# 10TH ADVANCED TRAINING COURSE ON LAND REMOTE SENSING

## Streamlined observations of forests using EO Browser and eo-learn

Matej Batič, Sinergise

# About Sinergise

SINERGISE

*A small company from a small country, changing the world one step at a time*

sentinelhub

*Service providing you fast (and simple) access to satellite data*

*EO data inside*

eoresearch@sinergise.com

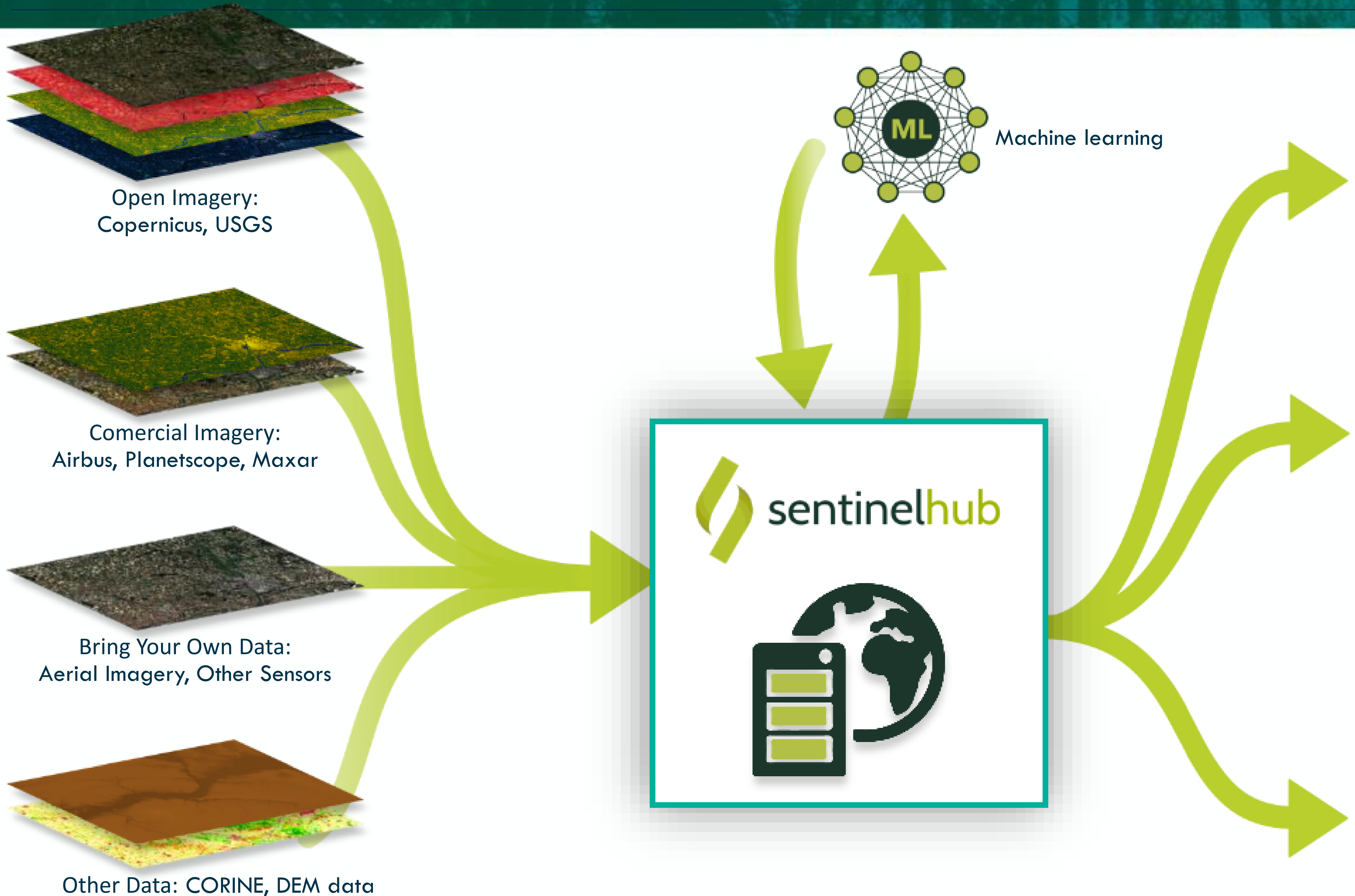
*Providing answers to questions you didn't know you had.*

The Earth Monitoring Competition
Copernicus masters
OVERALL WINNER
AZO

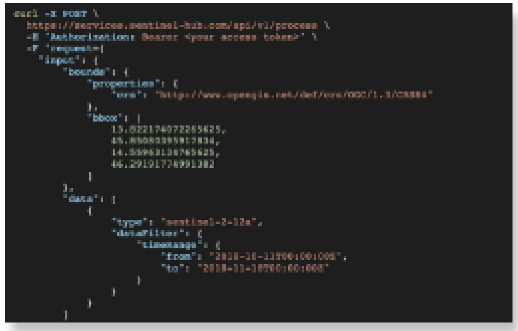
The Earth Monitoring Competition
Copernicus masters
WINNER 2016
T-SYSTEMS OPEN TELEKOM CLOUD CHALLENGE
AZO

EUROPEAN EO COMPANY OF THE YEAR 2018
EARSC

# Sentinel Hub



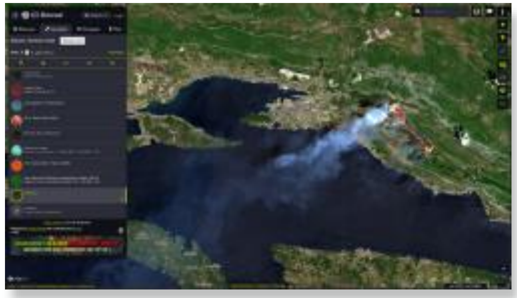
API



```
curl -X POST \
  https://services.sentinel-hub.com/api/v1/process \
  -H 'Authorization: Bearer <your access token>' \
  -F 'request={
    "input": {
        "bounds": {
            "properties": {
                "crs": "http://www.opengis.net/def/crs/OGC/1.3/CRS84"
            },
            "bbox": [
                13.822174072265625,
                45.85080395917834,
                14.55963134765625,
                46.29191774991382
            ]
        },
        "data": [
            {
                "type": "sentinel-2-l2a",
                "dataFilter": {
                    "timeRange": {
                        "from": "2018-10-01T00:00:00Z",
                        "to": "2018-11-01T00:00:00Z"
                    }
                }
            }
        ]
```

Programming: Javascript, Python, R…

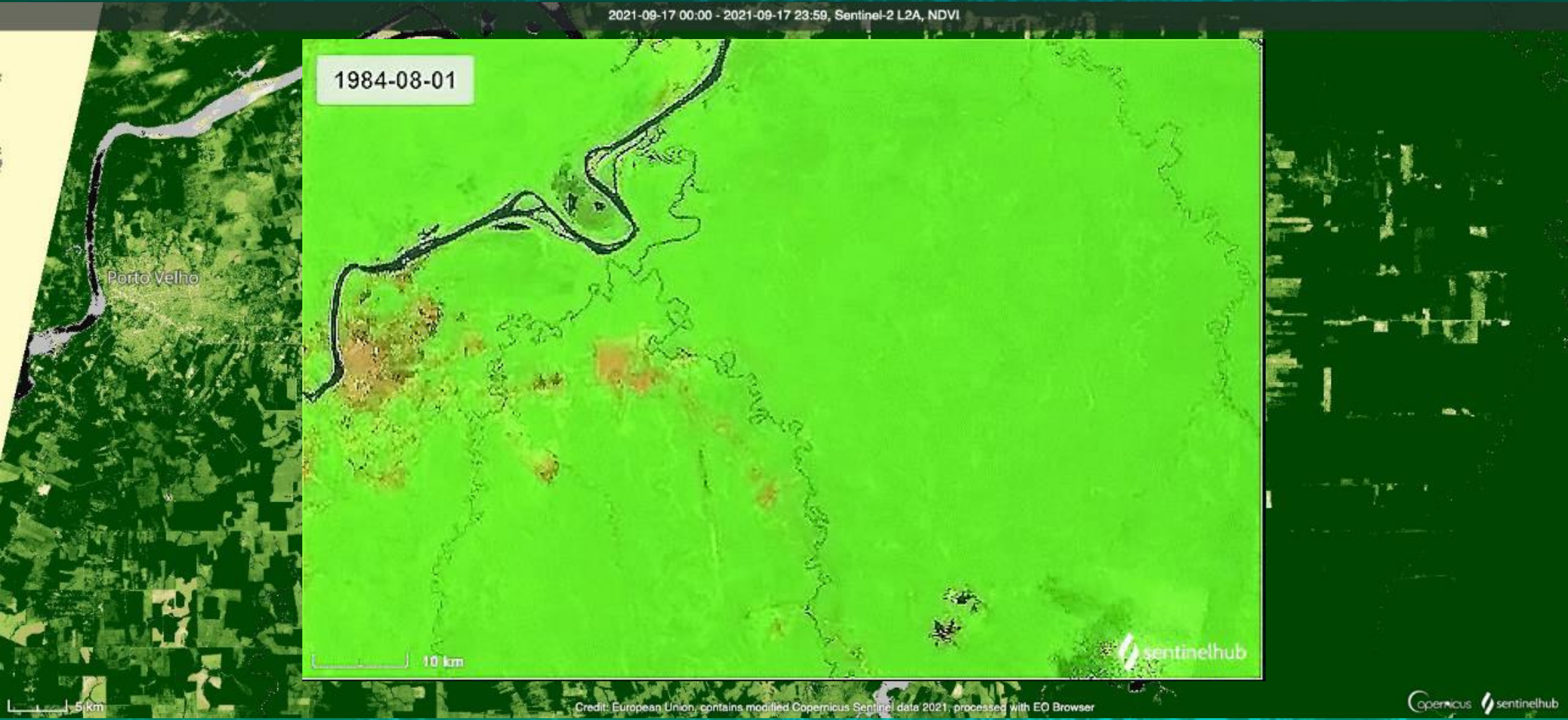
OGC



Web/Mobile Apps

GIS: Cloud and Desktop Apps

# Sentinel Hub



Credit: European Union, contains modified Copernicus Sentinel data 2021, processed with EO Browser

# Sentinel Hub OGC API

WMS REQUEST

WCS REQUEST

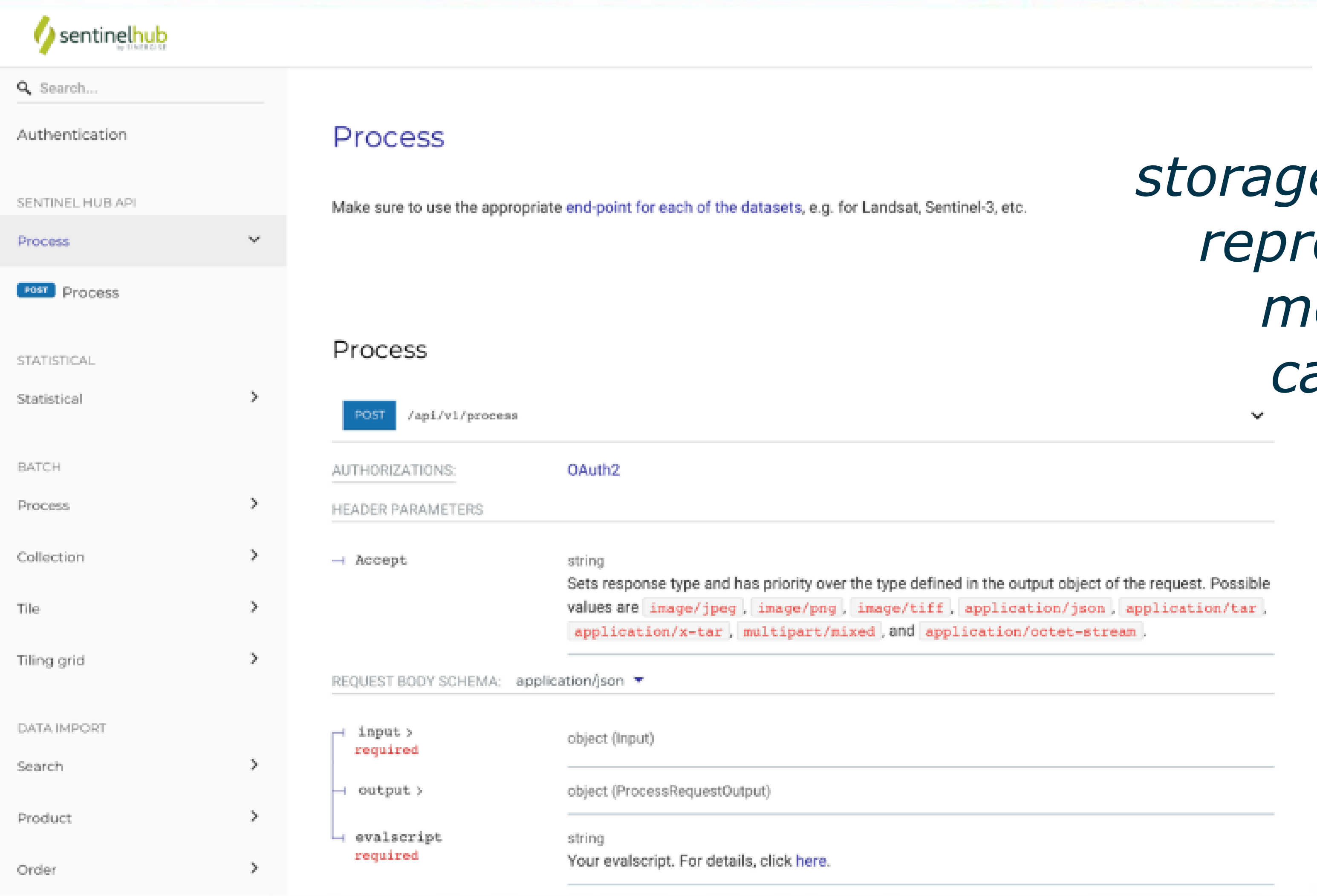
WFS REQUEST

WMTS REQUEST

```
https://services.sentinel-hub.com/ogc/wcs/<INSTANCE-ID>
?SERVICE=WCS
&REQUEST=GetCoverage
&COVERAGE=1_VV_ORTHORECTIFIED
&TIME=2018-07-17/2018-07-17
&BBOX=15.28815,107.07000,14.75496,106.09153
&RESX=20m
&RESY=20m
&FORMAT=image/tiff
&CRS=EPSG:4326
```

*No worries about download, storage, decoding, stitching scenes, reprojection, scaling, mosaicking, meta-data parsing, backscatter calibration, orthorectification …*

# Sentinel Hub API



sentinelhub

Search...

Authentication

SENTINEL HUB API

Process

POST Process

STATISTICAL

Statistical

BATCH

Process

Collection

Tile

Tiling grid

DATA IMPORT

Search

Product

Order

## Process

Make sure to use the appropriate end-point for each of the datasets, e.g. for Landsat, Sentinel-3, etc.

### Process

POST /api/v1/process

AUTHORIZATIONS: OAuth2

HEADER PARAMETERS

| | |
|---|---|
| Accept | string<br>Sets response type and has priority over the type defined in the output object of the request. Possible values are `image/jpeg`, `image/png`, `image/tiff`, `application/json`, `application/tar`, `application/x-tar`, `multipart/mixed`, and `application/octet-stream`. |

REQUEST BODY SCHEMA: application/json

| | |
|---|---|
| input ›<br>required | object (Input) |
| output › | object (ProcessRequestOutput) |
| evalscript<br>required | string<br>Your evalscript. For details, click here. |

*No worries about download, storage, decoding, stitching scenes, reprojection, scaling, mosaicking, meta-data parsing, backscatter calibration, orthorectification ...*

*Additional functionalities:*
*batch jobs,*
*statistical calculations,*
*bring your own data,*
*catalog,*
*third-party data import*

# Sentinel Hub API – basic processing options

## What data can one ”get” from SH?

Pre-defined product (true color, false color, etc.)

Custom band combination

```
return [B08, B04, B03];
```

Various indices

```
var val = (B08-B04)/(B08+B04);
return [val];
```

Complex algorithms

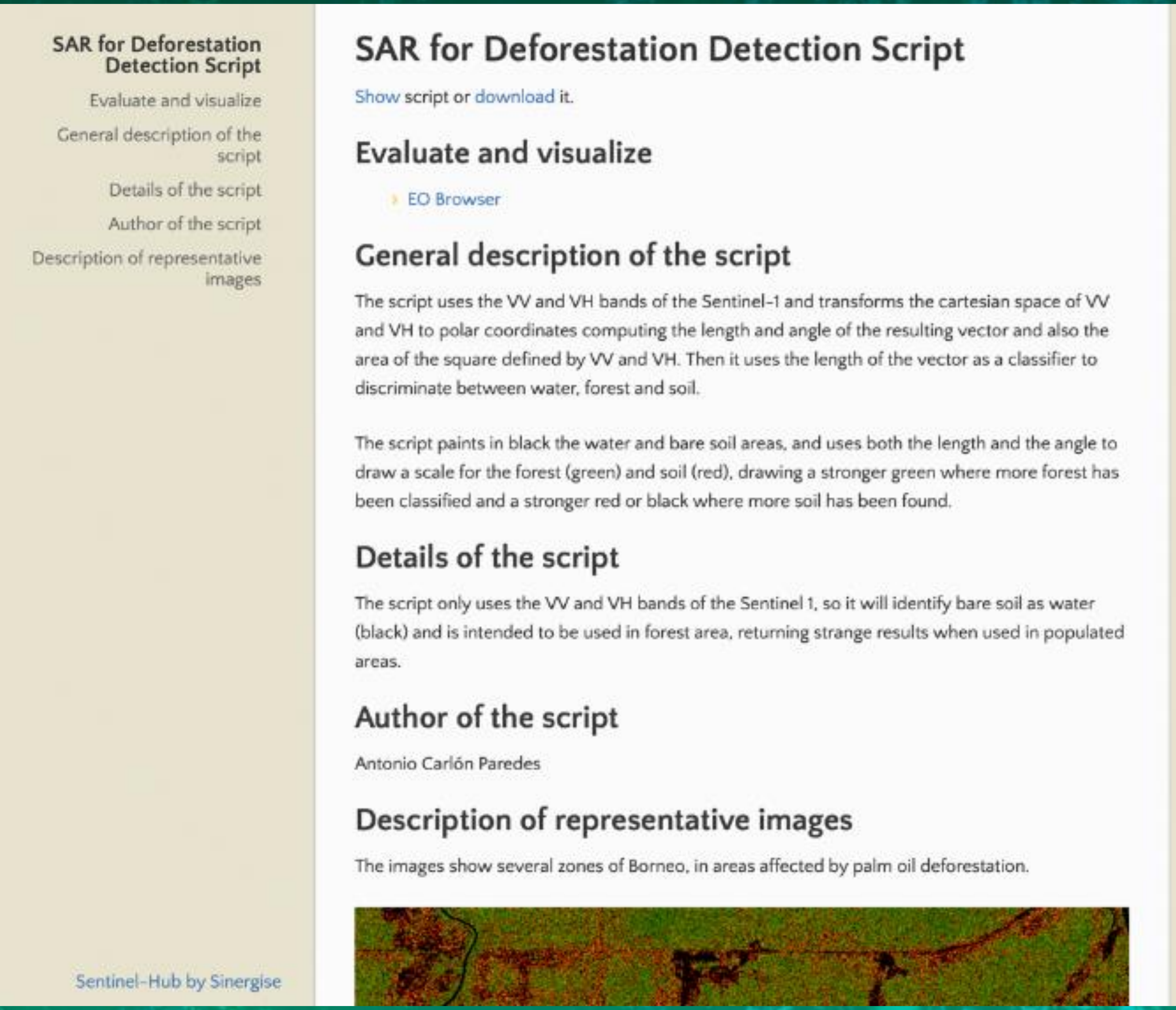
LAI, FAPAR – (basic) neural networks

Interpolated data, custom mosaicking, temporal computation

https://github.com/sentinel-hub/custom-scripts/

# Custom scripts repository

https://custom-scripts.sentinel



SAR for Deforestation Detection Script
Evaluate and visualize
General description of the script
Details of the script
Author of the script
Description of representative images

Sentinel-Hub by Sinergise

## SAR for Deforestation Detection Script

Show script or download it.

