



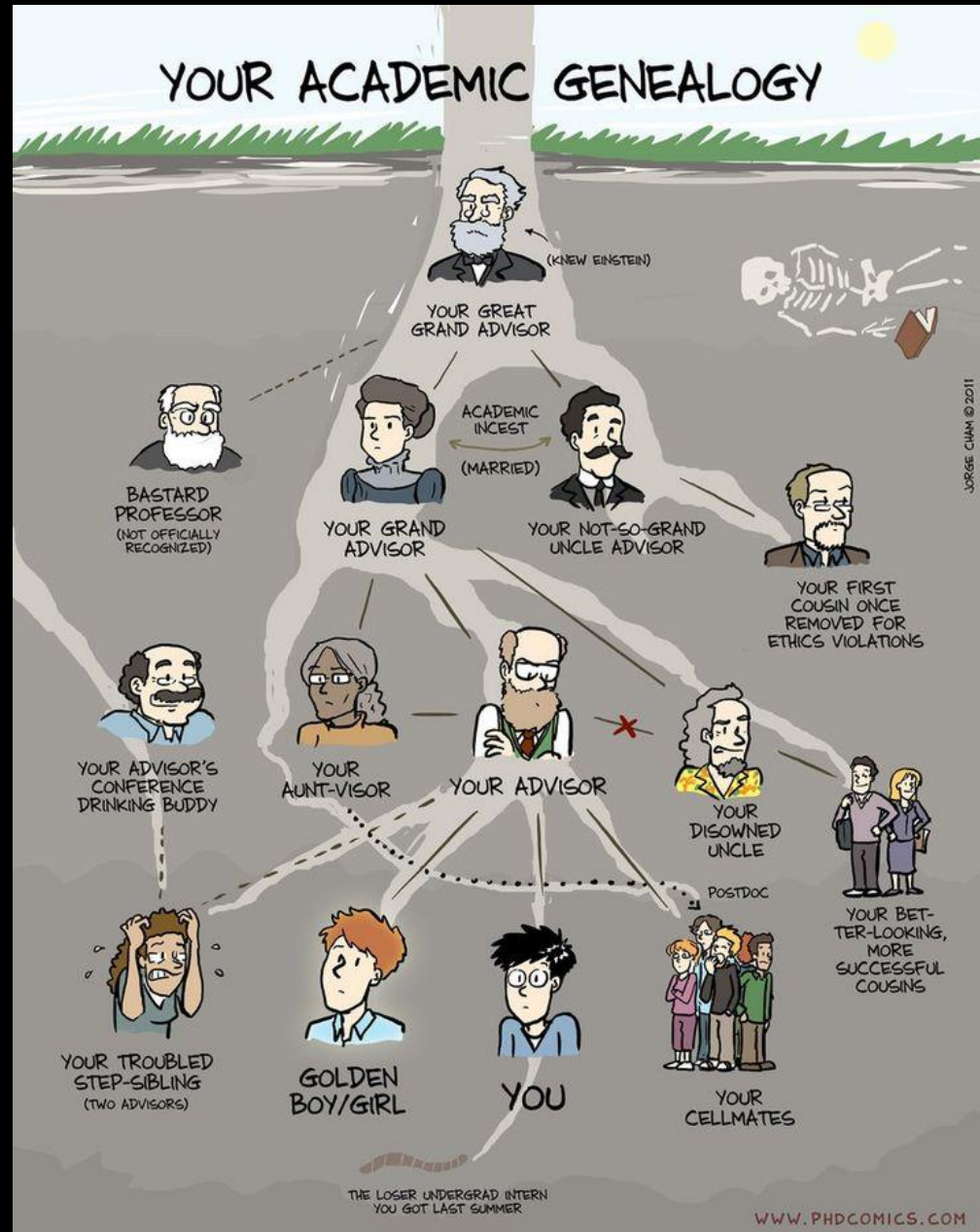
Universitatea
Transilvania
din Brașov

Landsat time series in forest research

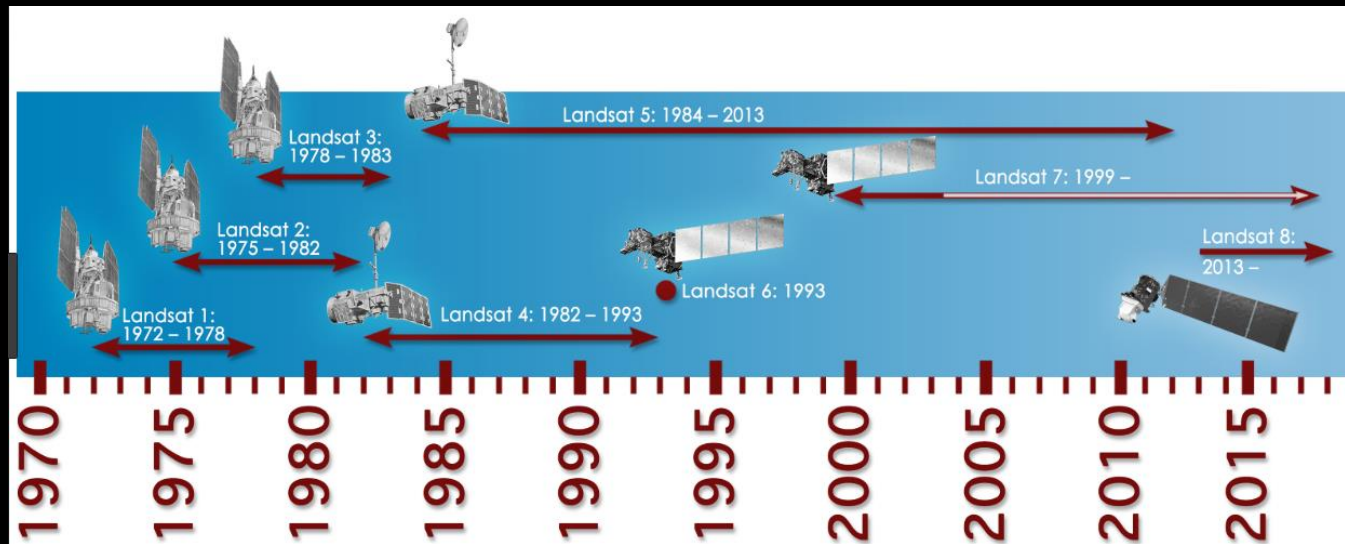
Mihai Nita

Overview

- Brief introduction in Google Earth Engine
- Few words on image classification using QGIS plugins
- Few insights on pixel long-term analysis



