15 June 2021

BIG DATA PROJECT – OKP4

YASSINE BEN MOHAMED & APPANNA MACHIMANDA
MSC AGFOOD DATA MANAGEMENT

SUMMARY

Today, in France and around the world, users rely on freely available data of Sentinel satellite constellations through platforms like ESA's SNAP or Google Earth Engine, to produce nitrogen modulation maps, biomass estimation maps, and more. Maps of plant health or vigor indicators are very useful to the farmer responsible for managing his farm, and for the agronomist responsible for providing advice based on the parameters observed. As part of a vision of enriching the decentralized OKP4 data exchange platform, we are interested in setting up a Geo service for the detection of irrigated plots in France, which will give further insights for better understanding crop yield, water resource usage, and other parameters.

METHODOLOGY

Through this project we will discuss topics in the following order:

- Introduction
- Research paper
- Labelled dataset
- Google Earth Engine
- Geemap python API
- Tkinter interface
- Results
- Conclusion

1. INTRODUCTION

The aim of this project was to identify if an agricultural parcel has been irrigated or not, by using remote sensing data. Remote sensing means that we should be able to carry out our analysis without having to be physically present at the location.

In our case, this was made possible because of using Sentinel-1 satellite data. This data represents the physical characteristics of the agricultural parcel (soil water content) in terms of the signal backscatter values from vertical polarization bands.

The data is then matched with a labelled dataset of "irrigated" and "non-irrigated" classes to carry out supervised machine learning.

2. RESEARCH PAPER

We used a research paper titled "**Irrigation mapping using Sentinel-1 time series at field scale" by** Q. Gao, M. Zribi, M.J. Escorihuela, N. Baghdadi, P. Quintana Segui

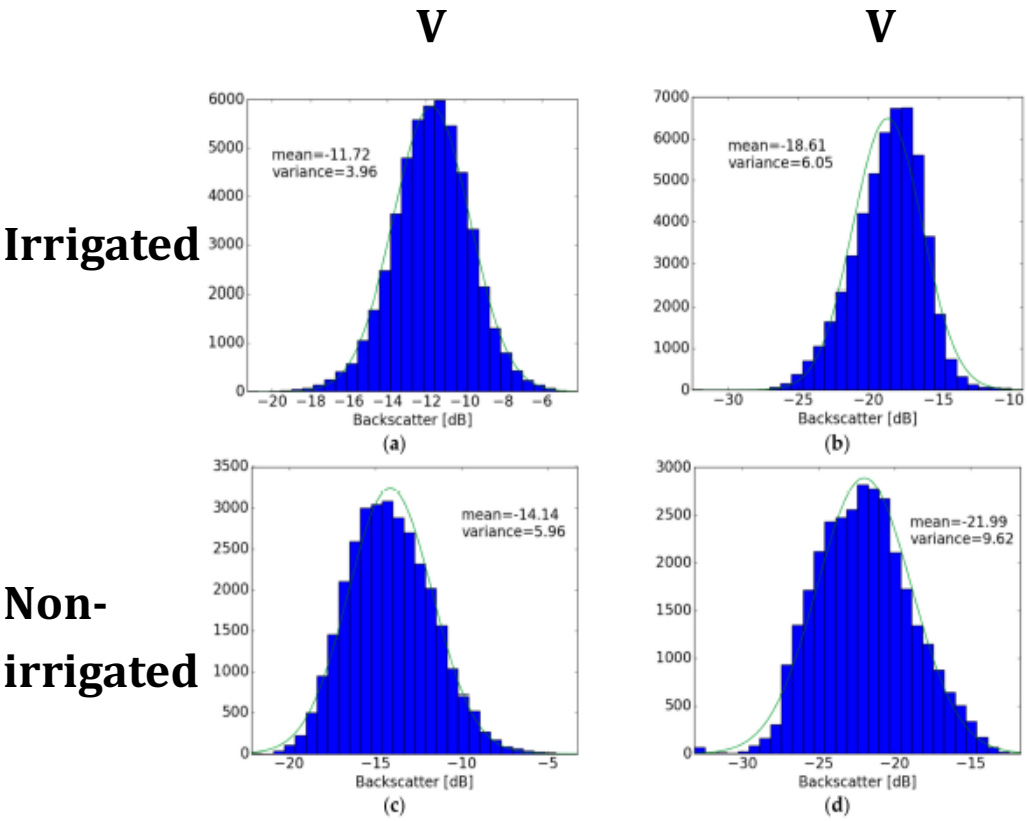
The key learnings from this paper in connection with our project are:

- The Vertical polarization backscatter values of Sentinel 1 data, when considered temporally (values across time) offer separability between the irrigated and non-irrigated parcels.
- The resolution of images is 10m and temporal resolution of the Sentinel images, (time between 2 images of same pixel) is about 12 days.
- The mean and variance of the pixel backscatter values over time, are different for the 2 classes because of the changes due to soil moisture.
- The Vertical polarization shows high sensitivity to soil moisture. So we use both VV(Vertical-Vertical) and VH(Vertical-Horizontal) bands.