### Evaluate and visualize

- EO Browser

### General description of the script

The script uses the VV and VH bands of the Sentinel-1 and transforms the cartesian space of VV and VH to polar coordinates computing the length and angle of the resulting vector and also the area of the square defined by VV and VH. Then it uses the length of the vector as a classifier to discriminate between water, forest and soil.

The script paints in black the water and bare soil areas, and uses both the length and the angle to draw a scale for the forest (green) and soil (red), drawing a stronger green where more forest has been classified and a stronger red or black where more soil has been found.

### Details of the script

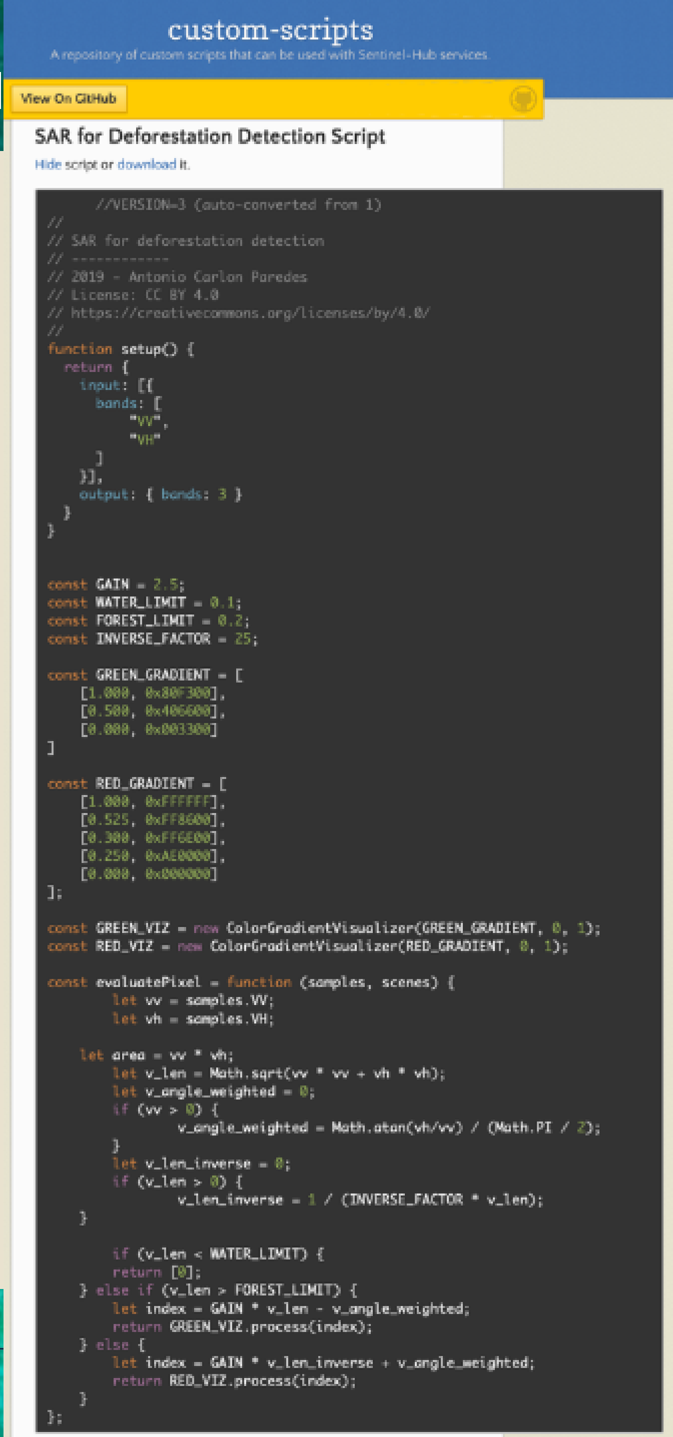
The script only uses the VV and VH bands of the Sentinel 1, so it will identify bare soil as water (black) and is intended to be used in forest area, returning strange results when used in populated areas.

### Author of the script

Antonio Carlón Paredes

### Description of representative images

The images show several zones of Borneo, in areas affected by palm oil deforestation.



custom-scripts

A repository of custom scripts that can be used with Sentinel-Hub services

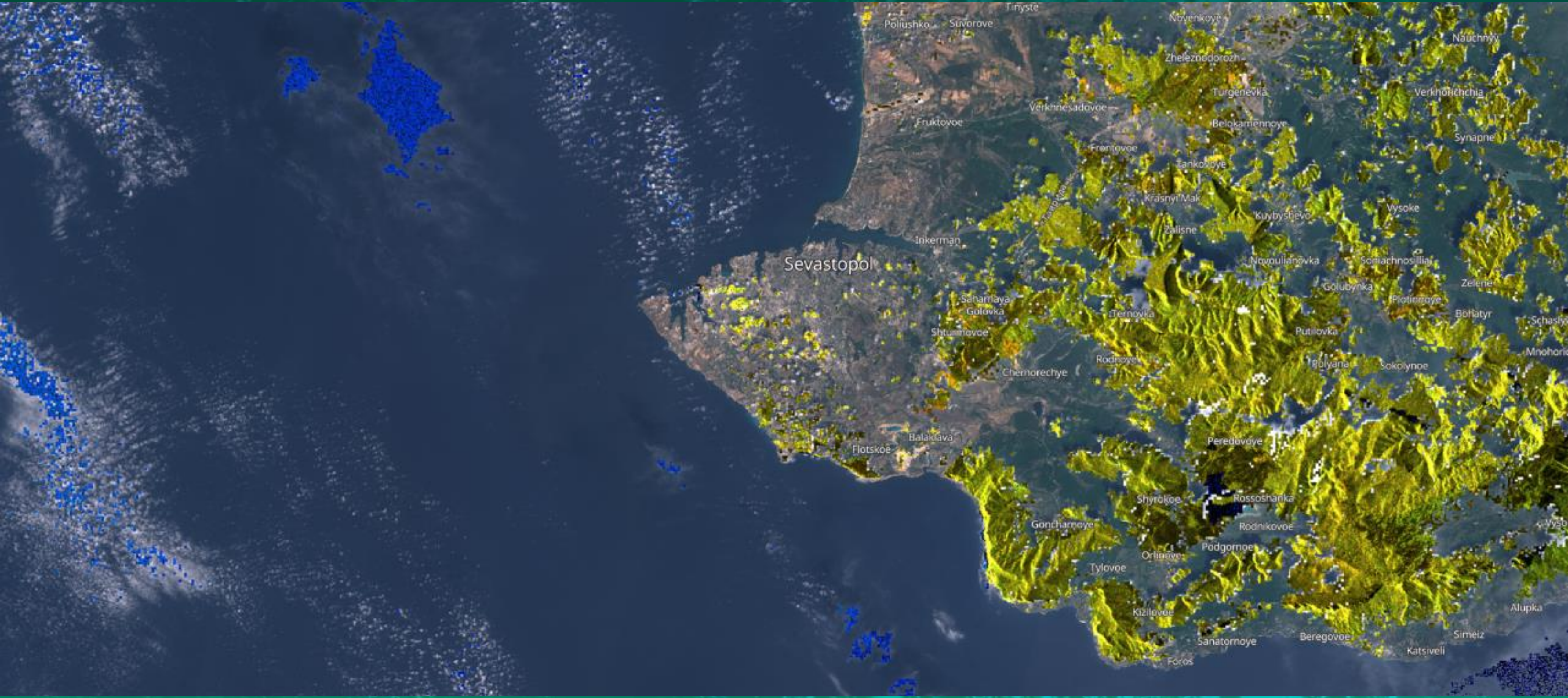
## SAR for Deforestation Detection Script

Hide script or download it.

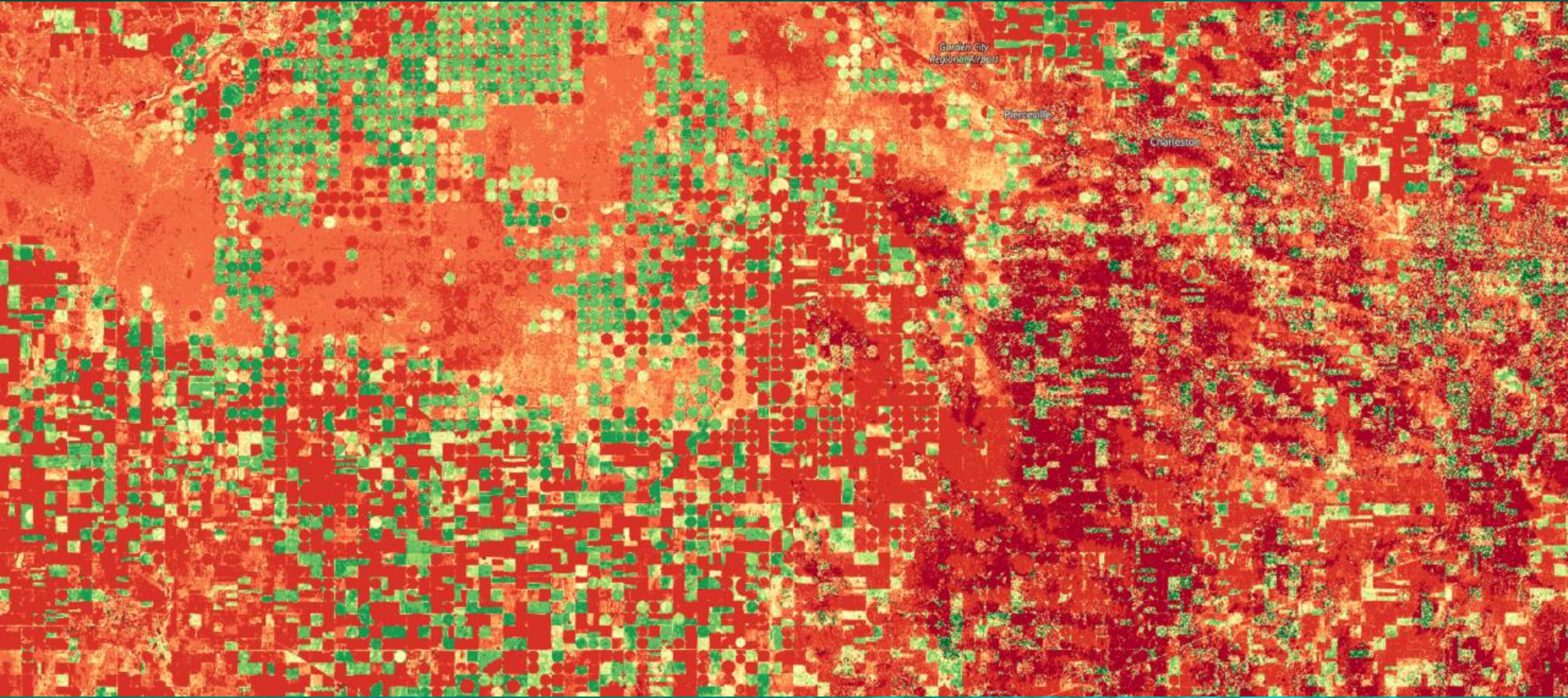
```
    //VERSION=3 (auto-converted from 1)
//
// SAR for deforestation detection
// ------------
// 2019 - Antonio Carlon Paredes
// License: CC BY 4.0
// https://creativecommons.org/licenses/by/4.0/
//
function setup() {
  return {
    input: [{
      bands: [
          "VV",
          "VH"
      ]
    }],
    output: { bands: 3 }
  }
}


const GAIN = 2.5;
const WATER_LIMIT = 0.1;
const FOREST_LIMIT = 0.2;
const INVERSE_FACTOR = 25;

const GREEN_GRADIENT = [
    [1.000, 0x80F300],
    [0.500, 0x406600],
    [0.000, 0x003300]
]

const RED_GRADIENT = [
    [1.000, 0xFFFFFF],
    [0.525, 0xFF8600],
    [0.300, 0xFF6E00],
    [0.250, 0xAE0000],
    [0.000, 0x000000]
];

const GREEN_VIZ = new ColorGradientVisualizer(GREEN_GRADIENT, 0, 1);
const RED_VIZ = new ColorGradientVisualizer(RED_GRADIENT, 0, 1);

const evaluatePixel = function (samples, scenes) {
        let vv = samples.VV;
        let vh = samples.VH;

    let area = vv * vh;
        let v_len = Math.sqrt(vv * vv + vh * vh);
        let v_angle_weighted = 0;
        if (vv > 0) {
                v_angle_weighted = Math.atan(vh/vv) / (Math.PI / 2);
        }
        let v_len_inverse = 0;
        if (v_len > 0) {
                v_len_inverse = 1 / (INVERSE_FACTOR * v_len);
    }

        if (v_len < WATER_LIMIT) {
        return [0];
    } else if (v_len > FOREST_LIMIT) {
        let index = GAIN * v_len - v_angle_weighted;
        return GREEN_VIZ.process(index);
    } else {
        let index = GAIN * v_len_inverse + v_angle_weighted;
        return RED_VIZ.process(index);
    }
};
```

# Temporal processing

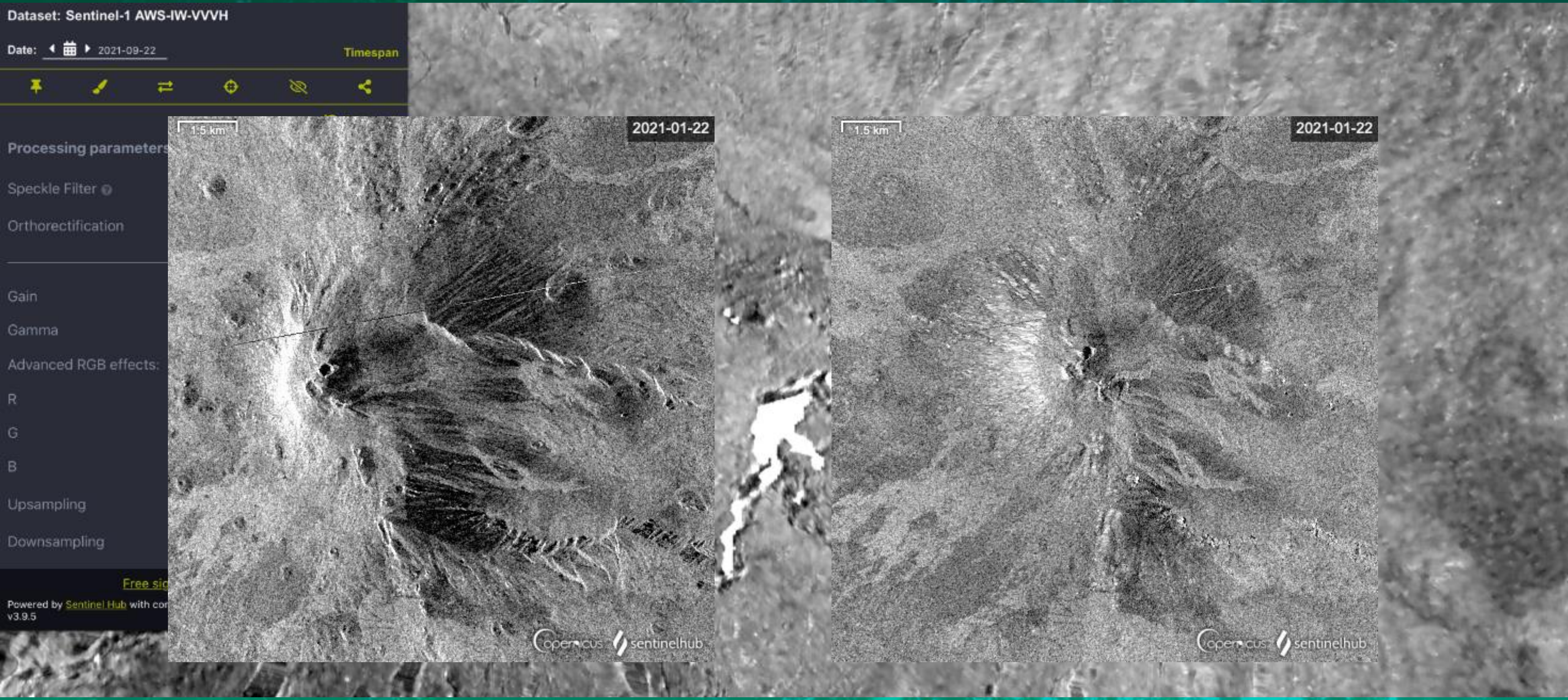
# Data fusion


Tinyste
Poliushko
Suvorove
Novenkoye
Nauchnyy
Zheleznodorozh
Turgenevka
Verkhorichchia
Verkhnesadovoe
Fruktovoe
Belokamennoye
Synapne
Frontovoe
Tankovoye
Krasnyi Mak
Kuybyshevo
Vysoke
Zalisne
Inkerman
Sevastopol
Novoulianovka
Golubynka
Zelene
Sakharnaya Golovka
Ternovka
Bohatyr
Shturmovoe
Putilovka
Rodnoye
Polyana
Sokolynoe
Chernorechye
Balaklava
Flotskoe
Peredovoye
Shyrokoe
Rossoshanka
Goncharnoye
Rodnikovoe
Podgornoe
Orlinoye
Tylovoe
Alupka
Kizilovoe
Simeiz
Sanatornoye
Beregovoe
Katsiveli
Foros