Long term forest monitoring



COPERNICUS AND ITS SENTINELS

European Earth Observation Programme Copernicus: observing our planet for a safer world

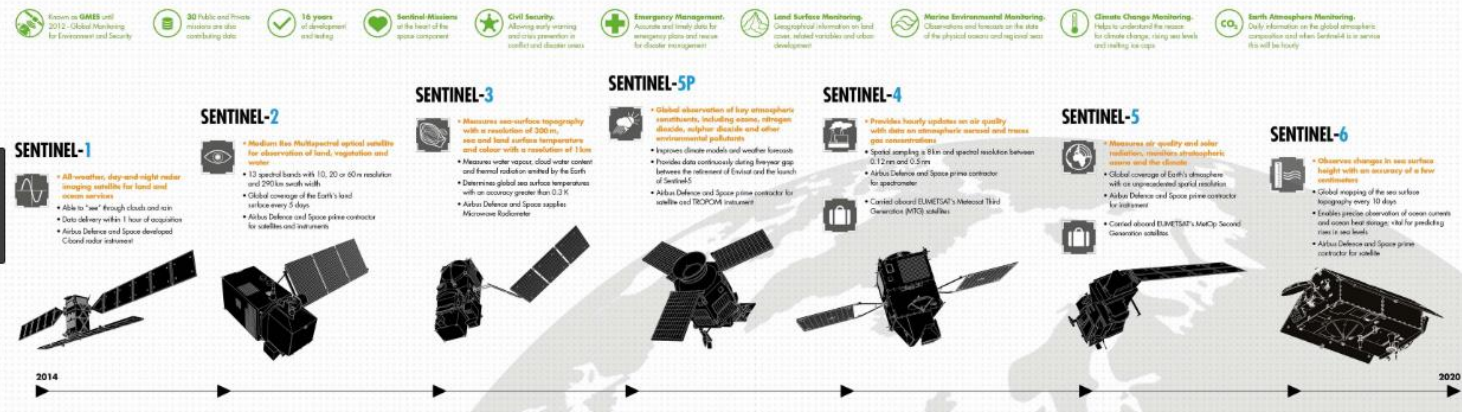
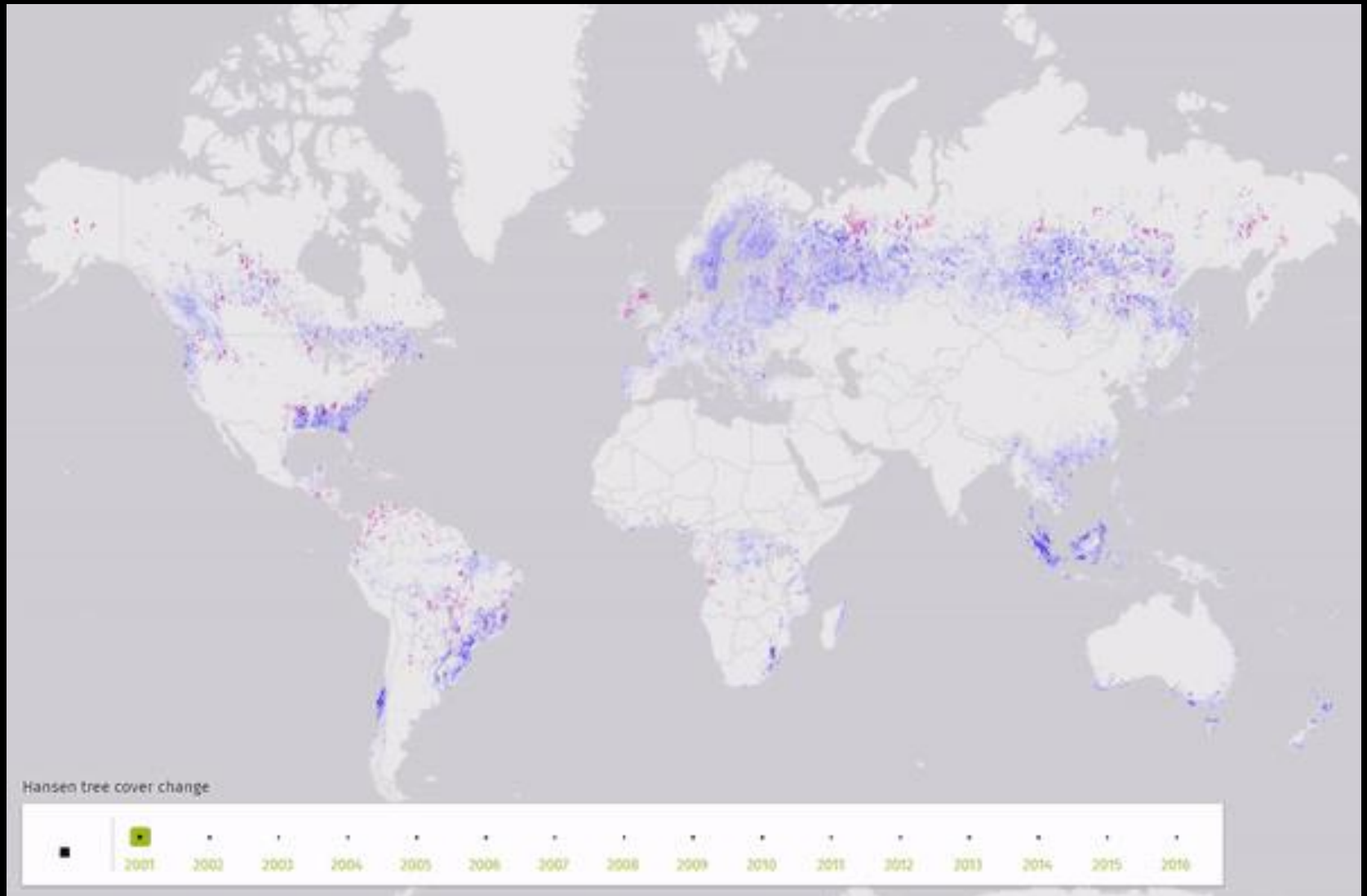
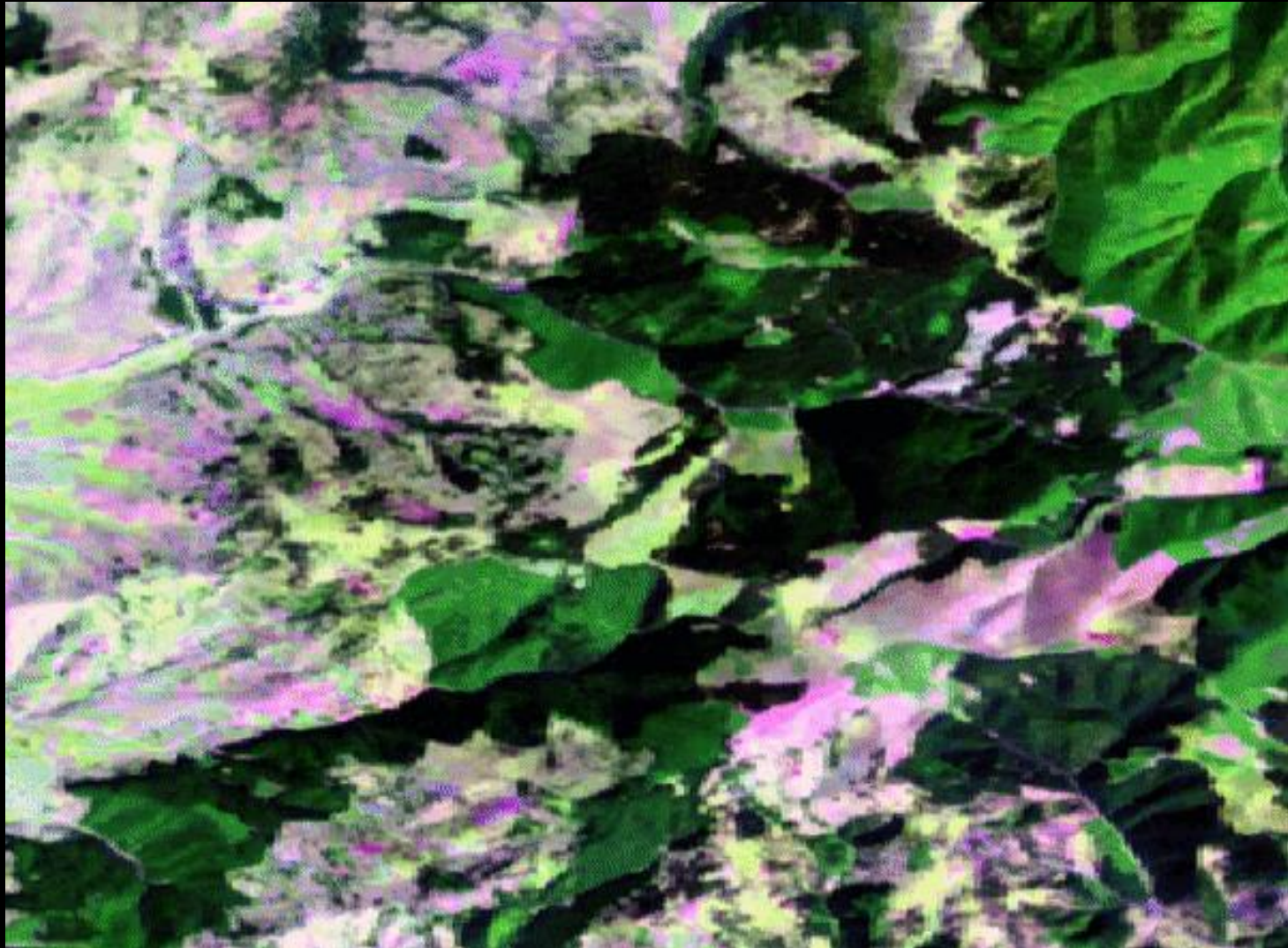