Backscatter values are different for Irrigated vs Non-irrigated classes since the water content creates differences in soil dielectric constant.

VV mean is more sensitive to soil moisture and VH variance is more sensitive to vegetation.

The VV penetrates the soil more and the VH is susceptible to dispersion by contacting vegetation. Described below are the results from the research paper.



We then tested out if the separation in values is true for our dataset as well. Described below is a table which confirms the hypothesis.

Irrigated				Not Irrigated			
VH_mean	VH_variance	VV_mean	VV_variance	VH_mean	VH_variance	VV_mean	VV_variance
-21,8053547	14,23196439	-12,41082206	7,481139995	-22,3972138	16,55717559	-13,7867217	8,061101542
-19,0454165	15,94858681	-11,40373285	13,94855979	-23,14136613	17,01204504	-13,66727912	10,66690987
-17,2051128	15,78768167	-10,87674928	13,91291006	-23,00960737	30,20305834	-14,0403085	16,27940799
-19,33613862	11,71965291	-12,40407007	5,369426122	-22,92884193	21,97138336	-14,36153806	11,8765518
-16,35346707	4,999627943	-10,28780651	3,908149751	-23,99853087	24,06818235	-15,63996399	8,282459472
-18,43107883	9,459681026	-10,787711	5,973177064	-19,78348444	6,94739556	-11,73839858	4,092249622
-19,36154888	9,618377559	-11,23423183	5,18146284	-22,54277933	28,39699062	-14,3489027	10,59786409
-19,78110045	11,13464254	-12,2709568	6,661404798	-16,56169095	8,879823748	-4,68445492	8,338575166
-16,72651711	4,228032286	-10,84145691	3,008602557	-18,07165795	7,668036982	-12,03294069	4,957430025
-17,79158129	4,925316865	-11,88577002	3,833523697	-14,17673725	11,49716677	-9,55947044	7,758063164
-19,12430255	11,03440487	-11,52371051	7,654903376	-12,63639583	3,541870805	-8,164672277	4,244608736
-21,70507345	11,15759027	-13,05546261	6,756620471	-18,29908914	5,357973001	-12,78774282	4,046800717
-19,98635831	15,44302045	-12,00186752	8,411362784	-29,98423445	31,88895967	-21,56557659	11,27349254
-20,5354466	18,37523036	-12,84786523	11,76003011	-29,96755558	22,21686435	-21,90391098	11,94036617
-16,62246443	4,226840078	-10,89942599	3,064636683	-22,34732055	9,173909089	-16,32064308	5,464301391
Mean							
-18,92073077	10,81937667	-11,64877595	7,128394007	-21,32310037	16,35872235	-13,6401683	8,525345486

Notice the difference in mean and variance values between the irrigated and non-irrigated classes.

3. LABELLED DATASET

For supervised classification of our machine learning model, we needed to use a labelled dataset. We used a SIGPAC dataset from the Catalonia open data portal catalogue, for the region of Urgell. The data release for 2020 was used.

This dataset contains a column describing “irrigated” (blue) and “non-irrigated” (red) parcels.

The data can be downloaded in the form a shapefile.

This file has to be “ingested” into Google Earth Engine since we will be using this to carry out the machine learning.

The ground reality of the parcels is reflected in this data. However, we must note that the actual pixel values within an agricultural parcel can have varying values.

4. GOOGLE EARTH ENGINE

Earth Engine offers a complete suite of geographical processing tools which run on the cloud.