# Data fusion



# Sentinel-1 processing



Dataset: Sentinel-1 AWS-IW-VVVH

Date: 2021-09-22

Timespan

Processing parameters

Speckle Filter

Orthorectification

Gain

Gamma

Advanced RGB effects:

R

G

B

Upsampling

Downsampling

Free sig

Powered by Sentinel Hub with co
v3.9.5

# Sentinel Hub – data collections

## Open Data (global and full archives)

Sentinel-1 GRD*

Sentinel-2 L1C and L2A**

Sentinel-3 OLCI and SLSTR

Sentinel-5P

Landsat-1-8

MODIS

DEM (Copernicus, MapZen)

Copernicus Services layers

* CARD4L compliant

** L2A global from 2017 onwards

## Commercial data (on-demand, payable)

PlanetScope

Airbus Pleiades

Airbus SPOT

Maxar

## Bring your own data

Satellite imagery

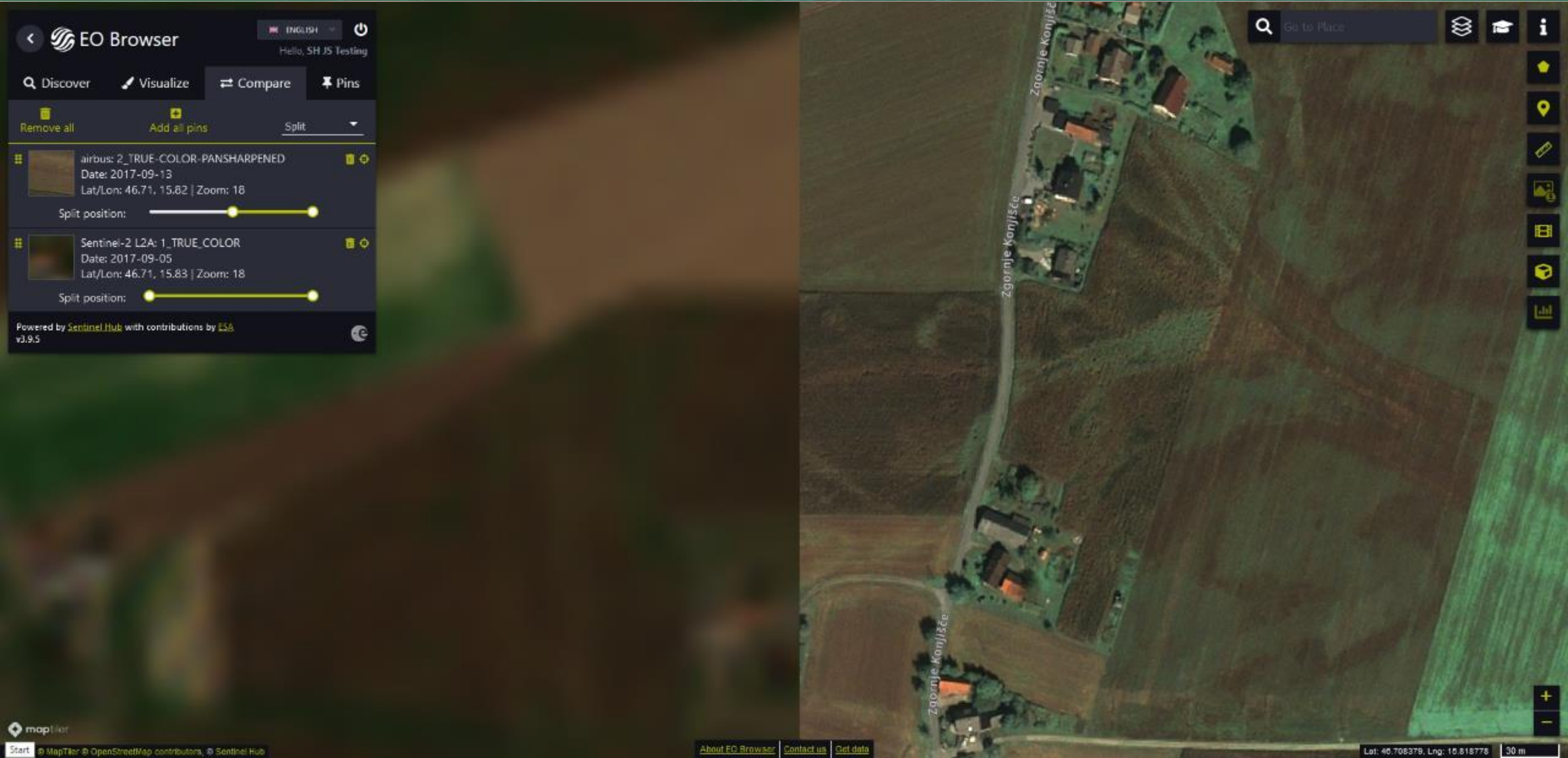
Derived products

COG/zarr format

Your data residing on AWS (EU-1 or US-West-2), Mundi, CreoDIAS, CODE-DE

Full Sentinel Hub functionality

# Third party data (commercial data)



EO Browser

ENGLISH

Hello, SH JS Testing

Discover | Visualize | Compare | Pins

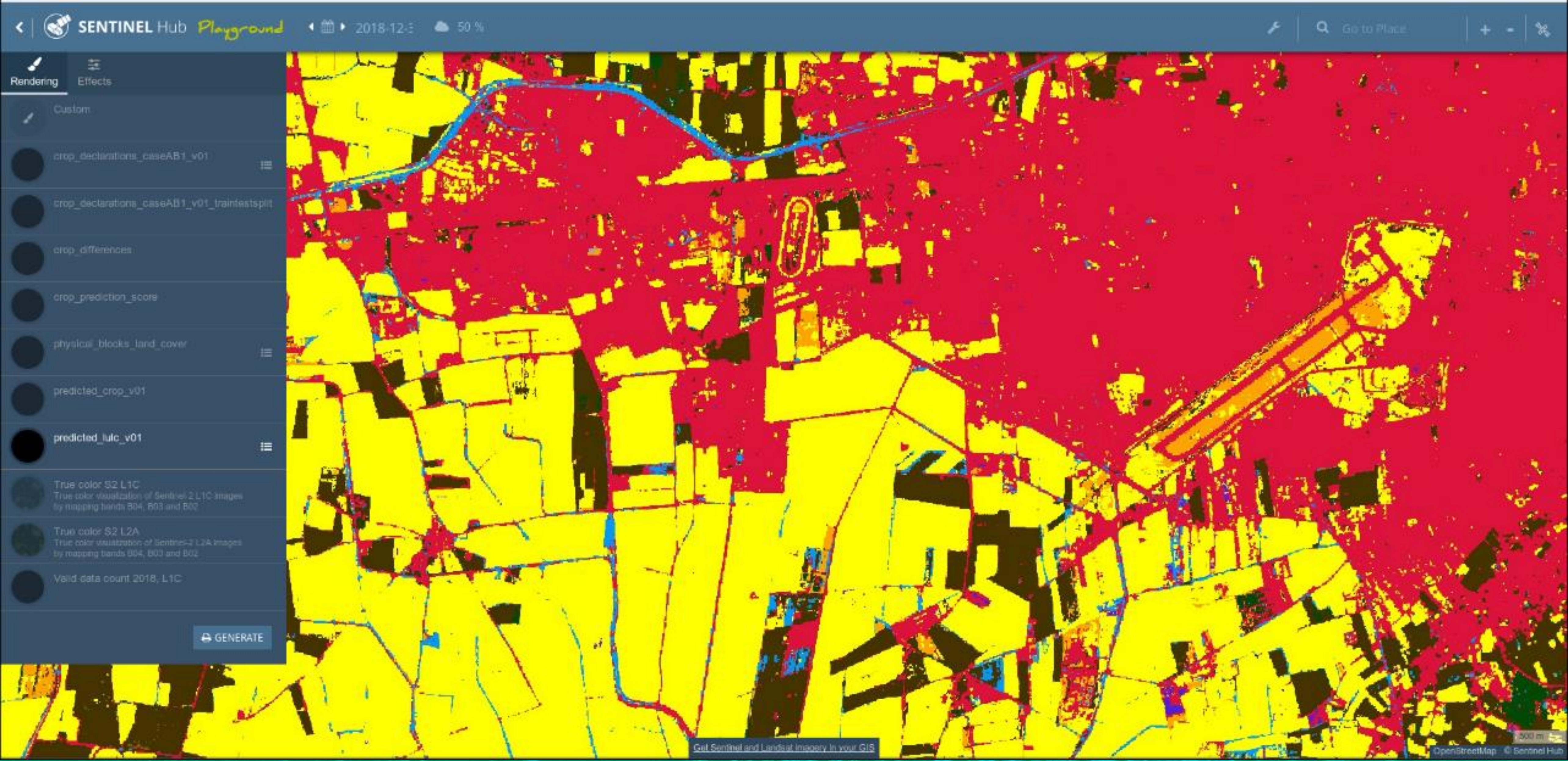
Remove all | Add all pins | Split

airbus: 2_TRUE-COLOR-PANSHARPENED
Date: 2017-09-13
Lat/Lon: 46.71, 15.82 | Zoom: 18
Split position:

Sentinel-2 L2A: 1_TRUE_COLOR
Date: 2017-09-05
Lat/Lon: 46.71, 15.83 | Zoom: 18
Split position:

Powered by Sentinel Hub with contributions by ESA
v3.9.5

# Bring your own data



Custom

crop_declarations_caseAB1_v01

crop_declarations_caseAB1_v01_traintestsplit

crop_differences

crop_prediction_score

physical_blocks_land_cover

predicted_crop_v01

predicted_lulc_v01

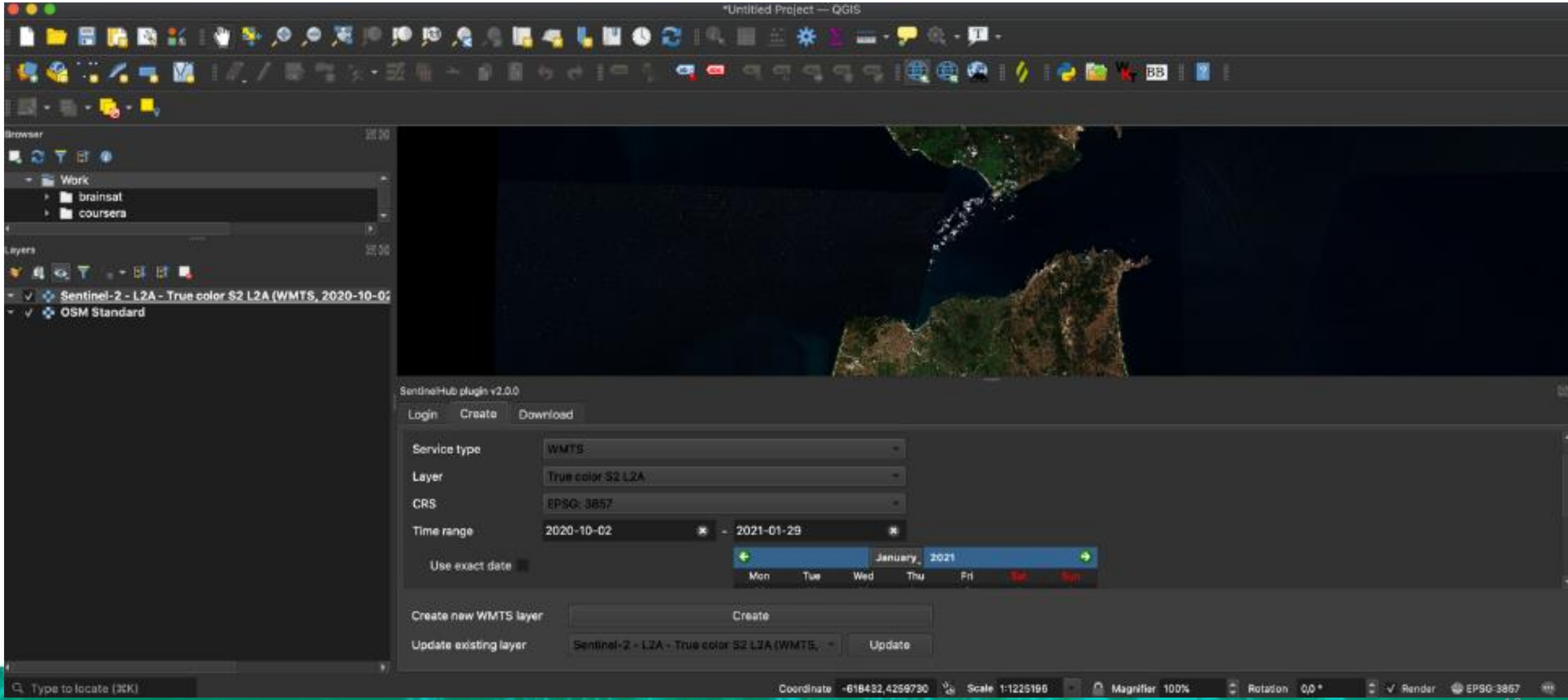
True color S2 L1C
True color visualization of Sentinel-2 L1C images by mapping bands B04, B03 and B02

True color S2 L2A
True color visualization of Sentinel-2 L2A images by mapping bands B04, B03 and B02

Valid data count 2018, L1C

# Integration capabilities

- WMS/WCS/WMTS – QGIS, ArcGIS Desktop, ArcGIS on-line
- QGIS PlugIn
- sentinelhub-py
- sentinelhub-js
- xcube-sh



*Untitled Project — QGIS

Browser

Work
brainsat
coursera

Layers

Sentinel-2 - L2A - True color S2 L2A (WMTS, 2020-10-02
OSM Standard

SentinelHub plugin v2.0.0

Login | Create | Download

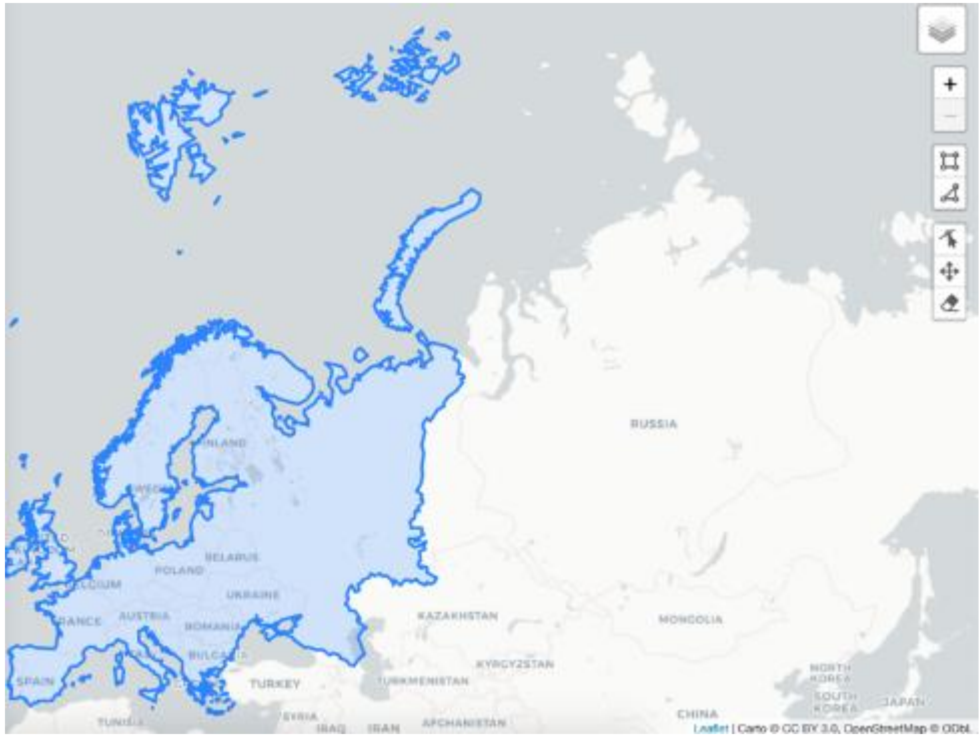
Service type: WMTS
Layer: True color S2 L2A
CRS: EPSG: 3857
Time range: 2020-10-02 - 2021-01-29
Use exact date

January 2021
Mon Tue Wed Thu Fri Sat Sun

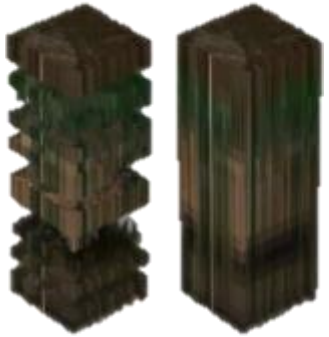
Create new WMTS layer: Create
Update existing layer: Sentinel-2 - L2A - True color S2 L2A (WMTS, | Update

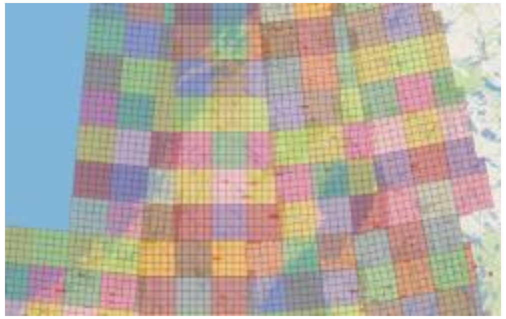
Type to locate (⌘K)
Coordinate -618432,4259730 | Scale 1:1225198 | Magnifier 100% | Rotation 0,0 ° | ✓ Render | EPSG:3857

