chietera, Philippe Fournand, Richard Hall, David Hassan, Michele Lazzarini, Adrian Luna, Dharmen Punjani, George Stamoulis, Giulio Weikmann, Marcin Ziółkowski, Xenia Ziouvelou, Manolis Koubarakis and Vangelis Karkaletsis, "Bridging the European Earth-Observation and AI Communities for Data-Intensive Innovation", In: *2023 IEEE Ninth International Conference on Big Data Computing Service and Applications (BigDataService)*, Athens, Greece, 2023, pp. 9-16, doi: 10.1109/BigDataService58306.2023.00008. [LINK]

- Giulio. Weikmann, Claudia Paris and Lorenzo Bruzzone, "TimeSen2Crop: A Million Labeled Samples Dataset of Sentinel 2 Image Time Series for Crop-Type Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 4699-4708, 2021, doi: 10.1109/JSTARS.2021.3073965.

- Claudia Paris, Giulio Weikmann, Lorenzo Bruzzone, "Monitoring of agricultural areas by using Sentinel 2 image time series and deep learning techniques," Proc. SPIE 11533, Image and Signal Processing for Remote Sensing XXVI, 115330K (21 September 2020); https://doi.org/10.1117/12.2574745

- Giulio Weikmann, Claudia Paris, Lorenzo Bruzzone, "Multi-year crop type mapping using pre-trained deep long-short term memory and Sentinel 2 image time series," Proc. SPIE 11862, Image and Signal Processing for Remote Sensing XXVII, 118620O (12 September 2021); https://doi.org/10.1117/12.2600559

- Sergio Albani, Omar Barrilero, Michele Lazzarini, Adrián Luna, Paula Saameño, "Exploring the Climate-Security nexus with spaceborne data". European Commission, Joint Research Centre, Albani, S., Loekken, S., Soille, P., Proceedings of the 2021 conference on Big Data from Space – 18-20 May 2021, Albani, S.(editor), Loekken, S.(editor), Soille, P.(editor), Publications Office, 2021, https://data.europa.eu/doi/10.2760/125905

- Manolis Koubarakis (Ed.). 2023. Geospatial Data Science: A Hands-on Approach for Building Geospatial Applications Using Linked Data Technologies (1st. ed.). ACM Books, Vol. 51. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3581906

- Koubarakis, M., Bereta, K., Bilidas, D., Pantazi, DA., Stamoulis, G. (2022). A Data Science Pipeline for Big Linked Earth Observation Data. In: Curry, E., Auer, S., Berre, A.J., Metzger, A., Perez, M.S., Zillner, S. (eds) Technologies and Applications for Big Data Value . Springer, Cham. https://doi.org/10.1007/978-3-030-78307-5_19

- AI4Copernicus consortium, "AI4Copernicus Technical Documentation" [LINK]

- Lorenzo Bruzzone, Multitemporal Analysis, 6th ESA Advanced Training Course on Land Remote Sensing, [LINK]

------End of Document------