EasyChair Preprint
№ 592

Advances in Interactive Processing and Visualisation with Jupyter on the JRC Big Data Platform (JEODPP)

Davide De Marchi and Pierre Soille

October 26, 2018

ADVANCES IN INTERACTIVE PROCESSING AND VISUALISATION WITH JUPYTER ON THE JRC BIG DATA PLATFORM (JEODPP)

Davide De Marchi, Pierre Soille

European Commission, Joint Research Centre (JRC)
Directorate I. Competences, Unit I.3 Text Data Mining,
via Fermi 2749, 21027 Ispra (VA), Italy

ABSTRACT

Inside the JRC Earth Observation Data and Processing Platform (JEODPP), the *Interactive data visualization and analysis component on Jupyter notebooks* is more and more taking the shape of a complete and powerful prototyping environment. This paper describes the latest developments in the interactive platform by highlighting recent advances in fields like: datasets availability, new processing functions, new visualization methods and applications for users without programming capabilities.

Index Terms— JEODPP, Jupyter notebooks, interactive, visualization, Sentinel, Copernicus.

1. INTRODUCTION

The developments presented in this paper are implemented on the JRC Earth Observation Data and Processing Platform (JEODPP) []. This platform serves the needs of JRC policy support activities requiring big data capabilities for analyzing geospatial data and is implemented on a hardware solution scalable to the multi-petabyte range. This is achieved on the JEODPP by considering distributed storage coupled with a cluster of computing nodes. The JEODPP can be viewed as a three layers' pyramid with a petabyte scale storage and processing basis. The first layer accommodates massive batch processing. The second layer provides a remote desktop environment with all software needed for further developing legacy applications. Interactive visualization and analysis is provided by a web-based environment integrated in a Jupyter notebook [].

Inside this last component of the platform, many new datasets and functions were recently developed, and this paper gives an overview of the most significant advances.

2. ADVANCES IN DATA SETS AVAILABLE

Since its inception, the JEODPP platform has been characterized by providing users with a wide variety of geographic datasets, both raster and vector. Complex collections such as those originating from the Sentinel-1, Sentinel-2 and Sentinel-3 satellites are available for users along with many global DEMs (EU-DEM [], SRTM [], GEBCO [], etc.), as well as vector datasets such as Natura2000 [], Effis forest fires [], Administrative Nuts [],

dataset regarding land use and land cover at European level (Corine Land Cover [] and Urban Atlas []). The process of acquisition of new datasets is continuous and has recently led to the availability, as an example, of a new global DEM at a resolution of 30 meters produced by the Japan Aerospace Exploration Agency (JAXA) and called "ALOS World 3D AW3D30" []. As in the case of the other DEMs, the deferred mode processing functions of the JEODPP APIs allow to obtain high-impact visualizations such as that presented in figure 1 and displaying the coloured hill-shading with a custom colour scheme.

Fig. 1: ALOS Global Digital Surface Model "ALOS World 3D - 30m (AW3D30)" displayed inside JEODPP as a colored hill-shading [].

Other dataset recently made available are the Sentinel-1 global mosaic [], and a cloud-free Sentinel-2 global mosaic calculated from images acquired in 2017 []. Many others will follow in the coming months. No less important is the availability of many basemaps that can be used as background of views and ranging from the classic OpenStreetMap, OpenTopoMap, up to MODIS data with daily granularity, high resolution aerial images, maps in neutral colors that better enhance the geographic content superimposed on them. The selection of the basemap to use or of the dataset to be displayed, takes place in a very simple way that allows the user to view the datasets available in a tree structure, easily navigable and searchable using keywords as well as the self-completion functions available in the Python language.

3. EMBEDDING PYTHON CODE INTO THE SERVER-SIDE TILE ENGINE

At the base of the interactive component of JEODPP there is a library developed in C++ language that represents the real heart of the Tile Engine, the highly parallelized component that creates in real-time and in deferred mode, the raster tiles to send to the visualization on the ipyleaflet widget.

After selecting the datasets to be displayed, the user of the JEODPP platform can build processing chains that allow to transform and process the input data in order to extract the desired information. The basic elements of these chains are a series of processing steps that have been implemented within the Tile Engine through the integration of open source processing libraries or libraries developed over the years within the JRC. These libraries (mialib and pktools among all) provide substantially all the functions necessary for the processing of geographical data and allow the creation of complex processing chains having within them, for example, morphological operators, image classification and segmentation functions, band arithmetic, image filtering in space and time.

Although the list of processing steps is rather extensive and comprehensive, it has emerged the need to add functions defined by the user. In a language like Python dozens of very efficient libraries that manipulate images are available (such as Numpy, Scipy nd-image, Scikit-image, Python Imaging Library, OpenCV, etc ...). We have therefore added a special processing step, called *execute*, which allows the user to send to the Tile Engine that operates server-side, any function written in Python and that could potentially use such libraries.