# Large-scale processing - Batch API

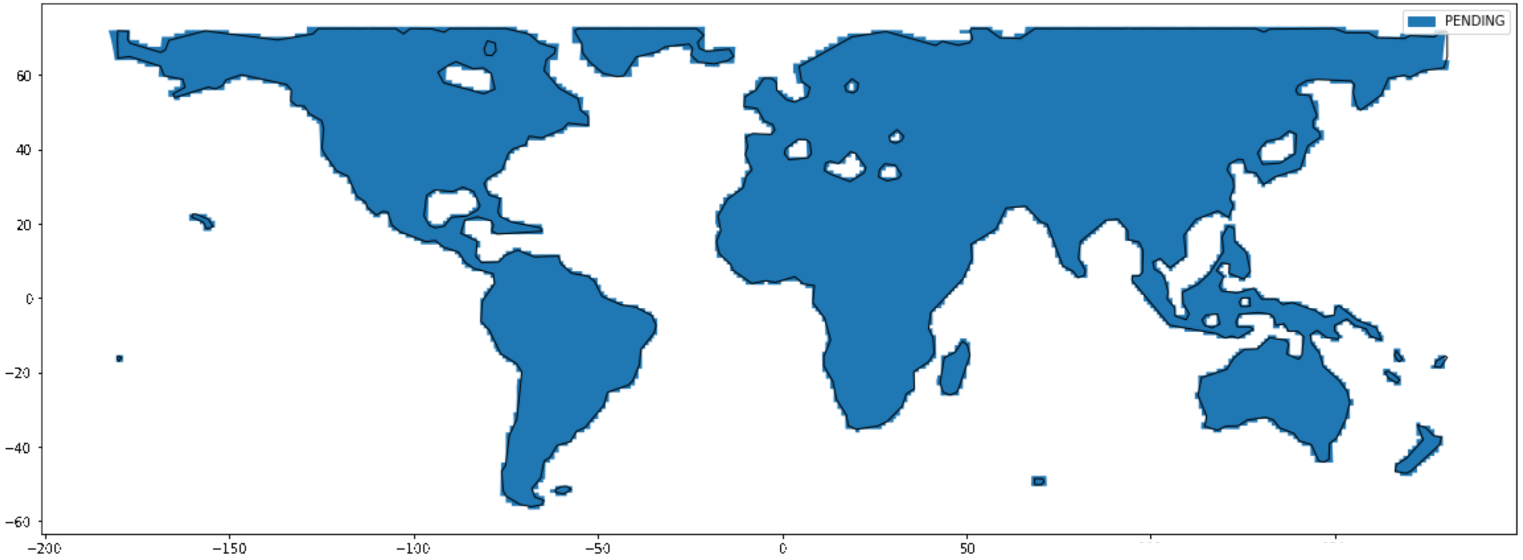


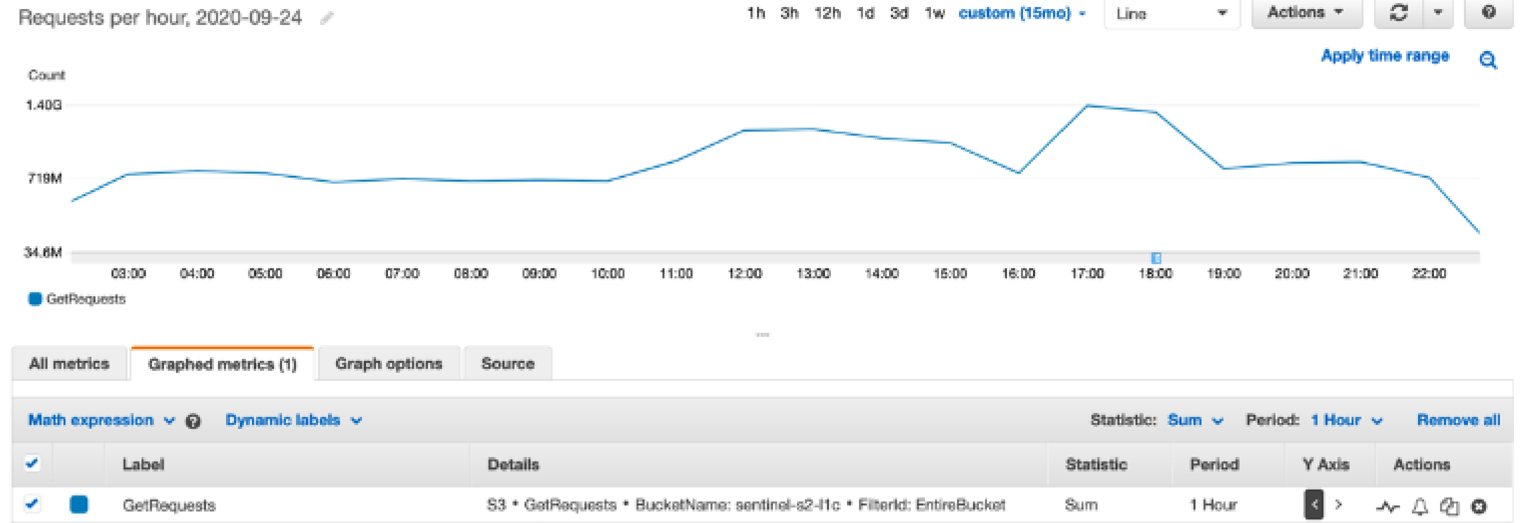
- Request data at large scale
  – either spatial or temporal.
- Run your algorithm for a whole continent.
- Pre-process vast amount of data.
- Execute, monitor for updates, follow-up processes





# Large-scale processing - Batch API





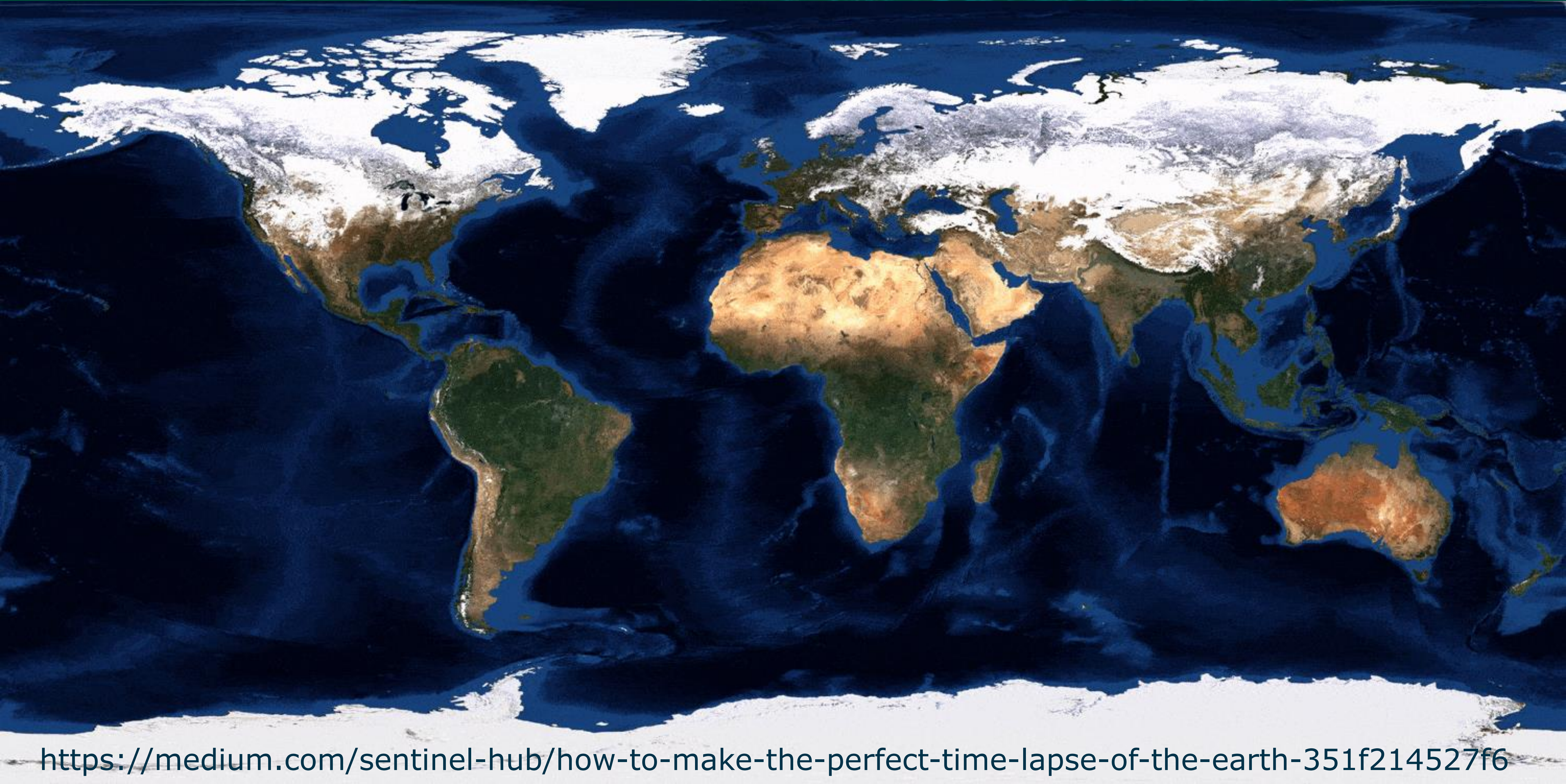
Requests per hour, 2020-09-24

Statistic: Sum Period: 1 Hour

| ☑ | | Label | Details | Statistic | Period | Y Axis | Actions |
|---|---|---|---|---|---|---|---|
| ☑ | 🟦 | GetRequests | S3 • GetRequests • BucketName: sentinel-s2-l1c • FilterId: EntireBucket | Sum | 1 Hour | < > | |

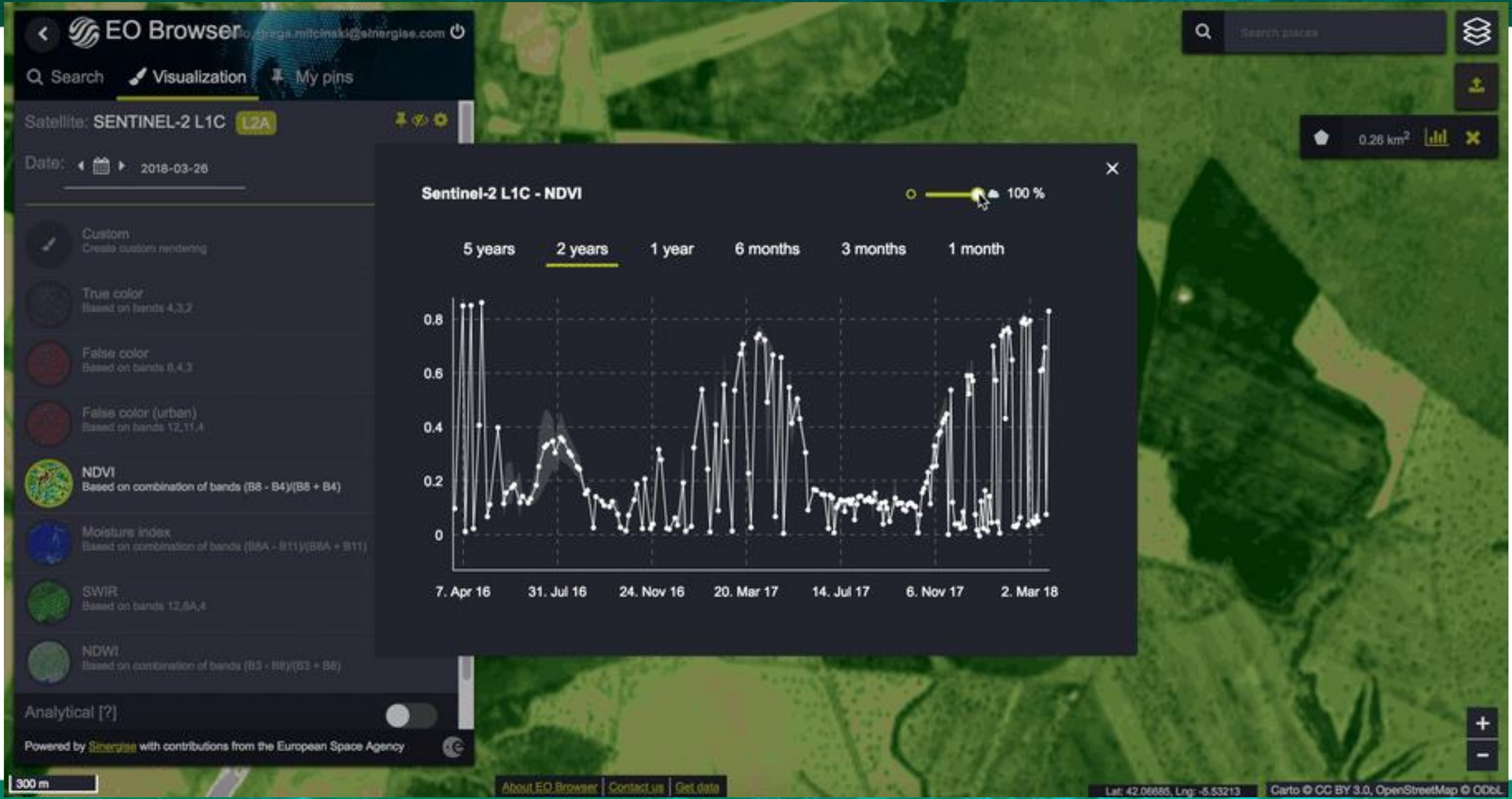
400.000 requests per second

# Large-scale processing - Batch API



https://medium.com/sentinel-hub/how-to-make-the-perfect-time-lapse-of-the-earth-351f214527f6

# Statistical API



Satellite: SENTINEL-2 L1C

Date: 2018-03-26

Sentinel-2 L1C - NDVI

Powered by Sinergise with contributions from the European Space Agency

# EO-Browser

https://apps.sentinel-hub.com/eo-browser

*Or google eo-browser*

# EO-Browser

- What is EO Browser?
- Education themes
- Search and view satellite data
- Custom visualizations on the fly
- Statistics
- Pins and comparison tool
- 3D viewer
- Downloading images
- Time-lapse creation