Photo credit: NASA and Airbus

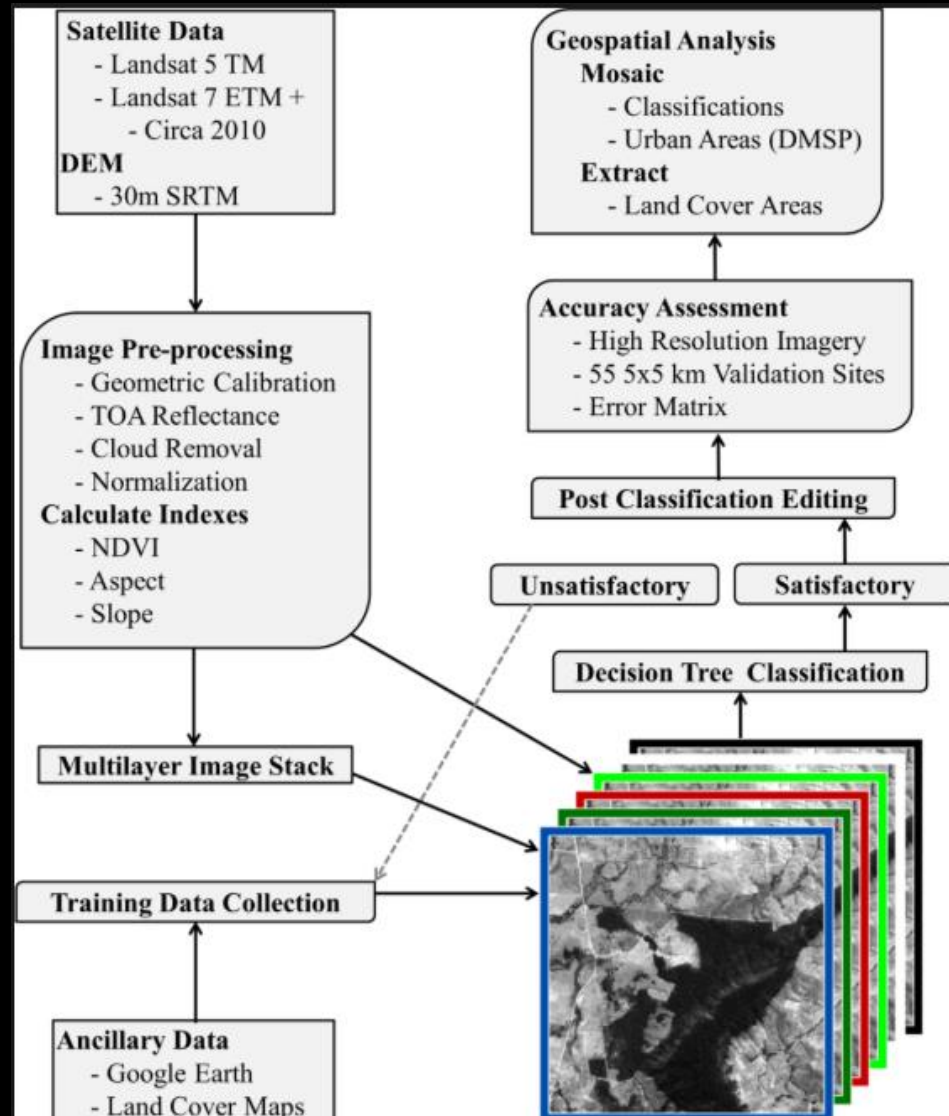
Landsat archive in forest research











Landsat archive in forest research



Challenge – processing – classification



Challenge - Clouds

Previous cloud-free images			Cloudy image
(a) 2015-06-07	2015-08-03	2015-08-10	2015-08-26
			
Cloud detection			Cloud removal
(b) Proposed mask	Fmask ⁶	Discrepancies	Estimated image
			

Solutions – open source solutions

- Google Earth Engine
- QGIS
- Python



Google Earth Engine

The screenshot displays the Google Earth Engine web interface. The browser address bar shows the URL <https://code.earthengine.google.com>. The interface is divided into several panels:

- Scripts Panel:** On the left, a list of scripts is shown under the user `users/nitamihaidaniel/default`. The script `TAT2018` is selected.
- Code Editor:** The central panel shows the script code for `TAT2018`. The code selects Landsat bands B3, B4, and B5, clips them to a 500m buffer, and adds them to a new band set `B543`. It then prints the band names and adds the layer to the map.
- Inspector Panel:** On the right, the `Inspector` tab shows the output of the `print` statement: `stacked_composite bands` with a list of bands `["B5", "B4", "B3"]`.
- Map:** The bottom panel shows a map of the region. A rectangular area is highlighted with a green and yellow color scale, representing the stacked composite of the selected bands. The map includes labels for locations like `Recea`, `Tosno`, `Remetea`, `Lăzarea`, `Gheorgheni`, and `Vargatac`, as well as `NEAMT COUNTY` and `HARGHITA COUNTY`.

The Google logo is visible in the bottom left corner of the map area.

Creating the code – New file

The screenshot shows the Google Earth Engine web interface. The browser address bar displays <https://code.earthengine.google.com>. The main header includes the Google Earth Engine logo and a search bar labeled "Search places and datasets...". Below the header, there are tabs for "Scripts", "Docs", and "Assets". The "Scripts" tab is active, showing a list of scripts under the "Owner (1)" section, specifically for "users/nitamihaidaniel/default". A "NEW" button is visible, and a dropdown menu is open, showing options: "Repository", "Folder", and "File". The "File" option is selected. In the foreground, a "Create file" dialog box is open. It contains a "Path" field with the text "users/nitamihaidaniel/default" and a text input field containing "TAT2018". Below this is a "Description" field with the placeholder text "optional commit message". At the bottom right of the dialog are "Cancel" and "OK" buttons.

Google Earth Engine

Search places and datasets...

Scripts Docs Assets

Filter scripts...

NEW

Repository

Folder

File

Owner (1)

- users/nitamihaidaniel/default
 - Landsat
 - Landsat - SCERIN
 - Landsat - SCERIN L5
 - Landsat - SCERIN classification
 - Landsat Month
 - Landsat Month (L7)
 - Landsat Simple Composite
 - S2Collection
 - Sentinel NDVI month
 - Sentinel1
 - Sentinel_Vidas

Writer

No accessible repositories.

Reader

No accessible repositories.

Examples

- Image
 - From Name
 - Normalized Difference
 - Expression

Create file

Path

Enter a name or path for the file:

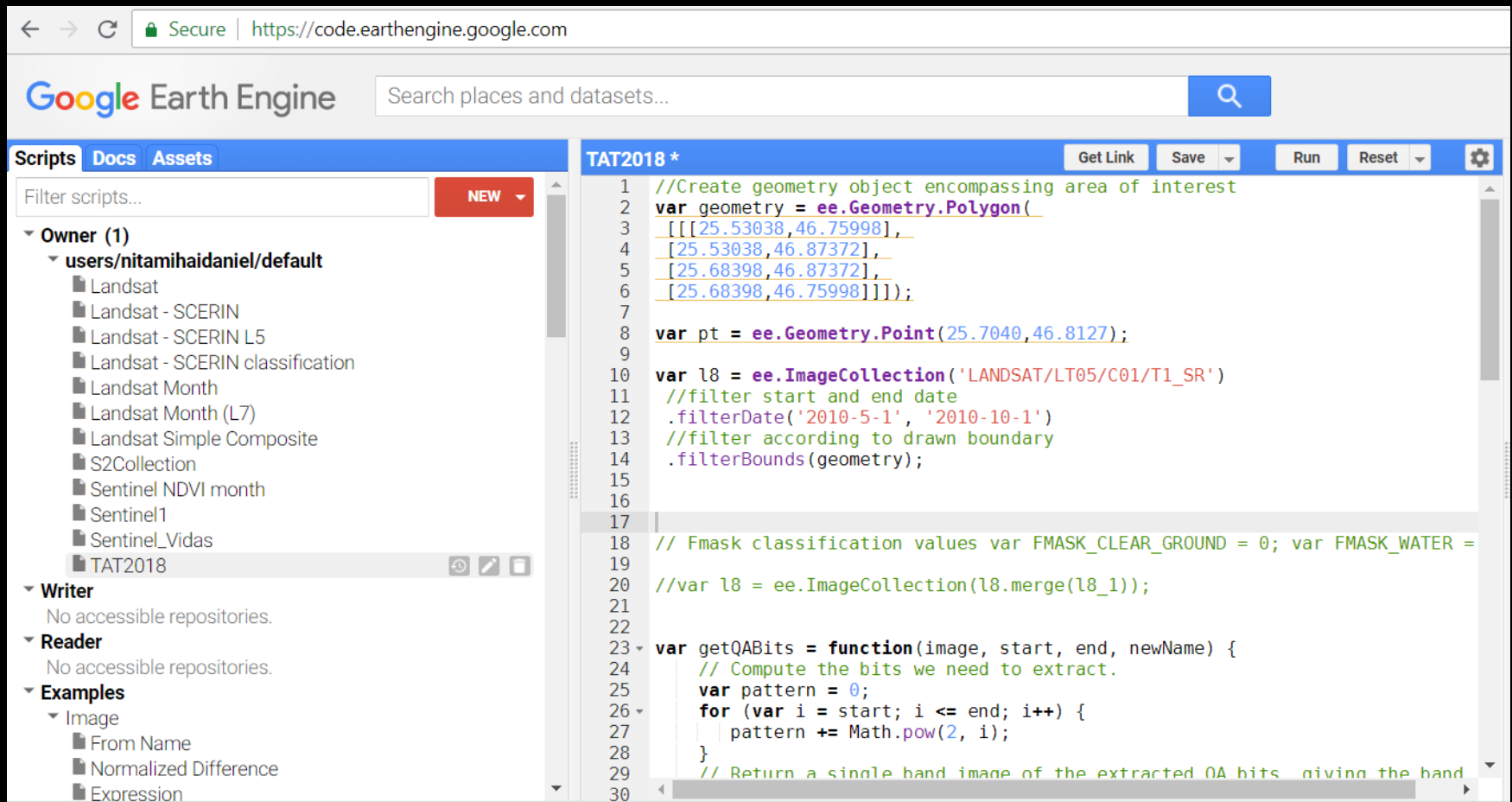
users/nitamihaidaniel/default TAT2018

Description

optional commit message

Cancel OK

Creating the code – Adding code



The screenshot shows the Google Earth Engine web interface. The browser address bar displays <https://code.earthengine.google.com>. The page header includes the Google Earth Engine logo and a search bar. The left sidebar contains a 'Scripts' tab with a search filter and a list of scripts under the owner 'users/nitamihaidaniel/default'. The script 'TAT2018' is selected. The main editor area shows the code for 'TAT2018', which includes creating a geometry, filtering an image collection, and defining a function to extract bits.

```
1 //Create geometry object encompassing area of interest
2 var geometry = ee.Geometry.Polygon(
3   [[[25.53038,46.75998],
4     [25.53038,46.87372],
5     [25.68398,46.87372],
6     [25.68398,46.75998]]]);
7
8 var pt = ee.Geometry.Point(25.7040,46.8127);
9
10 var l8 = ee.ImageCollection('LANDSAT/LT05/C01/T1_SR')
11 //filter start and end date
12 .filterDate('2010-5-1', '2010-10-1')
13 //filter according to drawn boundary
14 .filterBounds(geometry);
15
16
17
18 // Fmask classification values var FMASK_CLEAR_GROUND = 0; var FMASK_WATER =
19
20 //var l8 = ee.ImageCollection(l8.merge(l8_1));
21
22
23 var getQABits = function(image, start, end, newName) {
24   // Compute the bits we need to extract.
25   var pattern = 0;
26   for (var i = start; i <= end; i++) {
27     pattern += Math.pow(2, i);
28   }
29   // Return a single band image of the extracted QA bits giving the band
30
```

Simple copy paste the existing code from `landsat_TAT_code.txt`

Creating the code – Define variables

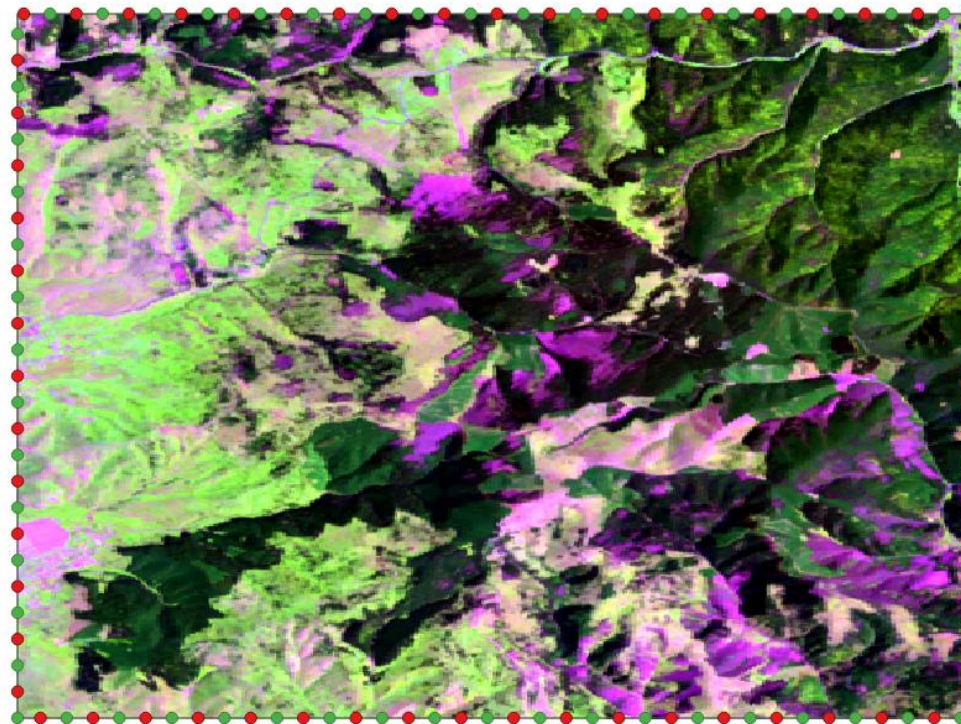
Extent

```
1 //Create geometry object encompassing area of interest
2 var geometry = ee.Geometry.Polygon(
3   [[[25.53038, 46.75998],
4     [25.53038, 46.87372],
5     [25.68398, 46.87372],
6     [25.68398, 46.75998]]]);
7
```