Thanks to the Python *inspect* module, the source lines of the user function are read and sent to the C++ Tile Engine server, where a Python on-the-fly interpreter is instantiated. The code is then executed within the interpreter at each tile request. The function can freely modify the input image pixels to pass them to the next step of the processing chain.

As an example, in figure 2, an application of the *execute* processing chain to the detection of stubble burning (the deliberate setting fire of the straw stubble that remains after wheat and other grains have been harvested) from Sentinel-2 images, based on a simple algorithm that takes as inputs the bands B04, B06, B08 and B11 (Short Wave Infra Red band).

This new development is opening many new scenarios to the JEODPP users that gain a completely new flexibility in the analysis and processing of geospatial datasets. Moreover, it will allow, in the near future, to also take benefit of the many Machine Learning libraries available in Python that could be injected inside the server-side processing chain to extract valuable information from EO data using artificial intelligence tools.

Stubble burning detection for Sentinel2:

B04 < 1000 AND
B06 < 1200 AND
B08 < 1200 AND
B11 > 500 AND
B11 < 1600

Function definiton that implements the stubble burning algorithm:

```python
def stubble(v4, v6, v8, v11min, v11max):
    global img
    img = numpy.ones_like(band0)
    img[band0>=v4] = 0
    img[band1>=v6] = 0
    img[band2>=v8] = 0
    img[numpy.logical_or(band3<=v11min,
                         band3>=v11max)] = 0
```

Multi-band processing chain containing the python function call:

```python
bands = ["B04","B06","B08","B11"]
p = coll.processMulti(bands)
        .execute(stubble,1000,1200,1200,500,1600)
        .band(0).scale(0,1).colorCustom(["Lime"])
```

Fig. 2: (Left) Informal description of the Stubble Burning detection algorithm. (Right) Python implementation of the algorithm using Numpy methods. (Bottom) Multi-band processing chain containing the *execute* step calling the custom Python function.

4. USER-FRIENDLY APPLICATIONS

Most JEODPP users are researchers or scientists who possess programming capabilities and use python to interact with the viewing and analysis environment within notebooks. Despite this, over time it has emerged the need to create simpler tools that would allow to work with geographic datasets through a graphical interface. This is even more true for the manager level users facing the platform without any programming knowledge. For these reasons we decided to develop applications that would allow customers to perform complex functions without the need to write code. Within the Jupyter world the choice fell on the ipywidgets suite for the creation of user interfaces and on the use of components such as Bqplot [] for charting functions and Qgrid [] for displaying alphanumeric data in rows with intuitive scrolling, sorting, and filtering controls.

4.1. S2-Explorer

The first fully fledged applicaton implemented is devoted to easy browsing and searching of Sentinel-2 products. It has a tabbed interface containing many functions going from searching of products covering the current view extents, displaying one or multiple products on the map, selecting between many possible RGB bands compositions, apply local stretching to the products visualization, filtering of the searched products on multiple criteria (for instance on cloud cover or product type or acquisition dates, etc.).

Input capabilities allow the users of S2-Explorer to easily add a custom vector shapefile to the map, and use it to select the Sentinel products covering its features. Many export functions are available, e.g. export the list of the selected Sentinel-2 products to be used for batch processing operations, export of the map view in high resolution Tiff file, creation of an animation video that displays one after the other all the filtered products. Recently the *measure* and *draw* tabs were added to allow measuring of distances and areas on

the map and the creation of vector features with the possibility to save them in shapefile format.
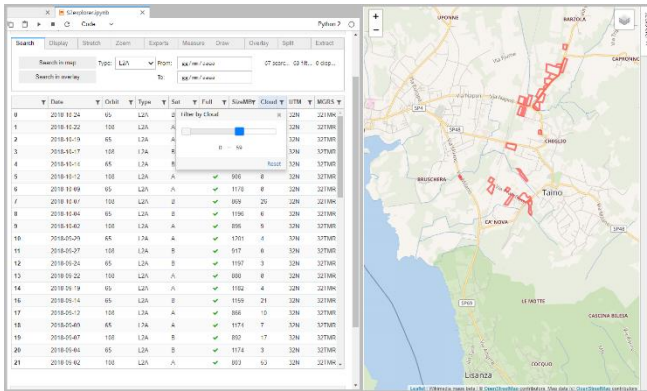

Fig. 3: S2-Explorer in action. Loading of a custom vector shapefile on the map, search of the Sentinel-2 products that cover it and filtering on cloud cover percentage.