# sentinelhub statistics, April 2021

## Cloud API

**150 k**
registered users

**500 M**
processed requests

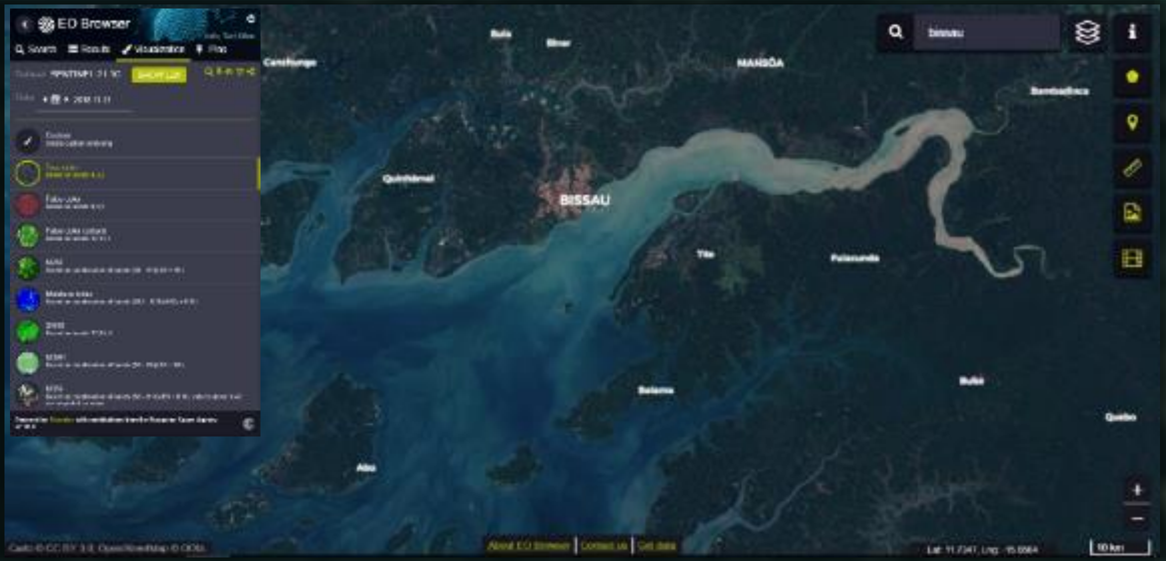
**30+ PB**
of satellite imagery

**500 TB**
added every month

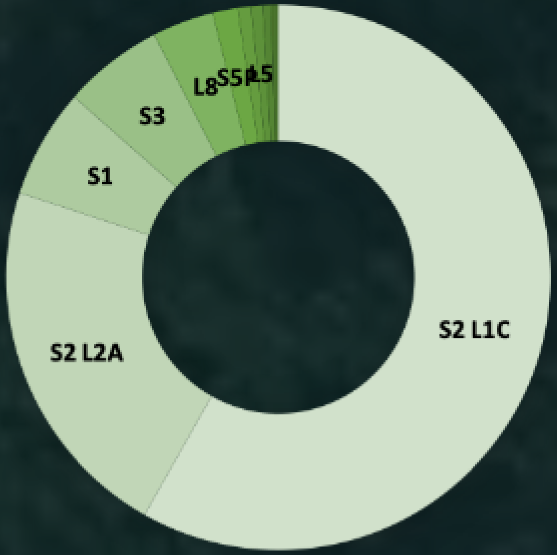
## EO Browser

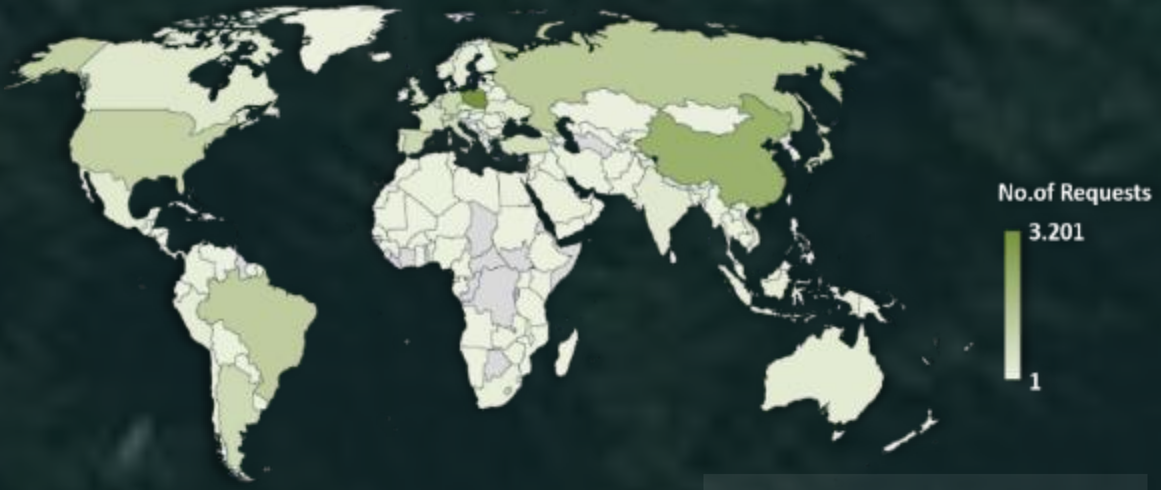
**58 k**
monthly visitors

**60 M**
processed requests



Open-source and free to use





Visitors from 159 countries

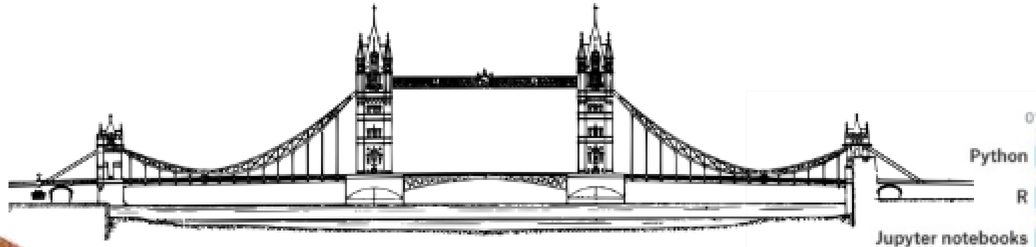
# eo-learn

https://github.com/sentinel-hub/eo-learn

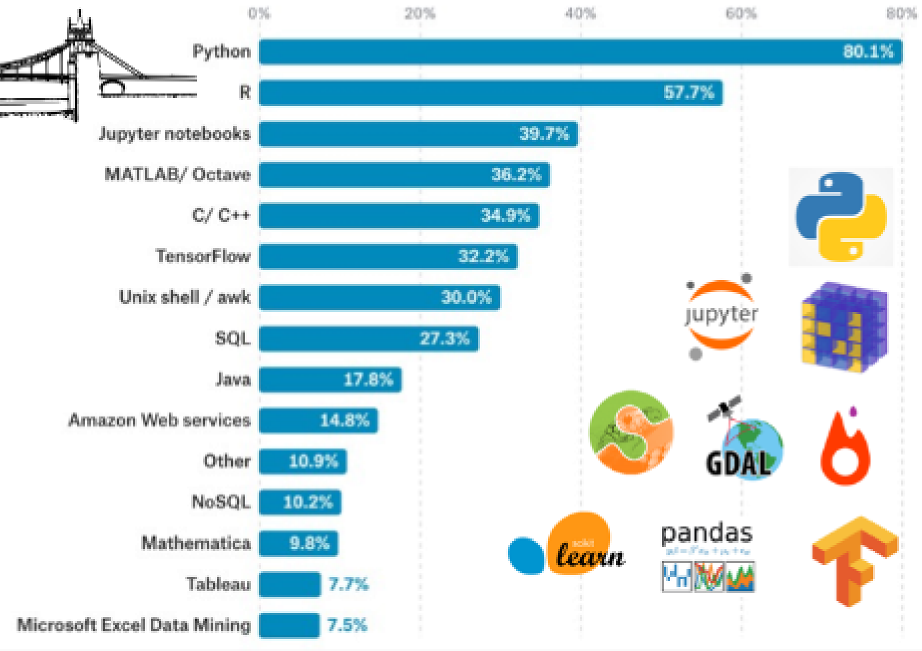
*Or google eo-learn*

# eo-learn

(EO) data

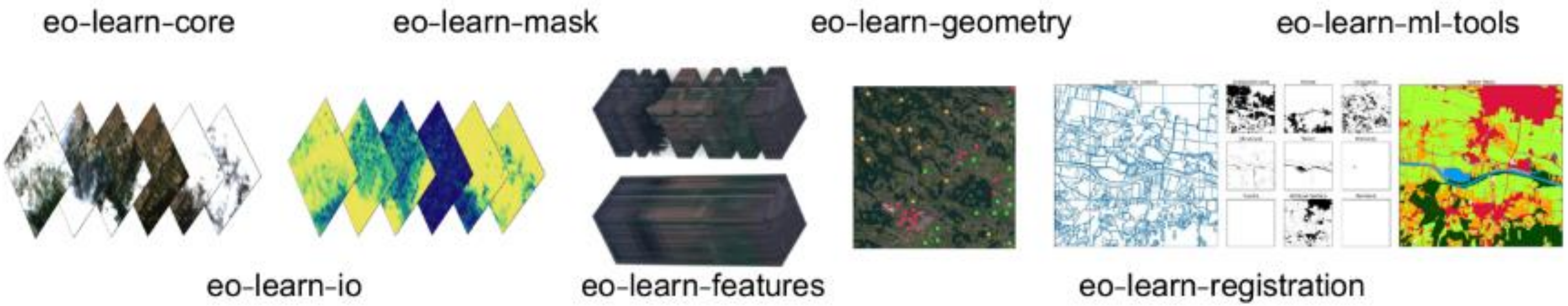


Python ecosystem
for data science and
machine learning



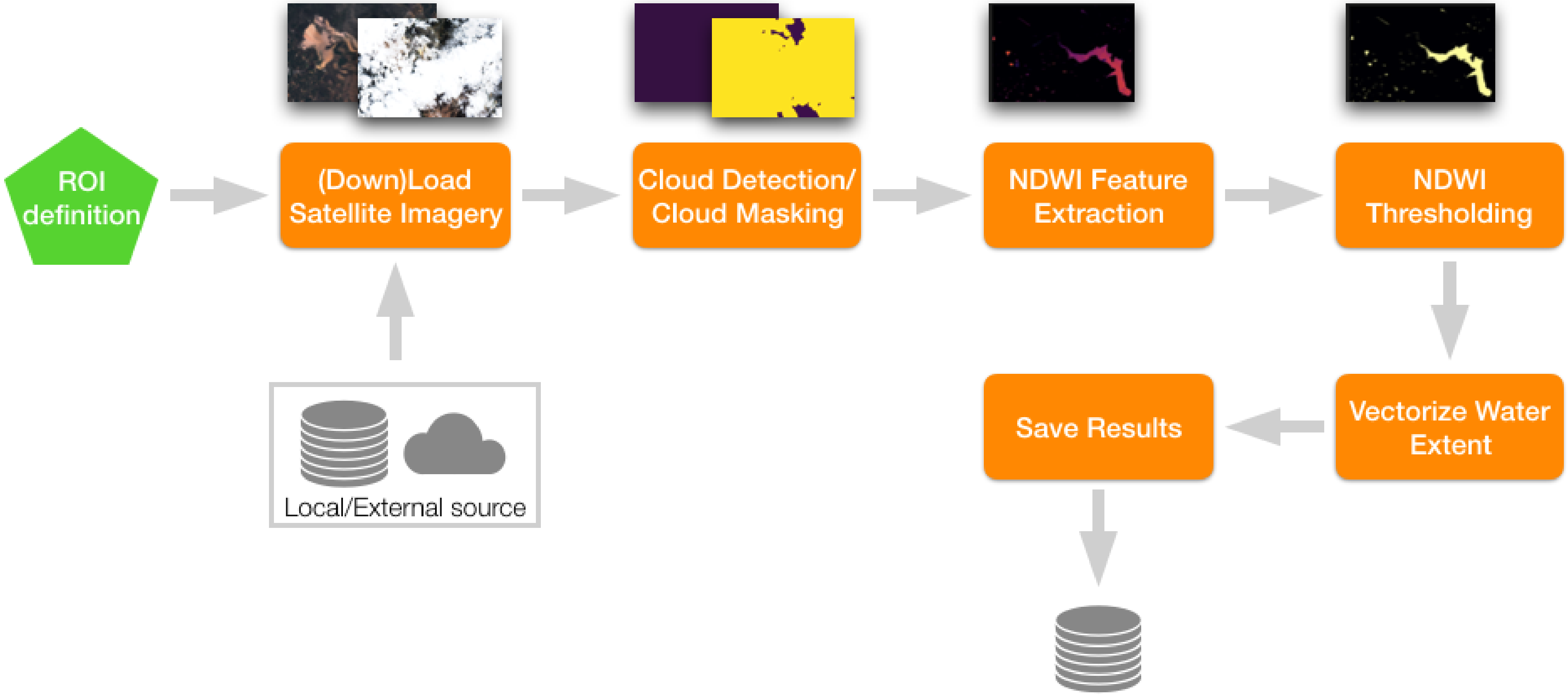
https://github.com/sentinel-hub/eo-learn

# eo-learn



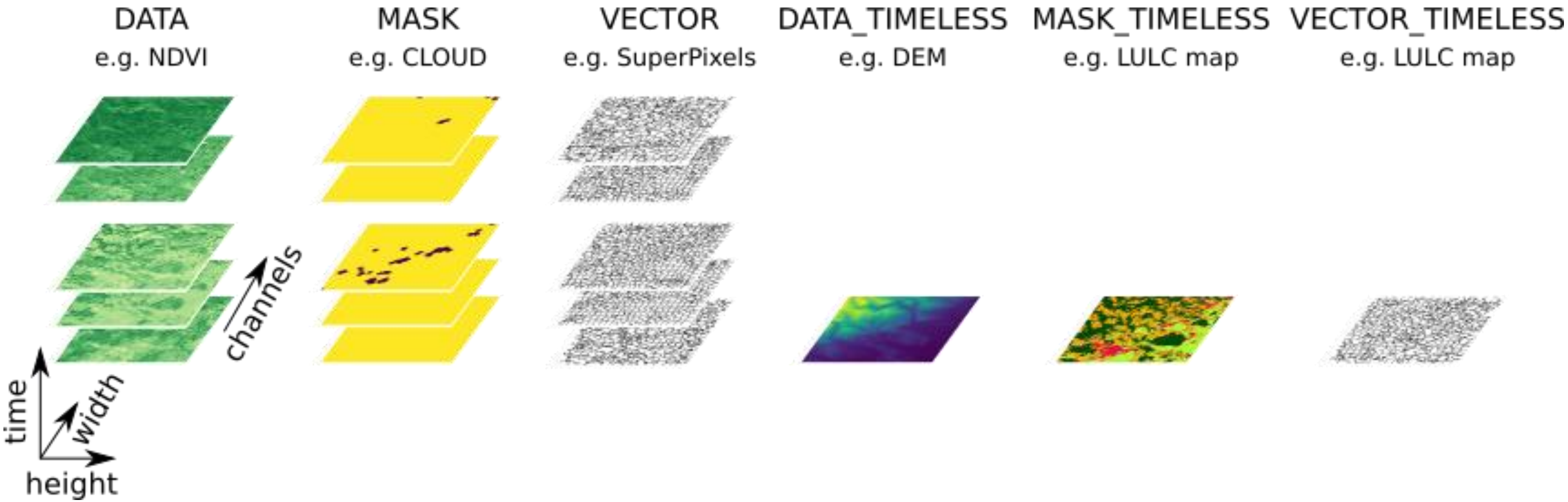
- **eo-learn-core** - The main subpackage which implements basic building blocks and commonly used functionalities.
- **eo-learn-coregistration** - The subpackage that deals with image co-registration.
- **eo-learn-features** - A collection of utilities for extracting data properties and feature manipulation.
- **eo-learn-geometry** - Geometry subpackage used for geometric transformation and conversion between vector and raster data.
- **eo-learn-io** - Input/output subpackage that deals with obtaining data from Sentinel Hub services or saving and loading data locally.
- **eo-learn-mask** - The subpackage used for masking of data and calculation of cloud masks.
- **eo-learn-ml-tools** - Various tools that can be used before or after the machine learning process.
- **eo-learn-visualization** - Visualization tools for core elements of eo-learn.

# eo-learn


ROI definition
(Down)Load Satellite Imagery
Cloud Detection/ Cloud Masking
NDWI Feature Extraction
NDWI Thresholding
Vectorize Water Extent
Save Results
Local/External source

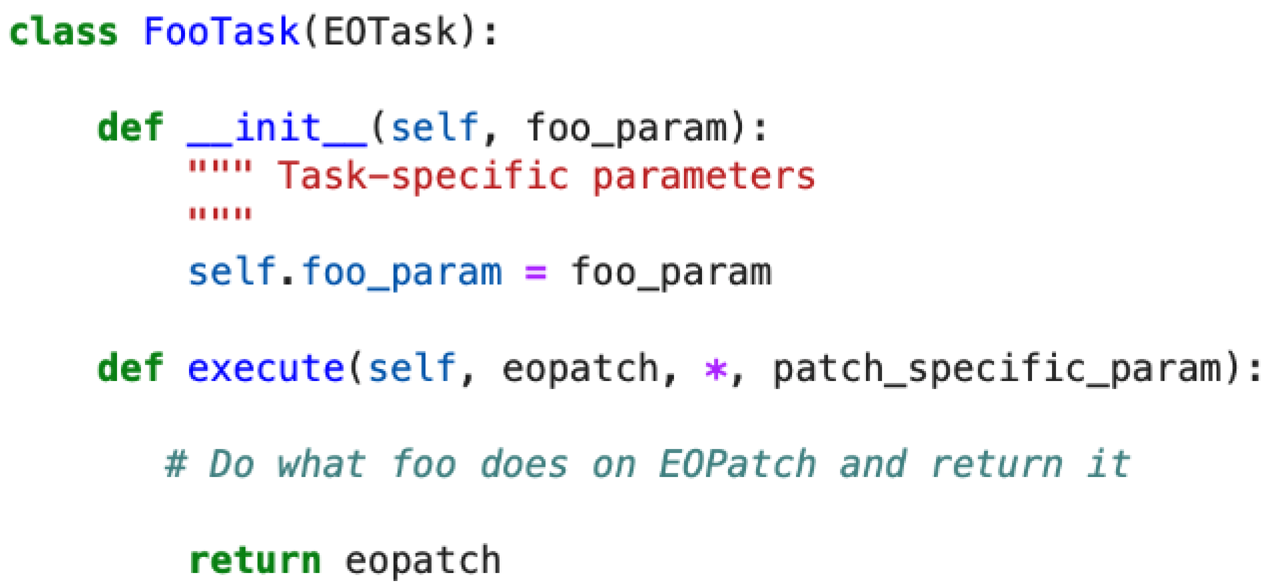
# eo-learn – building blocks

## EOPatch – storage



# eo-learn – building blocks

EOTask – a task/operation/function on (part of) EOPatch

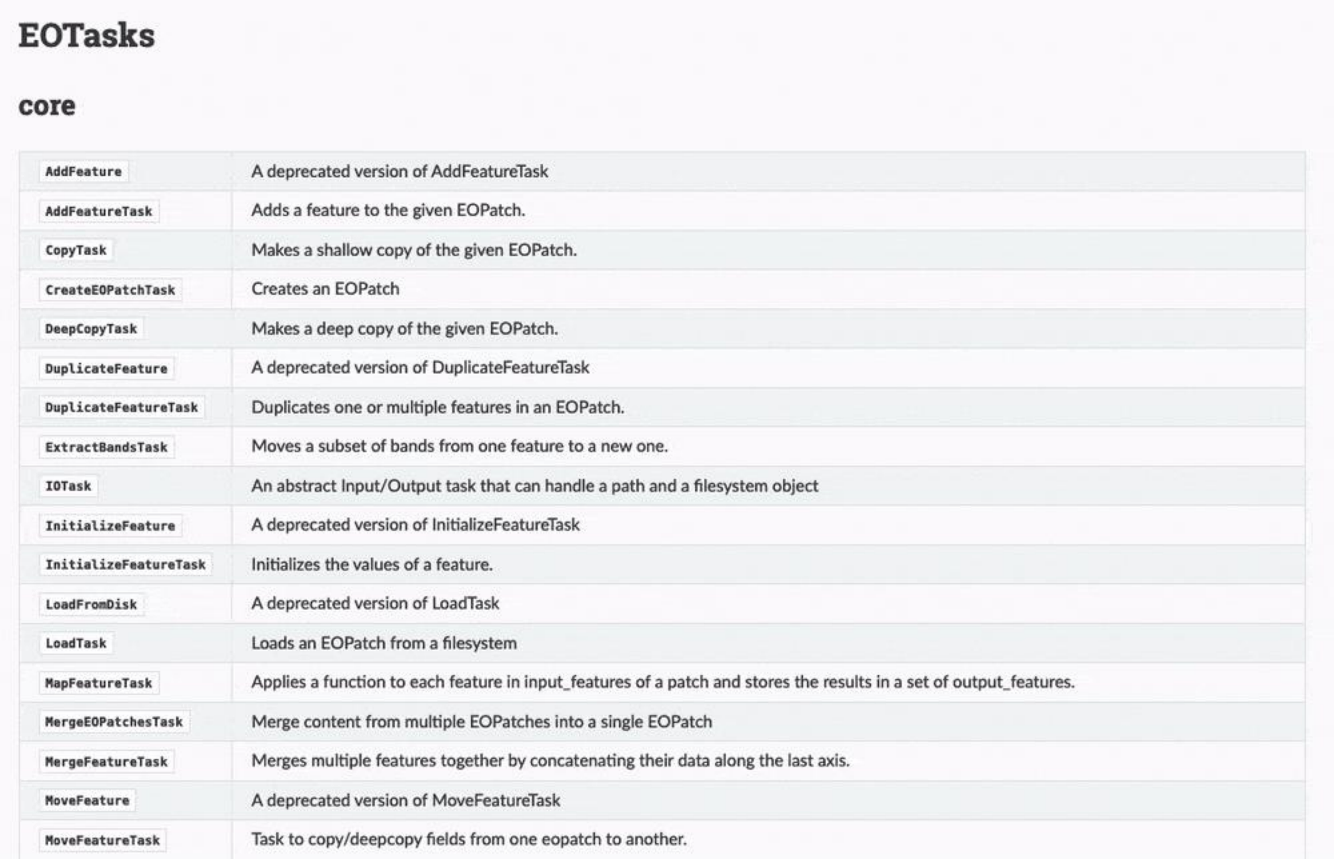


```
class FooTask(EOTask):

    def __init__(self, foo_param):
        """ Task-specific parameters
        """
        self.foo_param = foo_param

    def execute(self, eopatch, *, patch_specific_param):

       # Do what foo does on EOPatch and return it

        return eopatch
```