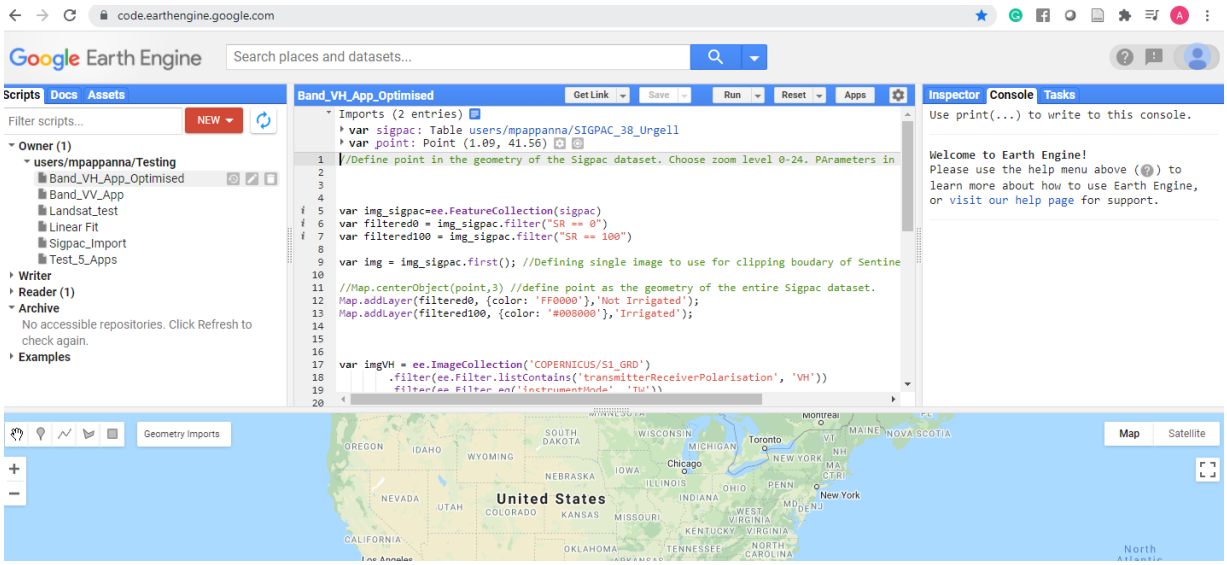
This allows us to process the extremely large datasets of satellite imagery like Sentinel 1 which are several hundreds of GB in size.

In the early days of our understanding how to use Earth Engine, we directly used the Javascript API in the code console.

However, with the increasing size of our code structures, and for our eventual implementation in an easy-to-use interface, we decided to use earth engine through the Python API using the “geemap” package.

The user must have a google account which can be used to create an account in Earth Engine, in order to authenticate and use the interface tool.

All export tasks will be downloaded into the user’s google drive cloud storage.



5. PYTHON API

The geemap package requires a special installation process where a separate environment must be created with a specific version of python.

The official documentation on the google earth engine website is only for the Javascript console. We used this to obtain conceptual understanding, and then referred to the geemap documentation to implement specific functionality.

There were many milestones we encountered in executing our solution using the Python API:

- Import and display on the map, the Image collection of Sentinel 1 data “COPERNICUS/S1_GRD”

- Ingest the SIGPAC data into Earth Engine and import the feature collection into the python workbook.
- Define a default scope of Sentinel data to load : geographically, temporally, and by the selected properties.
- Clip the sentinel data to the boundary geometry of the sigpac data so that resources are not wasted in rendering irrelevant pixel data.
- Selecting the machine learning model and defining number of points and scale.
- Exporting the model classified output of the same labelled dataset used for training.
- Optimizing the points, and scale for best output match.
- Defining the prediction zone and obtaining the output.