25.53038,46.87372
(2 - min,max)

25.68398,46.87372
(3 - max,max)

25.53038,46.75998
(1 - min,min)



25.68398,46.75998
(3 - max,min)

Creating the code – Define variables

```
var geometry = ee.Geometry.Polygon(  
  [[[25.53038,46.75998],  
    [25.53038,46.87372],  
    [25.68398,46.87372],  
    [25.68398,46.75998]]]);  
  
var pt = ee.Geometry.Point(25.7040,46.8127);  
  
var l8 = ee.ImageCollection('LANDSAT/LT05/C01/T1_SR')  
  //filter start and end date  
  .filterDate('2010-5-1', '2010-10-1')  
  //filter according to drawn boundary  
  .filterBounds(geometry);
```

Define the Landsat
image collection filtering
by date
by geometry

Custom name of
your Image
collection

The screenshot shows the Google Earth Engine web interface. On the left, the 'Scripts' tab is active, showing a list of scripts under the user 'users/nitamihaidaniel/default'. The main panel displays a search for 'landsat' in the 'PLACES' section, showing 'Landsatz, Damnatz, Germany'. Below this, the 'RASTERS' section lists various Landsat datasets. The dataset 'USGS Landsat 5 Surface Reflectance Tier 1' is selected, and an 'import >' button is visible. A yellow arrow points from the 'filterBounds(geometry)' line in the code to this dataset. A blue box in the bottom right corner displays the 'ImageCollection ID' as 'LANDSAT/LT05/C01/T1_SR' and an 'Import' button.

Creating the code – Masking clouds

```
var getQABits = function(image, start, end, newName) {  
  // Compute the bits we need to extract.  
  var pattern = 0;  
  for (var i = start; i <= end; i++) {  
    pattern += Math.pow(2, i);  
  }  
  // Return a single band image of the extracted QA bits, giving the band  
  // a new name.  
  return image.select([0], [newName])  
    .bitwiseAnd(pattern)  
    .rightShift(start);  
};
```

Getting the Fmask classification
FMASK_CLEAR_GROUND = 0;
FMASK_WATER = 2;
FMASK_CLOUD_SHADOW = 3;
FMASK_SNOW = 4;
FMASK_CLOUD = 5;

```
// A function to mask out cloudy pixels.  
var cloud_shadows = function(image) {  
  // Select the QA band.  
  var QA = image.select(['pixel_qa']);  
  // Get the internal cloud algorithm flag bit.  
  return getQABits(QA, 3, 3, 'Cloud_shadows').eq(0);  
  // Return an image masking out cloudy areas.  
};
```

Creates a single band with the cloud shadow value (3) calculated in QA band using Fmask classification

```
// A function to mask out cloudy pixels.  
var clouds = function(image) {  
  // Select the QA band.  
  var QA = image.select(['pixel_qa']);  
  // Get the internal cloud algorithm flag bit.  
  return getQABits(QA, 5, 5, 'Cloud').eq(0);  
  // Return an image masking out cloudy areas.  
};
```

Creates a single band with the clouds value (5) calculated in QA band using Fmask classification

```
var maskClouds = function(image) {  
  var cs = cloud_shadows(image);  
  var c = clouds(image);  
  image = image.updateMask(cs);  
  return image.updateMask(c);  
};
```

Constructs the function called maskClouds which applies the 2 masks defined earlier

Creating the code – Apply the mask

```
var l8 = l8.map(maskClouds);  
var B3 = l8.select(['B3']);  
var B3 = B3.median();  
var B4 = l8.select(['B4']);  
var B4 = B4.median();  
var B5 = l8.select(['B5']);  
var B5 = B5.median();
```

```
var l8 = ee.ImageCollection('LANDSAT/LT05/C01/T1_SR')  
  //filter start and end date  
  .filterDate('2009-5-1', '2009-7-1')  
  //filter according to drawn boundary  
  .filterBounds(geometry);
```