# eo-learn – building blocks

EOTask – a task/operation/function on (part of) EOPatch

https://eo-learn.readthedocs.io/en/latest/eotasks.html



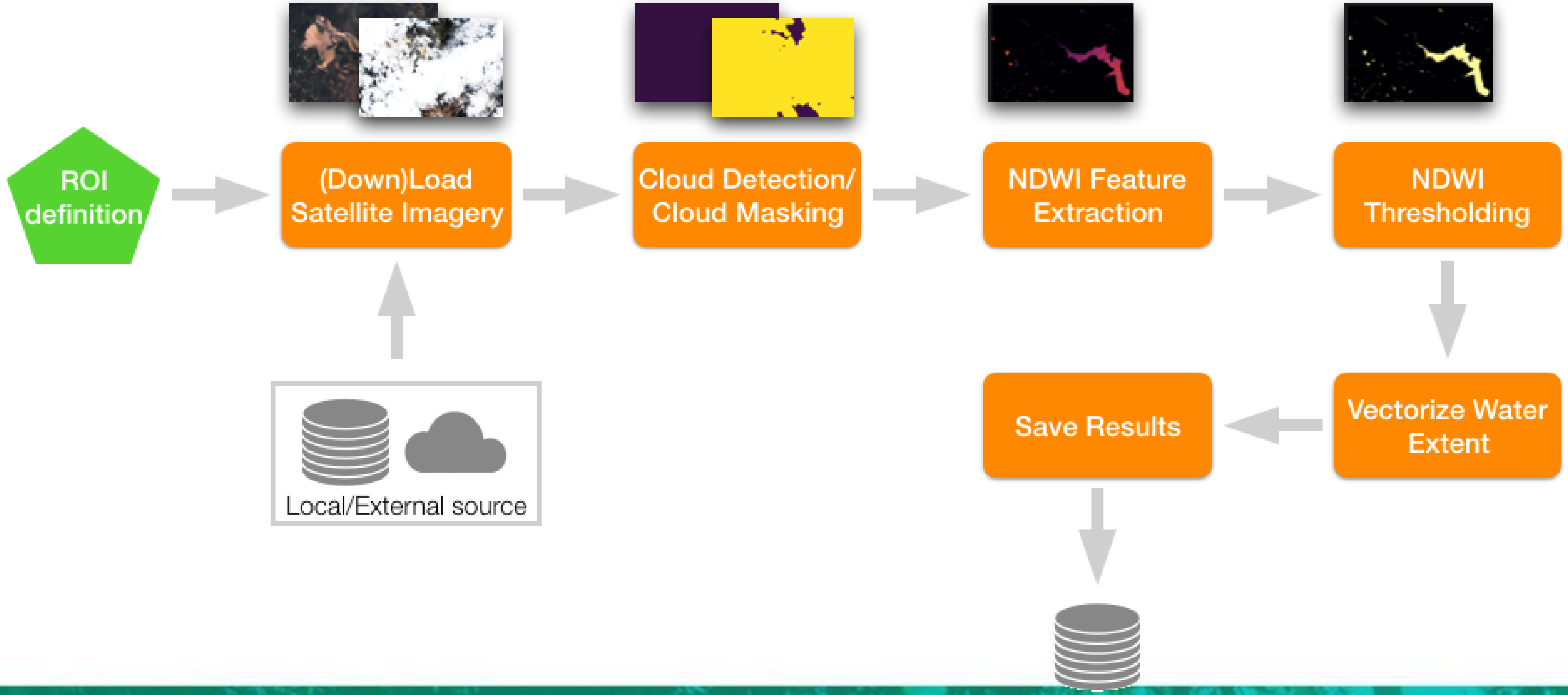
## EOTasks

### core

| | |
|---|---|
| `AddFeature` | A deprecated version of AddFeatureTask |
| `AddFeatureTask` | Adds a feature to the given EOPatch. |
| `CopyTask` | Makes a shallow copy of the given EOPatch. |
| `CreateEOPatchTask` | Creates an EOPatch |
| `DeepCopyTask` | Makes a deep copy of the given EOPatch. |
| `DuplicateFeature` | A deprecated version of DuplicateFeatureTask |
| `DuplicateFeatureTask` | Duplicates one or multiple features in an EOPatch. |
| `ExtractBandsTask` | Moves a subset of bands from one feature to a new one. |
| `IOTask` | An abstract Input/Output task that can handle a path and a filesystem object |
| `InitializeFeature` | A deprecated version of InitializeFeatureTask |
| `InitializeFeatureTask` | Initializes the values of a feature. |
| `LoadFromDisk` | A deprecated version of LoadTask |
| `LoadTask` | Loads an EOPatch from a filesystem |
| `MapFeatureTask` | Applies a function to each feature in input_features of a patch and stores the results in a set of output_features. |
| `MergeEOPatchesTask` | Merge content from multiple EOPatches into a single EOPatch |
| `MergeFeatureTask` | Merges multiple features together by concatenating their data along the last axis. |
| `MoveFeature` | A deprecated version of MoveFeatureTask |
| `MoveFeatureTask` | Task to copy/deepcopy fields from one eopatch to another. |

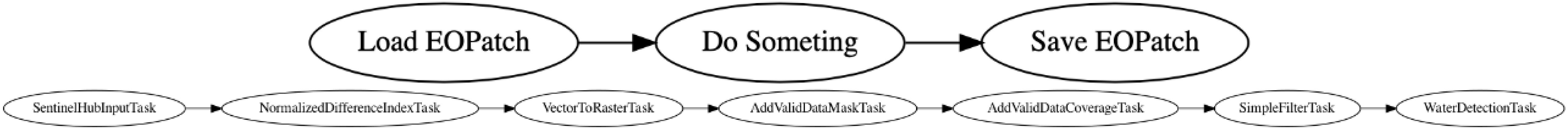
# eo-learn – building blocks

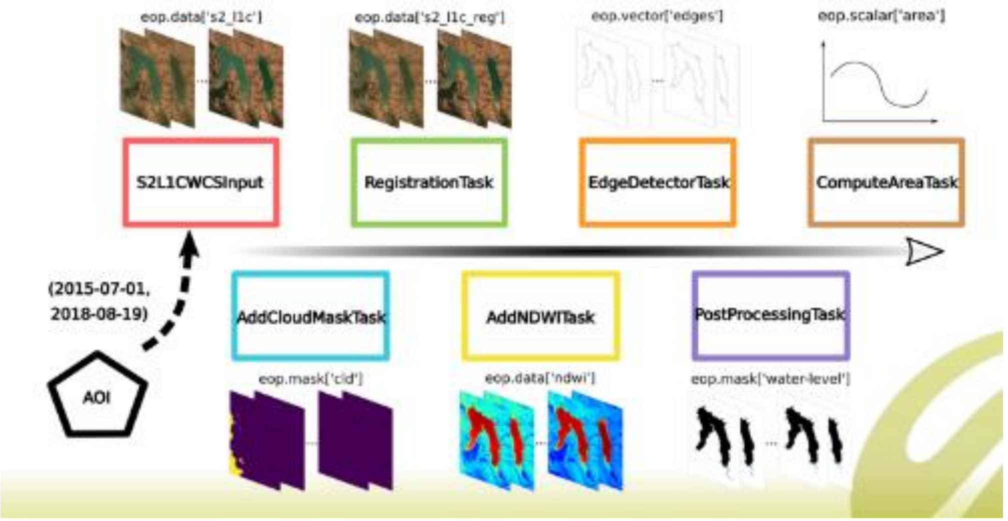
## EOWorkflow

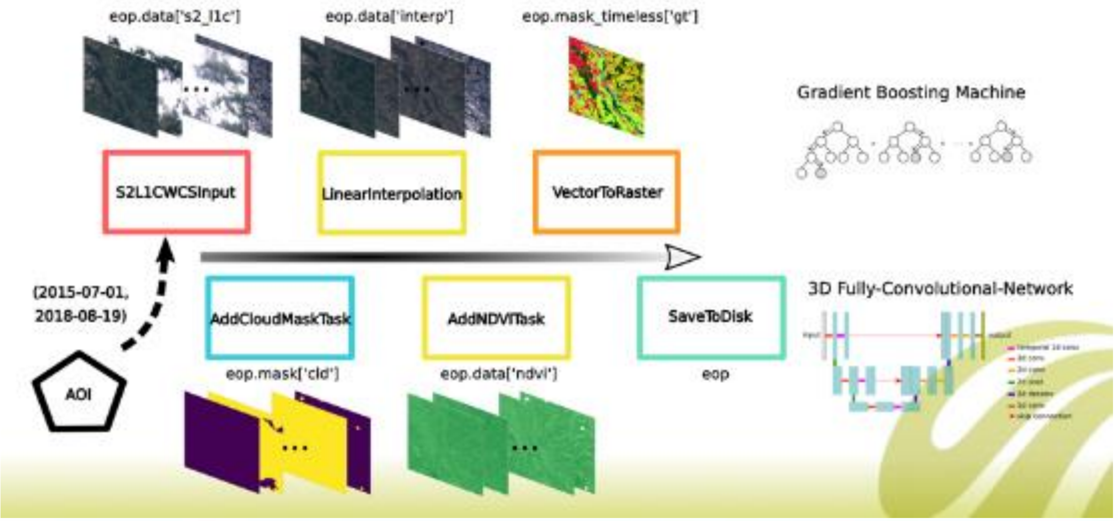


# eo-learn – building blocks

## EOWorkflow – an acyclic graph of EOTasks

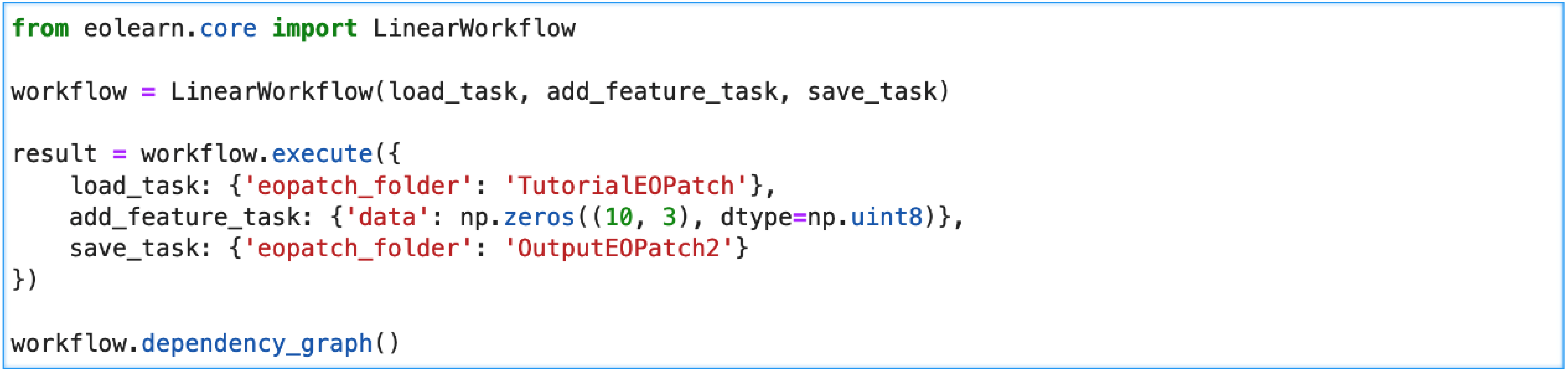




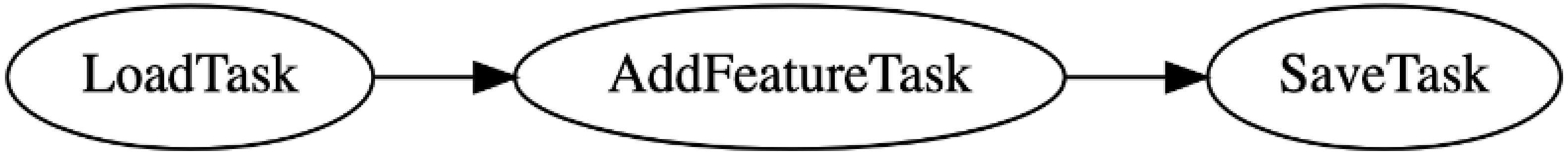


# eo-learn – building blocks

## EOWorkflow – an acyclic graph of EOTasks



```
from eolearn.core import LinearWorkflow

workflow = LinearWorkflow(load_task, add_feature_task, save_task)

result = workflow.execute({
    load_task: {'eopatch_folder': 'TutorialEOPatch'},
    add_feature_task: {'data': np.zeros((10, 3), dtype=np.uint8)},
    save_task: {'eopatch_folder': 'OutputEOPatch2'}
})

workflow.dependency_graph()
```



# eo-learn – building blocks

## EOWorkflow

Particularly useful:

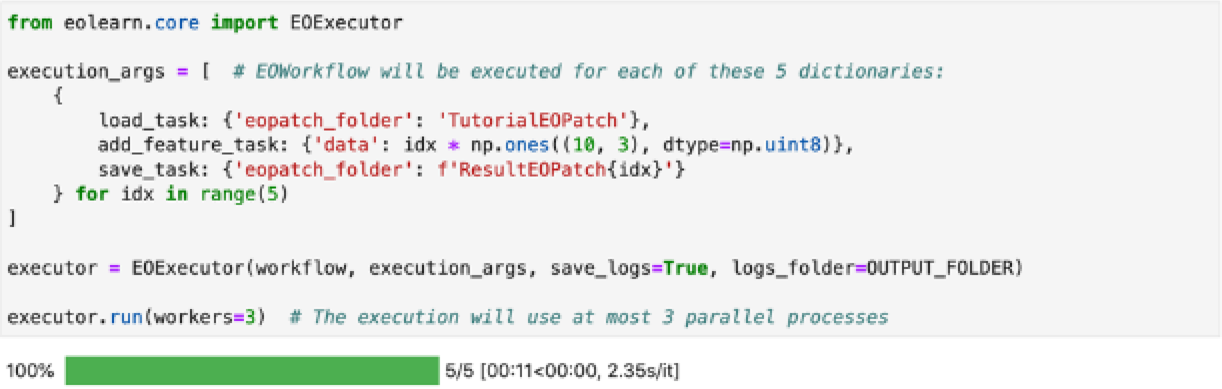
- when the (data) exploratory phase is finished
- things have to be run on (larger) scale
- it is time to automatise the “process”

Very powerful in conjunction with EOExecutor

# eo-learn – building blocks

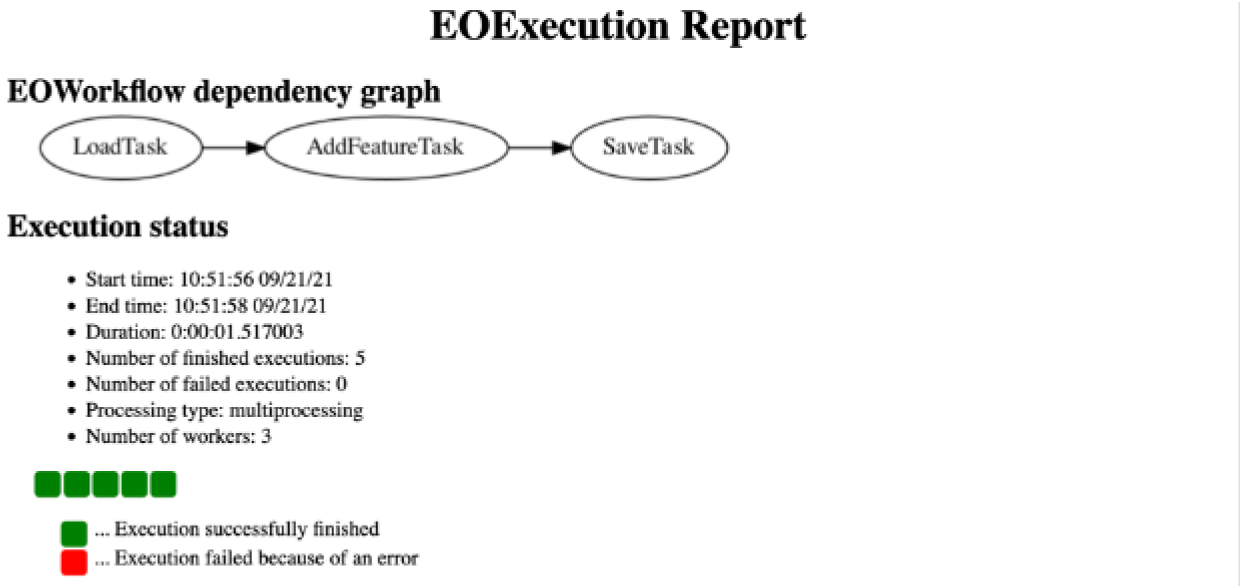
## EOExecutor

- handles execution and monitoring of workflows
- enables execution of a workflow multiple times and in parallel
- monitors execution times and handles/logs errors that might occur in the process
- generates a report which contains summary of the workflow and process of execution.



```
from eolearn.core import EOExecutor

execution_args = [  # EOWorkflow will be executed for each of these 5 dictionaries:
    {
        load_task: {'eopatch_folder': 'TutorialEOPatch'},
        add_feature_task: {'data': idx * np.ones((10, 3), dtype=np.uint8)},
        save_task: {'eopatch_folder': f'ResultEOPatch{idx}'}
    } for idx in range(5)
]

executor = EOExecutor(workflow, execution_args, save_logs=True, logs_folder=OUTPUT_FOLDER)

executor.run(workers=3)  # The execution will use at most 3 parallel processes
```

```
100% 5/5 [00:11<00:00, 2.35s/it]
```



### EOExecution Report

#### EOWorkflow dependency graph

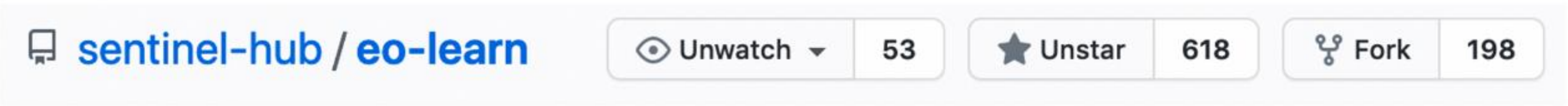
#### Execution status

- Start time: 10:51:56 09/21/21
- End time: 10:51:58 09/21/21
- Duration: 0:00:01.517003
- Number of finished executions: 5
- Number of failed executions: 0
- Processing type: multiprocessing
- Number of workers: 3