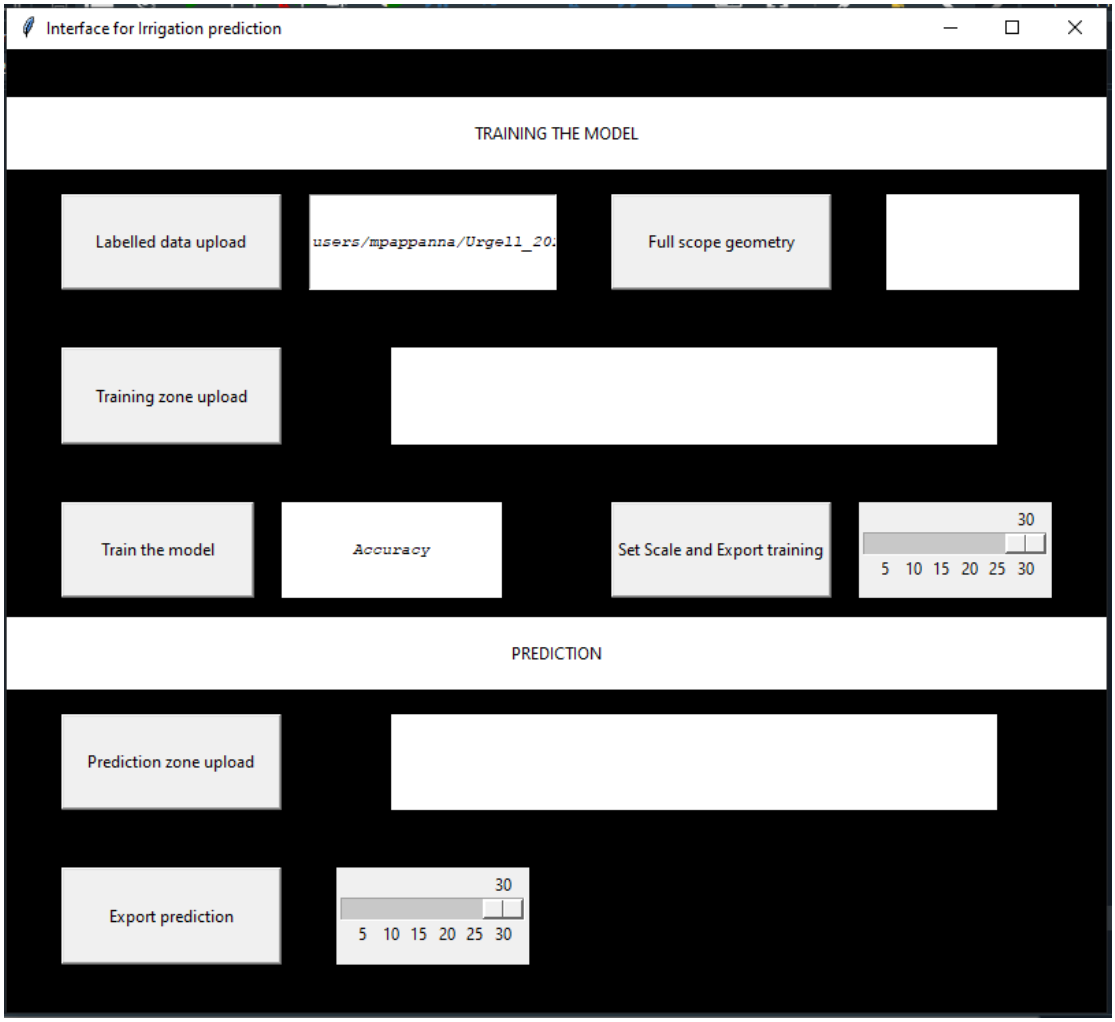
6. TKINTER INTERFACE

The interface was developed simultaneously with the backend production and changes were integrated periodically.

The following are the steps to use the interface:

- The “table ID” of the SIGPAC data (ingested into Earth engine) has to be pasted beside the “Labelled data upload” button. After pasting, the button is clicked to load the data.
- The “Full scope geometry” is the geojson file which contains the outline of the labelled dataset (SIGPAC data).
- A “Training zone” refers to an area selection within the labelled dataset geometry, so that a region with a good distribution of both irrigated and non-irrigated classes can be used for training. If this is not uploaded, the training scope defaults to the full scope of the labelled dataset.
- Then we can click the “Train the model” button. After the model completes training, the accuracy is displayed beside the button.
- Now we can set the scale for export and click the button to generate a .tif image of the classified training zone.
- To run a prediction, we can upload the geojson file of the region (Occitanie in our case) and then setting the scale and exporting the predition result as a .tif image.
- The export process can take a very long time based on the scale which has been set. However, once the task has been initiated, the

process runs on the earth engine cloud and downloads the output into the associated google drive account.



7. RESULTS

For prediction on the Occitanie region, a scale 30 or 20 export results in a single .tif image. Scale 10 produces 2 images and a scale 5 produces 8 images. If multiple images are produced, they have to be merged using a tool like QGIS.

To run prediction on any department, only the boundary geojson file is required. This can be downloaded readily from a web search. For predicting a specific region, the boundary can be drawn using a website tool like geojson.io

In our case, we ran prediction for the entire Occitanie region. We also retrieved the agricultural parcel boundaries from RPG (Registre parcellaire graphique) data.gouv.fr official website.