5. EXPLOITATION OF TEMPORAL DIMENSION

The Copernicus constellation of EO satellites is characterized by the high temporal resolution of its acquisitions. As an example, after the lunch of Sentinel-2B satellite, all emerged lands will be covered by a new optical product at least every five days. This gives space to many applications were the exploitation of the small revisit time allows important analytical results, being in the agricultural field, in the forest applications arena or in disaster monitoring. JEODPP tries to fully exploit the temporal dimension of Sentinel products by providing a series of analysis tools based on multi-temporal data.
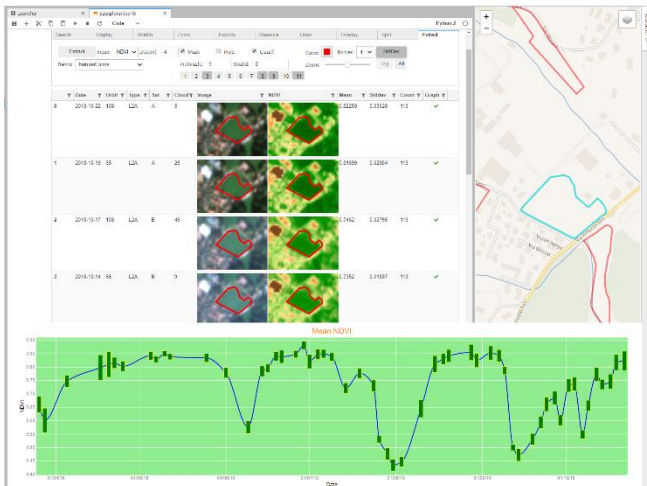

Fig. 4: Extraction of the multi-temporal NDVI profile over the pixels inside a polygon by the S2-Explorer tool.

5.1. Temporal profiles

One of the tabs of the S2-Explorer tools is dedicated to the extraction of multi-temporal information over the pixels covered by polygonal features. After having searched and filtered the Sentinel-2 products, users can ask the system to calculate, for each of the products, the mean value of an index (like NDVI or EVI, SAVI, etc.) or of a band, together with their standard deviation and plot both values on a temporal graph where the line series represent the mean value inside the polygon and the vertical bars the homogeneity measured by the standard deviation (see figure 4.). This function has many applications in the agricultural field and can be activated also on a custom edited polygon. It has to be noted that the extraction is done server-side on the highly parallel cluster and gives the results in few seconds even in case of dozens or hundreds of input products.

5.2. Easy comparison using the split map control

One of the mostly appreciated functions recently added to the interactive visualization component of JEODPP is the ability to easily compare two different datasets. It can be used for comparison of completing different datasets (like a DEM and a EO image, or a basemap and a vector dataset, etc.) or on comparison of two acquisition dates of the same product. It is based on a new function available on ipyleaflet: *the split map control*. When activated, the tool displays a vertical line at the center of the map and the two datasets on each side of the line. The user can move the line horizontally to quickly compare the left and right map on the same piece of land.
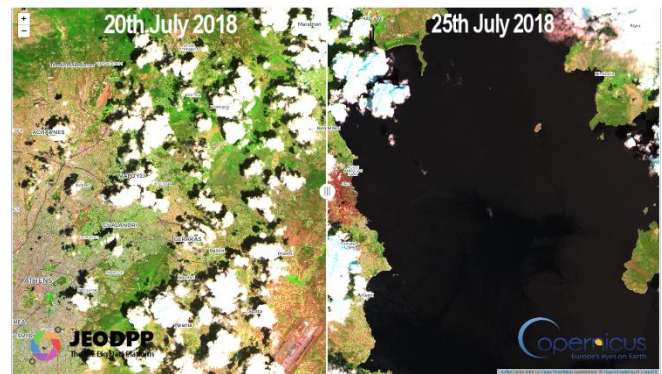

Fig. 5: Temporal comparison of two SWIR RGB compositions, from Sentinel-2 images acquired before and after the Mati fires in Greece.

Using the split map control, many videos were created by the JEODPP team in the recent months to monitor the effects of disasters like fires, drought and floods (Mati fire in Greece, fire near Pisa in Italy, fire in Monchique, Portugal, etc.), comparing the last Sentinel-2 product acquired before the event with the first acquired after it.

5.3. Georeferenced temporal video

Recent versions of ipyleaflet allow for the overlay of videos on top of the map. We are using this new function to create georeferenced videos starting from the multi-temporal Sentinel-2 products acquired over the same area. The videos can be "played" directly over the map (even while zooming and panning) and give an interesting insight on the evolution of the land over time with possible application in agriculture (like the evaluation of harvesting times) or in forest monitoring (e.g. control illegal deforestation).

Fig. 6: Video overlay of a georeferenced temporal video showing one after the other all the Sentinel-2 images acquired in time over the area.

6. CONCLUDING REMARKS AND OUTLOOK

Work on the JEODPP interactive component will continue, following the needs of the many real use cases that are helping to shape the future of the platform, both in terms of datasets availability and of advanced visualization and analysis functions. …

7. REFERENCES

[] T. Tadono, H. Ishida, F. Oda, S. Naito, K. Minakawa, H. Iwamoto, *Precise Global DEM Generation by ALOS PRISM*, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol.II-4, pp.71-76, 2014.

[2] Jones, C.D., A.B. Smith, and E.F. Roberts, *Book Title*, Publisher, Location, Date.