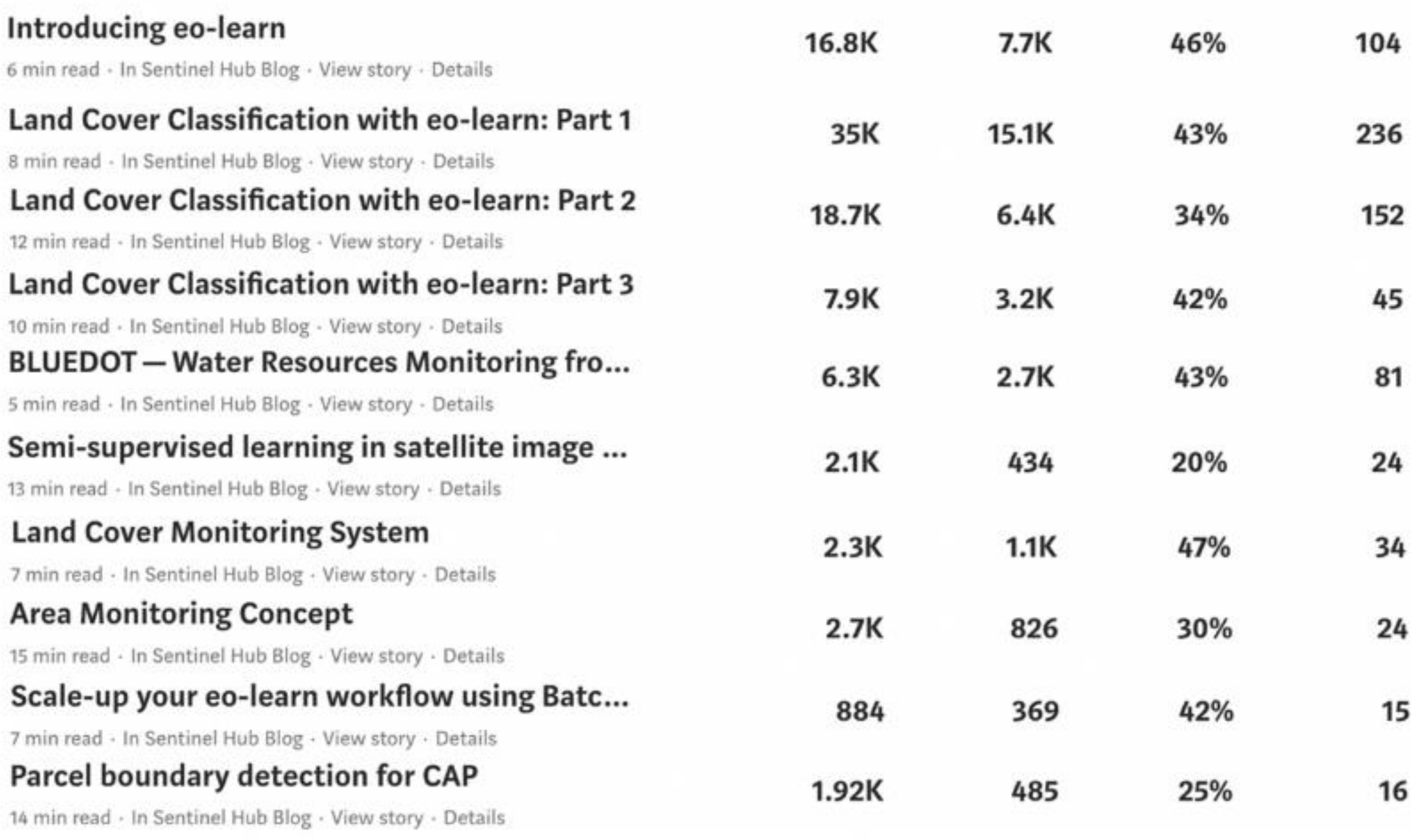
🟩🟩🟩🟩🟩

🟩 ... Execution successfully finished
🟥 ... Execution failed because of an error

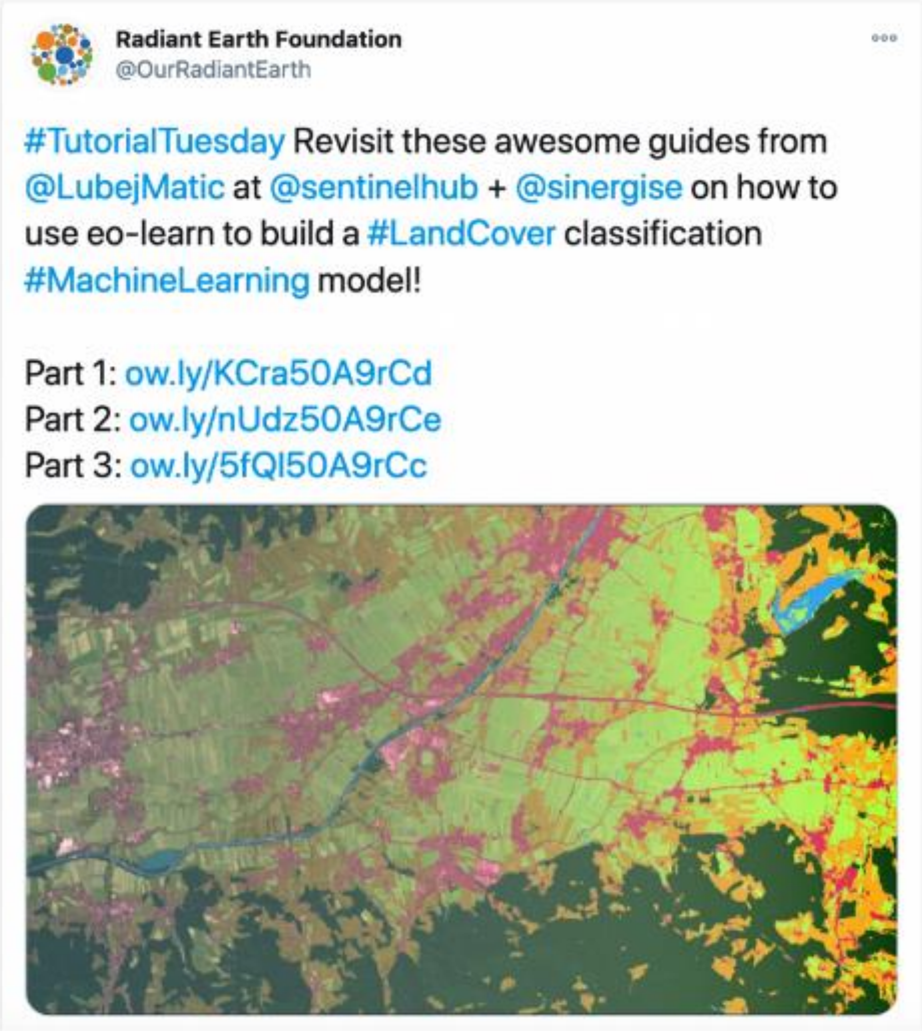
# eo-learn



sentinel-hub / eo-learn



| | | | | |
|---|---|---|---|---|
| **Introducing eo-learn**<br>6 min read · In Sentinel Hub Blog · View story · Details | 16.8K | 7.7K | 46% | 104 |
| **Land Cover Classification with eo-learn: Part 1**<br>8 min read · In Sentinel Hub Blog · View story · Details | 35K | 15.1K | 43% | 236 |
| **Land Cover Classification with eo-learn: Part 2**<br>12 min read · In Sentinel Hub Blog · View story · Details | 18.7K | 6.4K | 34% | 152 |
| **Land Cover Classification with eo-learn: Part 3**<br>10 min read · In Sentinel Hub Blog · View story · Details | 7.9K | 3.2K | 42% | 45 |
| **BLUEDOT — Water Resources Monitoring fro...**<br>5 min read · In Sentinel Hub Blog · View story · Details | 6.3K | 2.7K | 43% | 81 |
| **Semi-supervised learning in satellite image ...**<br>13 min read · In Sentinel Hub Blog · View story · Details | 2.1K | 434 | 20% | 24 |
| **Land Cover Monitoring System**<br>7 min read · In Sentinel Hub Blog · View story · Details | 2.3K | 1.1K | 47% | 34 |
| **Area Monitoring Concept**<br>15 min read · In Sentinel Hub Blog · View story · Details | 2.7K | 826 | 30% | 24 |
| **Scale-up your eo-learn workflow using Batc...**<br>7 min read · In Sentinel Hub Blog · View story · Details | 884 | 369 | 42% | 15 |
| **Parcel boundary detection for CAP**<br>14 min read · In Sentinel Hub Blog · View story · Details | 1.92K | 485 | 25% | 16 |



Radiant Earth Foundation
@OurRadiantEarth

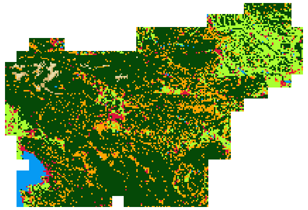
#TutorialTuesday Revisit these awesome guides from @LubejMatic at @sentinelhub + @sinergise on how to use eo-learn to build a #LandCover classification #MachineLearning model!

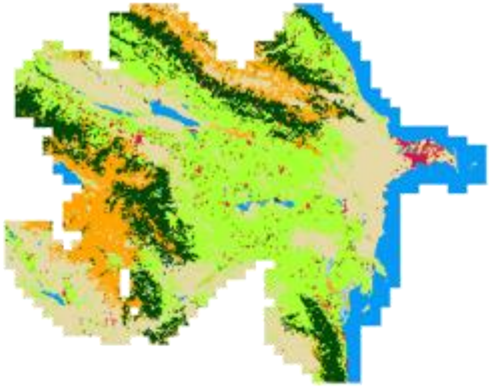
Part 1: ow.ly/KCra50A9rCd
Part 2: ow.ly/nUdz50A9rCe
Part 3: ow.ly/5fQI50A9rCc