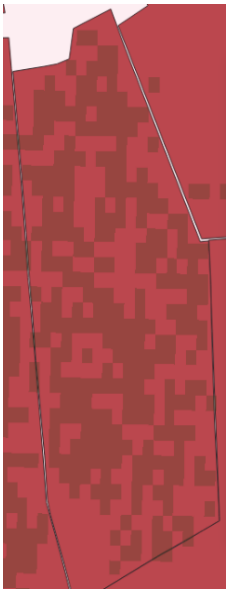
Using QGIS software, we calculated the zonal statistics for each parcel. If the percentage of irrigated pixels in a parcel is more than 50%, then we can conclude that the parcel has been irrigated.

The count of pixels in each parcel of the predicted image, varies based on the scale of export. A scale of 30m will have bigger pixels and hence lesser count compared to a scale of 5m which will have smaller pixels and hence more count.

Fid: 19875

ID_ILOT: 6034280

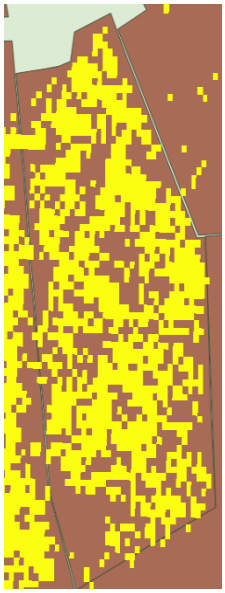
Area:156 km^2



Scale 20

Irr: 50.9%

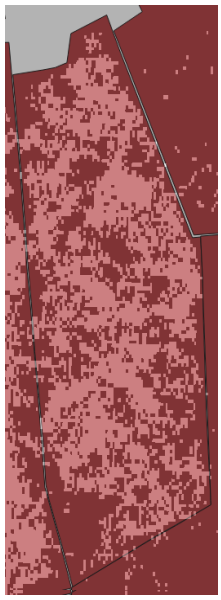
Count: 535



Scale 10

Irr: 50.6%

Count: 2145



Scale 5

Irr: 50.6%

Count: 8572

The parameters for the training of the model is fixed in the backend and cannot be changed in the interface. We used a random forest classifier model. For sampling the data, 20k random points were blasted onto the SIGPAC dataset where each point samples pixels at a scale of 30m. If the scale is set to be smaller than this, the training data becomes more sensitive to the intra parcel variation of pixel backscatter values and also speckle effect which negatively affect the model performance for predictions.

Described below are some example results with comparisons:

- For most parcels, an output of scale 10 gives reliable results.

- Reliable results are judged by sufficient count of pixels.
- For large parcels, scale 20 can also be used and the resolution of the prediction matches that of scale 10 and scale 5.
- For very small parcels, scale 20 and scale 10 may not show any output. Only scale 5 can be used reliably in this case.
- Exporting a scale 5 prediction image takes a very long time and has very large file size.
- The zonal statistics for the parcels are exported into an excel file for ready reference.

8. CONCLUSION

The skills required to create a process and tool for remotely identifying irrigated parcels were: Python programming, QGIS software, Google Earth Engine. The interface tool we have developed can be used by non-programmers for generating the training and prediction images. These images have to be manually loaded into QGIS software to compute the zonal statistics and identify if a parcel has more than 50% of the pixels classified as “Irrigated”.

The scale and number of sampling points are the key parameters for training the classifier. We need to have sufficient number of points to represent the parcels. Scale has to be low enough for intra-parcel variation in backscatter values but high enough to not be over-sensitive to the target label of training.

A training zone incorporating a good mix of both irrigated and non-irrigated classes can further improve the training and hence inference of the model.

The scale of export is an important consideration because smaller scale requires more computation time and a heavy output image file size.

Google Earth Engine allows us to use massive computational resource in order to use this tool. Still the size and scale of these images resulted in slow render times during the development of our program.

Further improvements to this tool can be incorporated to allow full functionality of predictions for completely non-technical users as well. Ideally a map interface can be included to define prediction areas on the fly and obtain a percentage of irrigation classification.

REFERENCES

- Q. Gao, M. Zribi, M.J. Escorihuela, N. Baghdadi, P. Quintana Segui. Irrigation mapping using,Sentinel-1 time series at field scale. Remote Sensing, MDPI, 2018, 10 (9), 18 p. 10.3390/rs10091495.hal-01900567
- https://developers.google.com/earth-engine
- https://code.earthengine.google.com/
- https://analisi.transparenciacatalunya.cat/Medi-Rural-Pesca/Sistema-d-informaci-geogr-fica-de-parcel-les-agr-c/5ytz-a6f2
- https://github.com/giswqs/geemap/tree/master/examples
- https://docs.python.org/3/library/tkinter.html