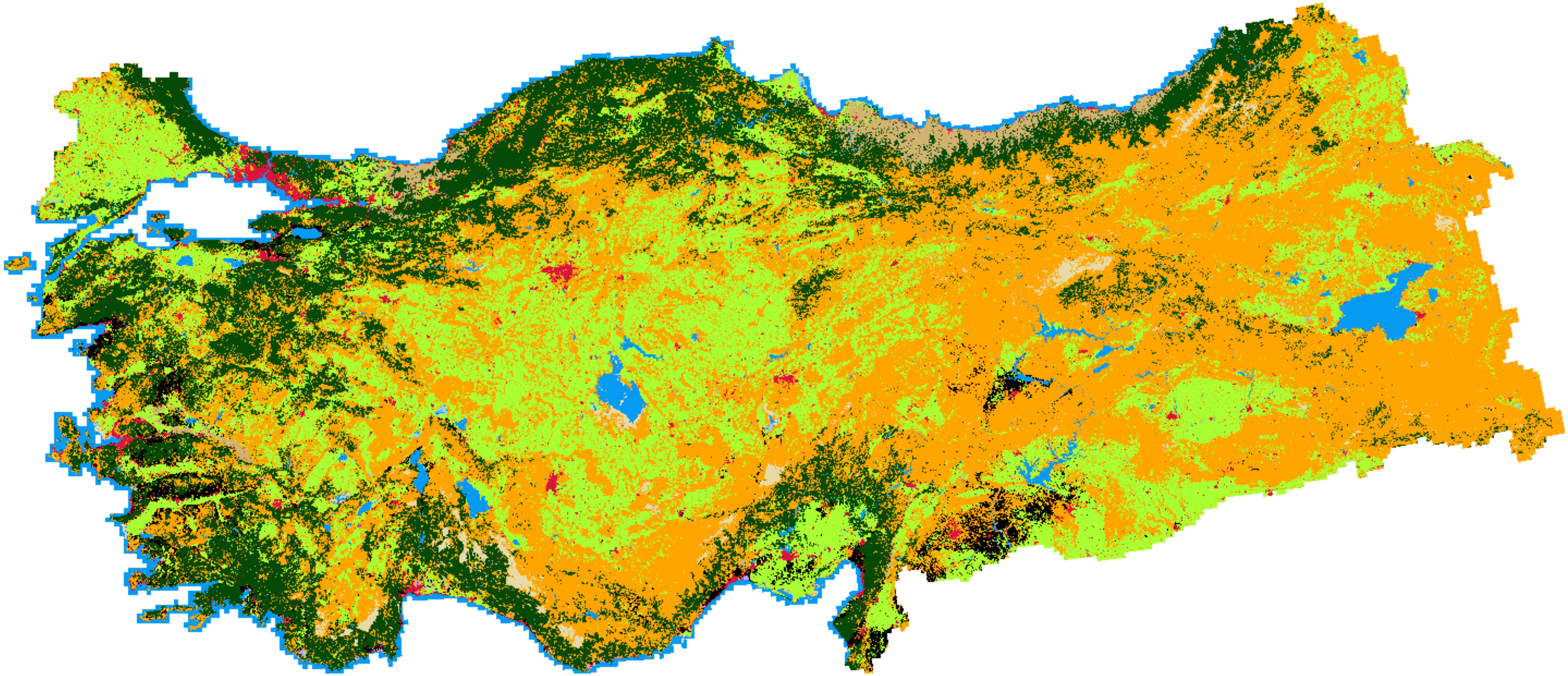
https://medium.com/sentinel-hub

# eo-learn – ”eat your own (dog) food”

## Land Cover and Land Cover Monitoring

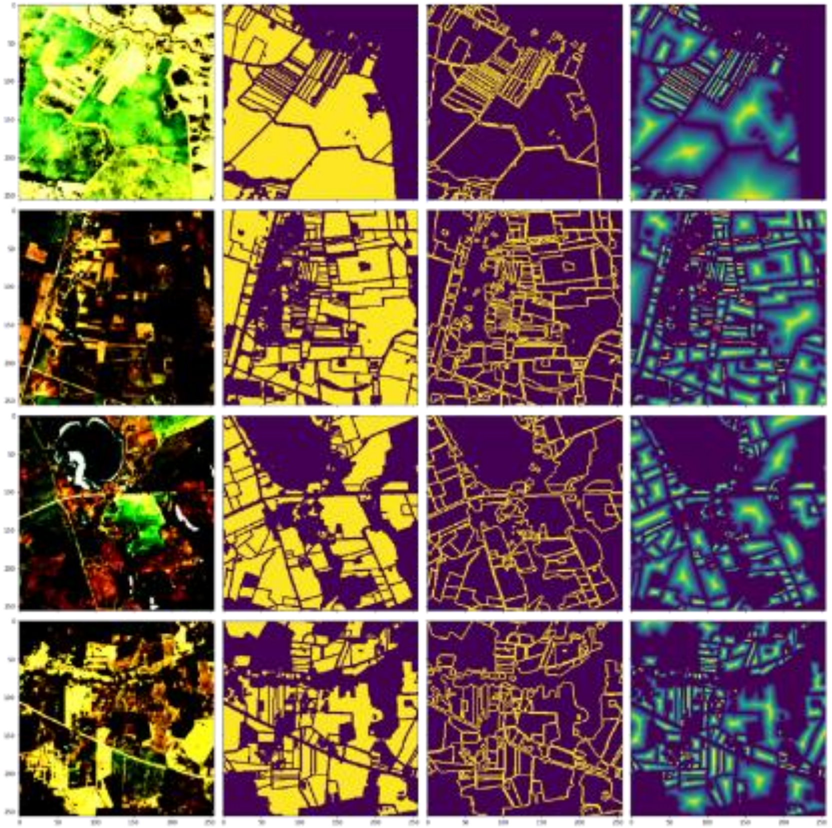


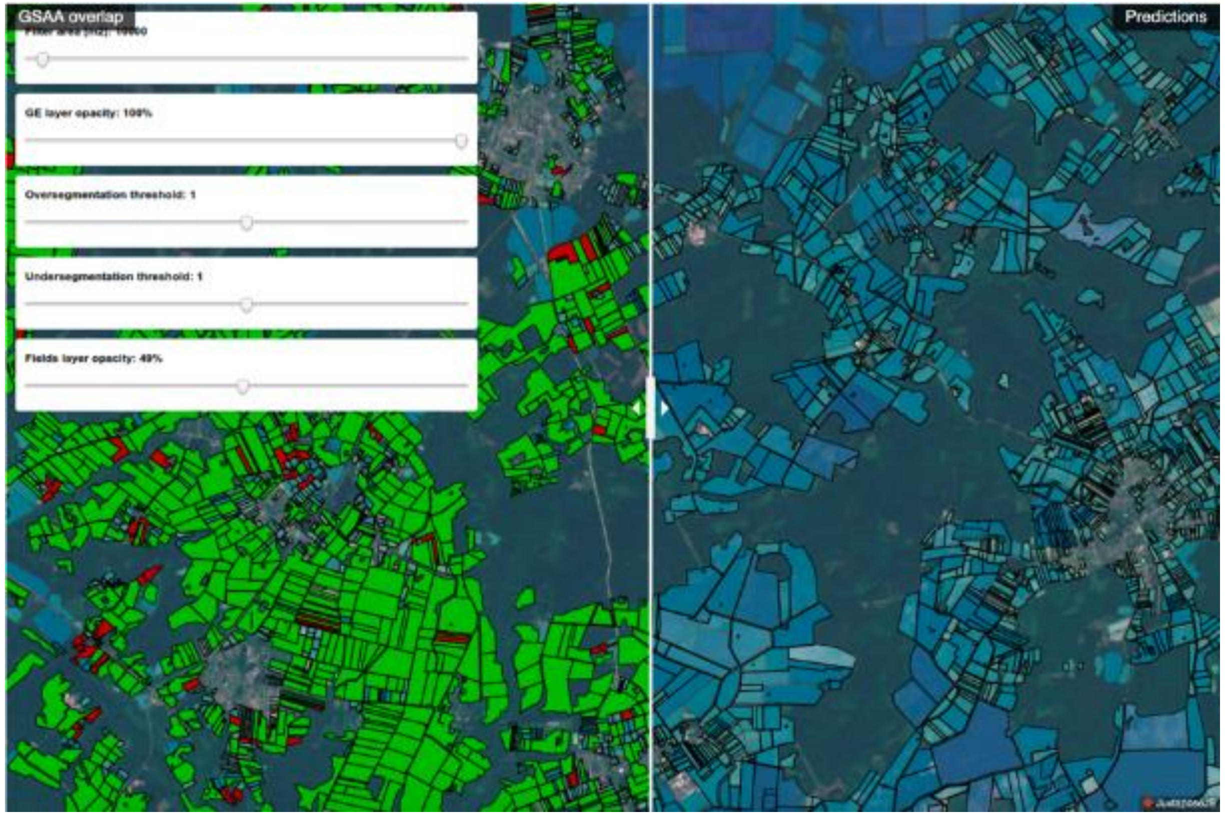




# eo-learn – ”eat your own (dog) food”

## Agricultural field delineation with Sentinel-2





Filter area (m2): 10000

GE layer opacity: 100%

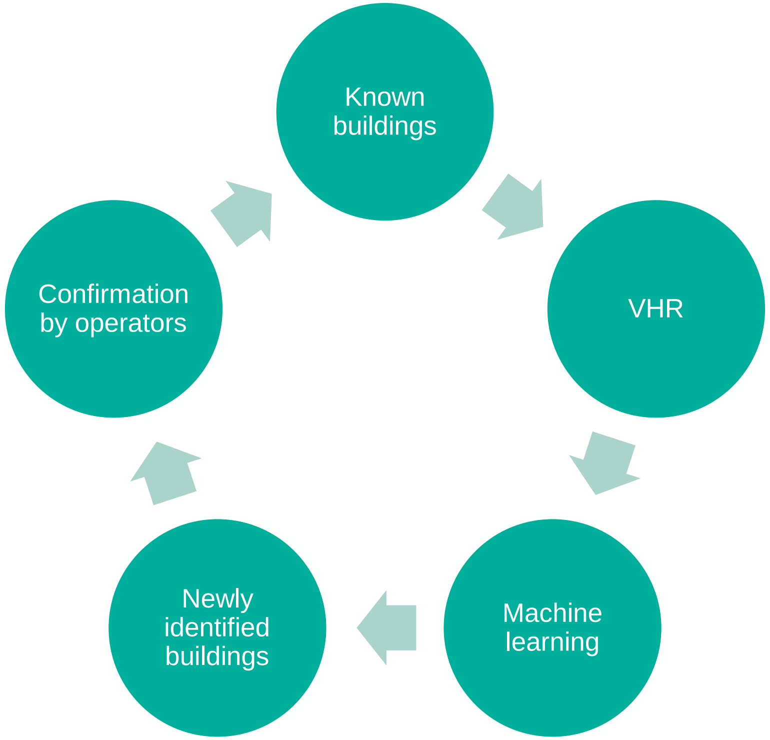
Oversegmentation threshold: 1

Undersegmentation threshold: 1

Fields layer opacity: 49%

# eo-learn – ”eat your own (dog) food”

## Building detection using SPOT imagery



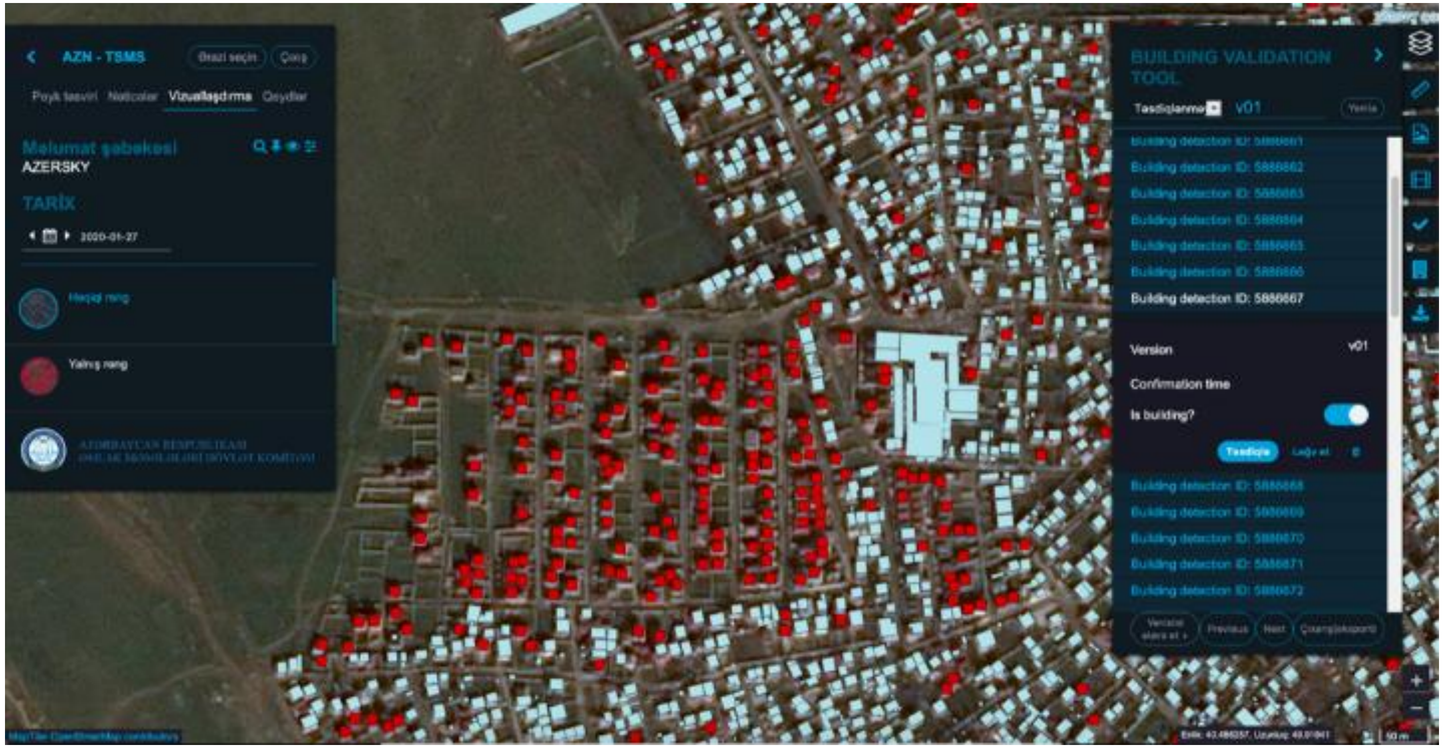
Known buildings

VHR

Machine learning

Newly identified buildings

Confirmation by operators



AZN - TSMS

Ərazi seçin
Çıxış

Peyk təsviri
Nəticələr
Vizuallaşdırma
Qeydlər

Məlumat şəbəkəsi

AZERSKY

TARİX

2020-01-27

Həqiqi rəng

Yalnız rəng

AZƏRBAYCAN RESPUBLİKASI
ƏMLAK MƏSƏLƏLƏRİ DÖVLƏT KOMİTƏSİ

BUILDING VALIDATION TOOL

Təsdiqlənmə v01 Yenilə

Building detection ID: 5888661
Building detection ID: 5888662
Building detection ID: 5888663
Building detection ID: 5888664
Building detection ID: 5888665
Building detection ID: 5888666
Building detection ID: 5888667

Version v01

Confirmation time

Is building?

Təsdiqlə
Ləğv et

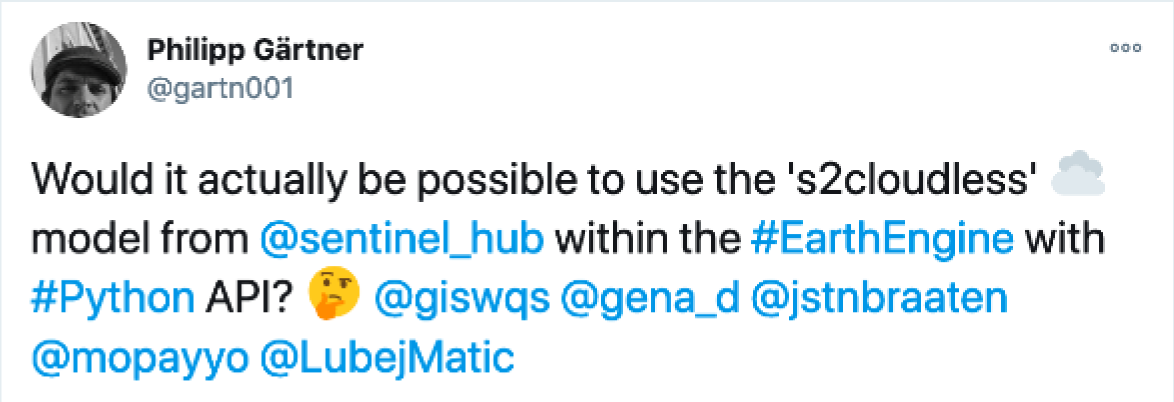
Building detection ID: 5888668
Building detection ID: 5888669
Building detection ID: 5888670
Building detection ID: 5888671
Building detection ID: 5888672

Versiləri əlavə et +
Previous
Next
Çoxqatlı eksport

Enlik: 43.486257, Uzunluq: 49.01841

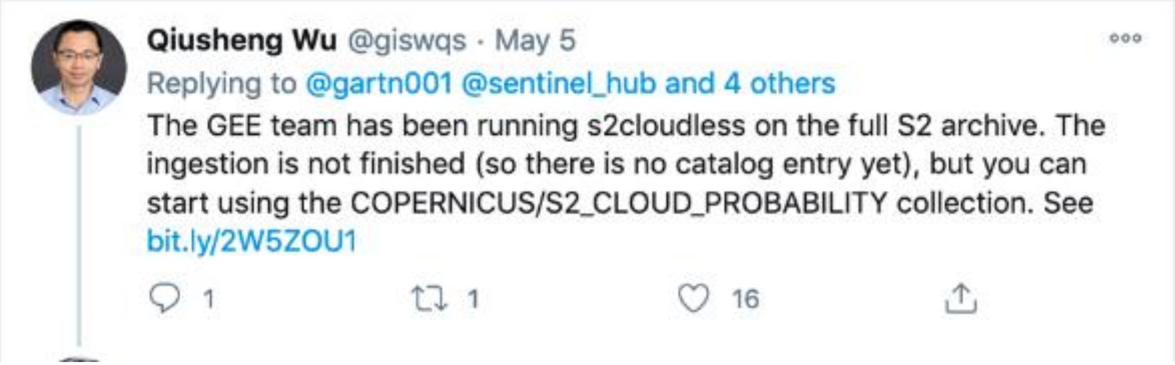
50 m

# eo-learn – success stories from others



**Philipp Gärtner** @gartn001

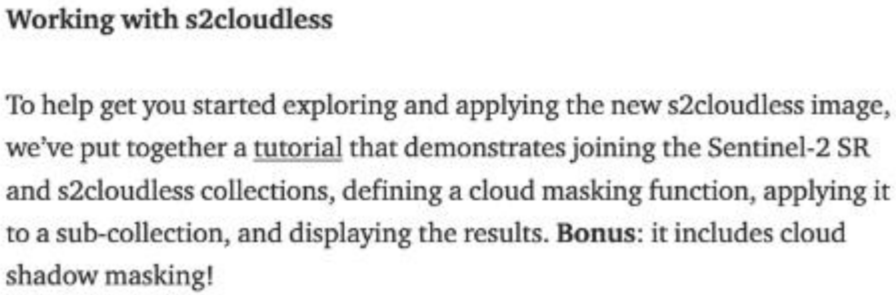
Would it actually be possible to use the 's2cloudless' ☁ model from @sentinel_hub within the #EarthEngine with #Python API? 🤔 @giswqs @gena_d @jstnbraaten @mopayyo @LubejMatic



**Qiusheng Wu** @giswqs · May 5

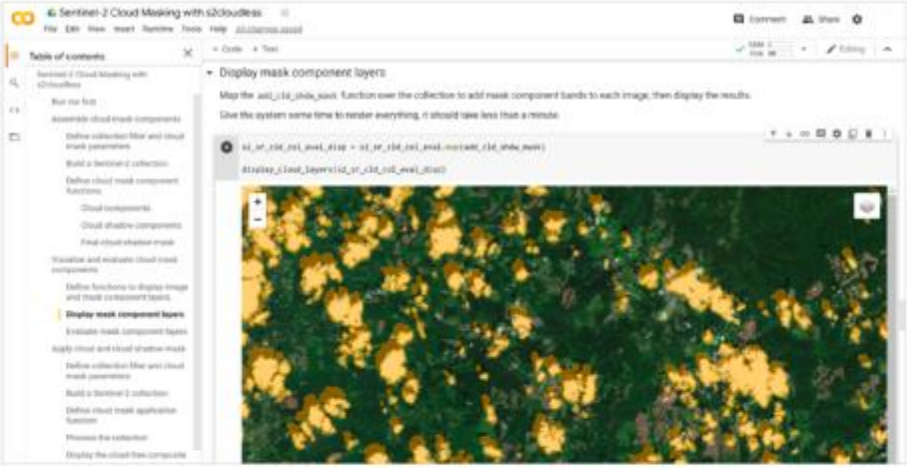
Replying to @gartn001 @sentinel_hub and 4 others

The GEE team has been running s2cloudless on the full S2 archive. The ingestion is not finished (so there is no catalog entry yet), but you can start using the COPERNICUS/S2_CLOUD_PROBABILITY collection. See bit.ly/2W5ZOU1



**Working with s2cloudless**

To help get you started exploring and applying the new s2cloudless image, we've put together a tutorial that demonstrates joining the Sentinel-2 SR and s2cloudless collections, defining a cloud masking function, applying it to a sub-collection, and displaying the results. **Bonus**: it includes cloud shadow masking!



Sentinel-2 Cloud Masking with s2cloudless

Display mask component layers

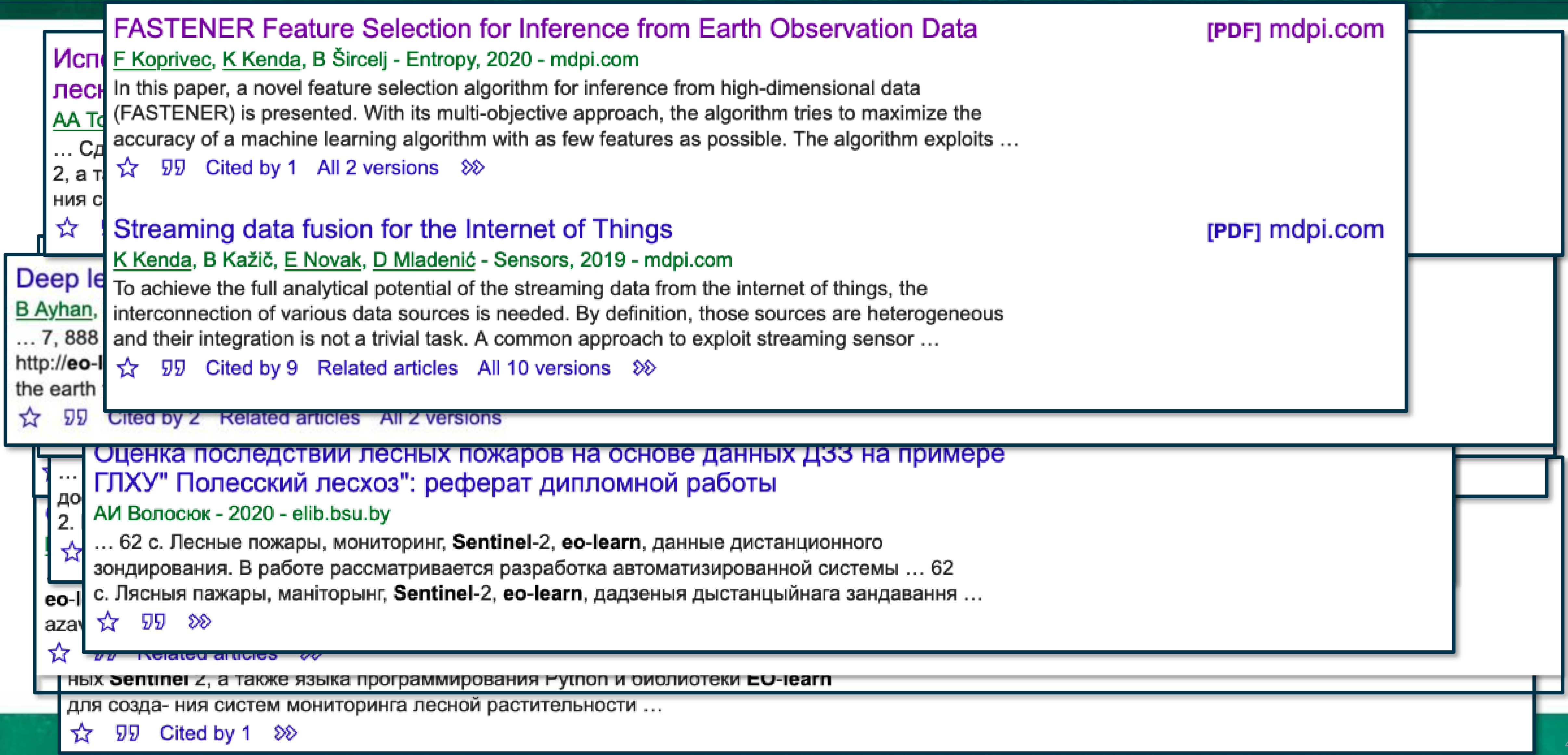
Map the add_cld_shdw_mask function over the collection to add mask component bands to each image, then display the results.

Give the system some time to render everything, it should take less than a minute.

```
s2_sr_cld_col_eval_disp = s2_sr_cld_col_eval.map(add_cld_shdw_mask)
display_cloud_layers(s2_sr_cld_col_eval_disp)
```

Earth Engine Colab notebook on using the new s2cloudless image for cloud and cloud shadow masking Sentinel-2 imagery.

# eo-learn – success stories from others



**FASTENER Feature Selection for Inference from Earth Observation Data** — [PDF] mdpi.com

F Koprivec, K Kenda, B Šircelj - Entropy, 2020 - mdpi.com

In this paper, a novel feature selection algorithm for inference from high-dimensional data (FASTENER) is presented. With its multi-objective approach, the algorithm tries to maximize the accuracy of a machine learning algorithm with as few features as possible. The algorithm exploits …

☆ 99 Cited by 1 All 2 versions »

**Streaming data fusion for the Internet of Things** — [PDF] mdpi.com

K Kenda, B Kažič, E Novak, D Mladenić - Sensors, 2019 - mdpi.com

To achieve the full analytical potential of the streaming data from the internet of things, the interconnection of various data sources is needed. By definition, those sources are heterogeneous and their integration is not a trivial task. A common approach to exploit streaming sensor …

☆ 99 Cited by 9 Related articles All 10 versions »

**Оценка последствий лесных пожаров на основе данных ДЗЗ на примере ГЛХУ" Полесский лесхоз": реферат дипломной работы**

АИ Волосюк - 2020 - elib.bsu.by

… 62 с. Лесные пожары, мониторинг, **Sentinel**-2, **eo-learn**, данные дистанционного зондирования. В работе рассматривается разработка автоматизированной системы … 62 с. Лясныя пажары, маніторынг, **Sentinel**-2, **eo-learn**, дадзеныя дыстанцыйнага зандавання …

☆ 99 »

Исп… лесн…

AA To…

… Сд… 2, а т… ния с…

Deep le…

B Ayhan, …

… 7, 888 … http://**eo-l**… the earth …

☆ 99 Cited by 2 Related articles All 2 versions

… до… 2.

**eo-l**… aza…

☆ 99 Related articles »

ных **Sentinel** 2, а также языка программирования Python и библиотеки **EO-learn** для созда- ния систем мониторинга лесной растительности …

☆ 99 Cited by 1 »

# More information

- https://sentinel-hub.com/
- https://apps.sentinel-hub.com/eo-browser/
- https://sentinel-hub.github.io/custom-scripts/
- https://education.sentinel-hub.com

- https://github.com/sentinel-hub/eo-learn
- https://eo-learn.readthedocs.io/en/latest/
- https://medium.com/sentinel-hub

# Thank You

Matej Batič,
EO research @ Sinergise

website: https://www.sentinel-hub.com/
blog: https://medium.com/sentinel-hub
twitter: @matejbatic @sinergise @sentinel_hub