Masked band



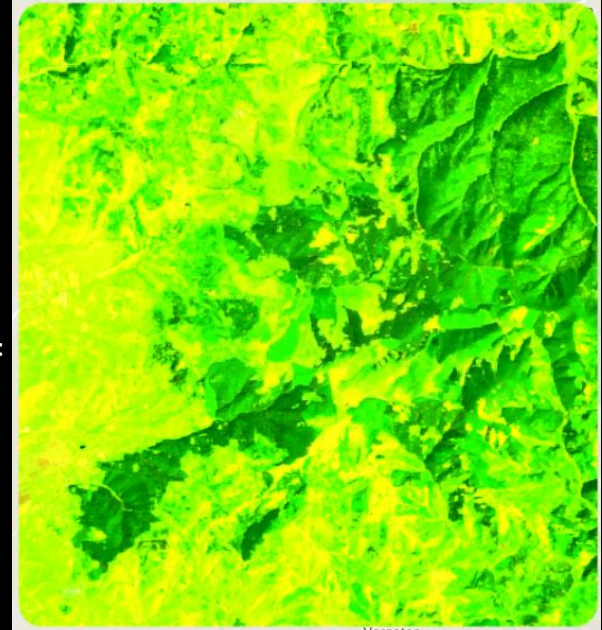
Mosaic band



+



=



Creating the code – Band stacking and clipping

```
// Clip input variable images
```

```
var B3 = B3.clip(geometry.buffer(500));  
var B4 = B4.clip(geometry.buffer(500));  
var B5 = B5.clip(geometry.buffer(500));  
  
var B54 = B5.addBands(B4);  
var B543 = B54.addBands(B3);  
print('stacked_composite bands', B543.bandNames());  
Map.centerObject(pt, 12);  
Map.addLayer(B543, {min:500, max:2500}, 'stack');
```

```
var geometry = ee.Geometry.Polygon(  
  [[[25.53038, 46.75998],  
    [25.53038, 46.87372],  
    [25.68398, 46.87372],  
    [25.68398, 46.75998]]]);
```

Inspector Console Tasks

Use print(...) to write to this console.

stacked_composite bands

▼ ["B5", "B4", "B3"]

0: B5

1: B4

2: B3

Creating the code – Exporting

The screenshot displays the 'Landsat - SCERIN L5' web application interface. On the left, a code editor shows JavaScript code for processing Landsat data. The 'Run' button in the top toolbar is highlighted with a red box. On the right, the 'Tasks' tab is active, showing a list of tasks with a 'RUN' button highlighted by a red box. A modal dialog box titled 'Task: Initiate image export' is open in the foreground, containing fields for task name, resolution, drive folder, and filename, with its 'Run' button also highlighted by a red box.

```
67 var B5 = B5.median();
68
69 // Clip input variable images
70
71 var B3 = B3.clip(geometry.buffer(500));
72 var B4 = B4.clip(geometry.buffer(500));
73 var B5 = B5.clip(geometry.buffer(500));
74
75 var B54 = B5.addBands(B4);
76 var B543 = B54.addBands(B3);
77 print('stacked_composite_bands', B543.bandNames());
78 Map.centerObject(pt, 12);
79 Map.addLayer(B543, {min:100, max:2500}, 'stack');
80
81 Export.image.toDrive({
82   image: B543,
83   description: 'B543',
84   scale: 30,
85   region: geometry,
86   maxPixels: 10E10
87 });
88 };
```

Task: Initiate image export

Task name (no spaces) *

B543

Resolution *

Scale (m/px) 30

☒ Drive ☐ Cloud Storage ☐ EE Asset

Drive folder

TAT

Filename *

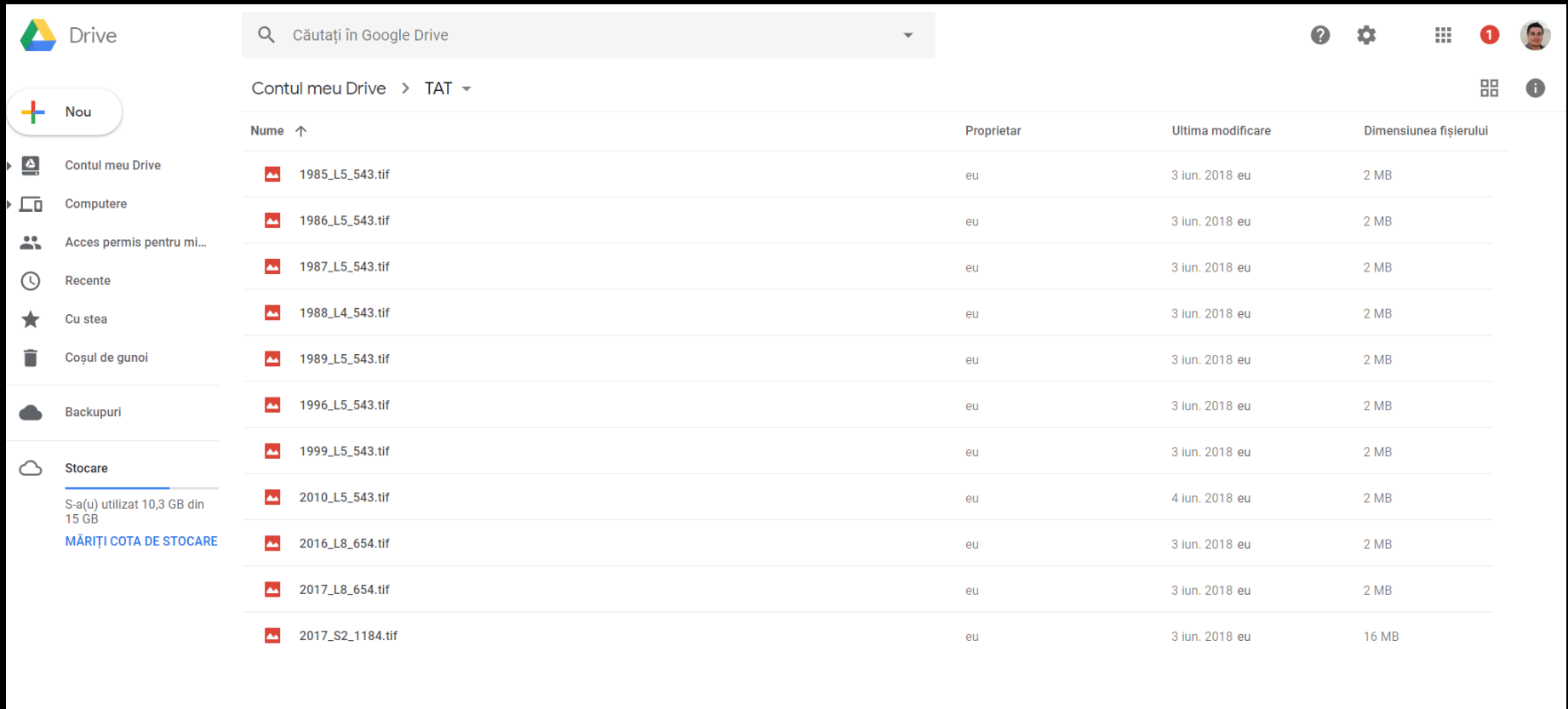
B543

Run Cancel












The 3 Runs:

- 1st Run – runs the entire procedure, adding the stack image on GE map
- 2nd Run – in tasks starts the procedure for exporting
- 3rd Run – define the scale, the filename and the path in Google Drive

Creating the code – Exporting

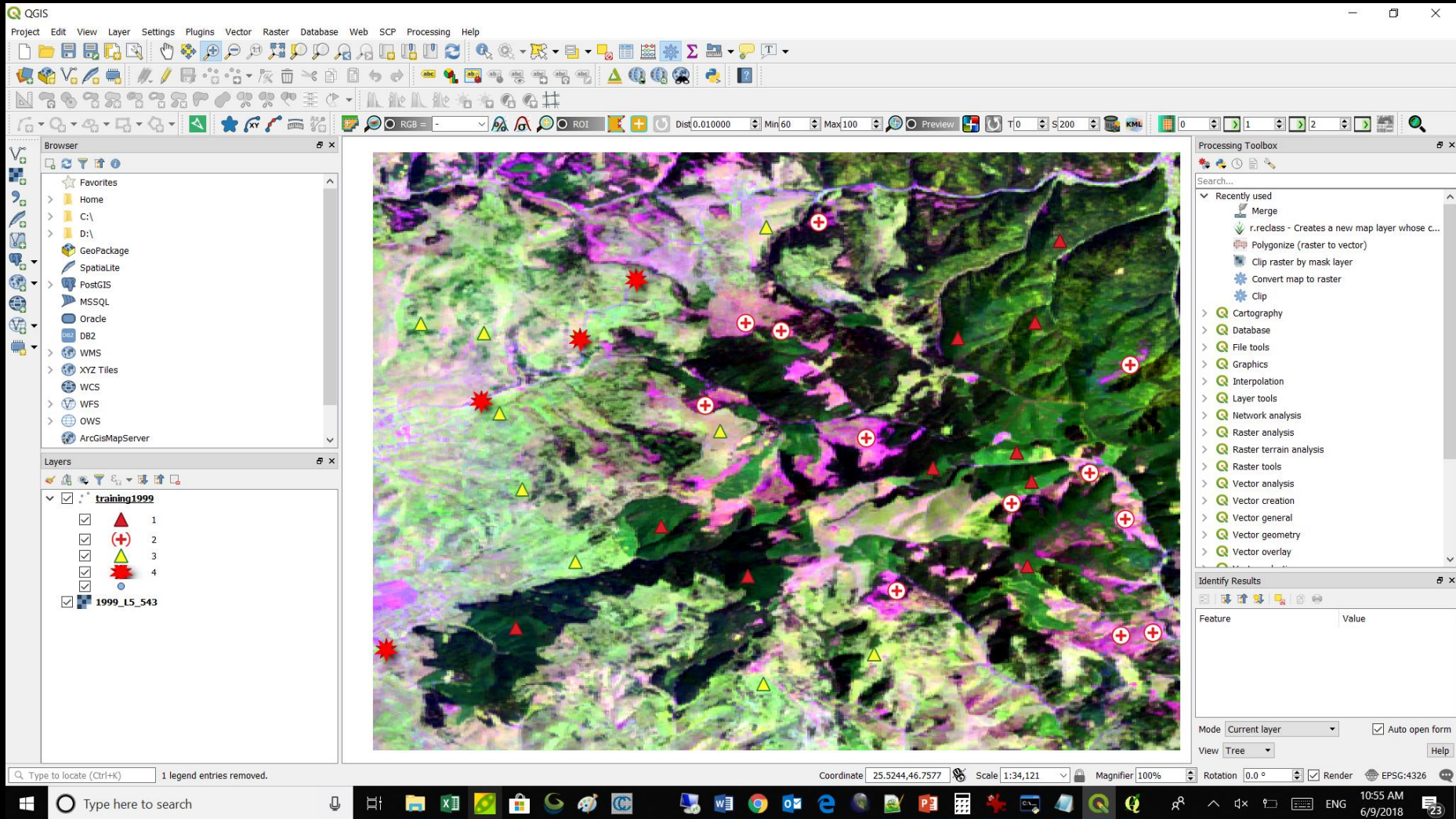


The screenshot shows the Google Drive web interface. On the left is a sidebar with navigation options: 'Nou', 'Contul meu Drive', 'Computere', 'Acces permis pentru mi...', 'Recente', 'Cu stea', 'Coșul de gunoi', 'Backupuri', and 'Stocare'. The 'Stocare' section shows a storage usage bar and a link to 'MĂRIȚI COTA DE STOCARE'. The main area has a search bar and a breadcrumb 'Contul meu Drive > TAT'. Below this is a table of files.

Nume ↑	Proprietar	Ultima modificare	Dimensiunea fișierului
 1985_L5_543.tif	eu	3 iun. 2018 eu	2 MB
 1986_L5_543.tif	eu	3 iun. 2018 eu	2 MB
 1987_L5_543.tif	eu	3 iun. 2018 eu	2 MB
 1988_L4_543.tif	eu	3 iun. 2018 eu	2 MB
 1989_L5_543.tif	eu	3 iun. 2018 eu	2 MB
 1996_L5_543.tif	eu	3 iun. 2018 eu	2 MB
 1999_L5_543.tif	eu	3 iun. 2018 eu	2 MB
 2010_L5_543.tif	eu	4 iun. 2018 eu	2 MB
 2016_L8_654.tif	eu	3 iun. 2018 eu	2 MB
 2017_L8_654.tif	eu	3 iun. 2018 eu	2 MB
 2017_S2_1184.tif	eu	3 iun. 2018 eu	16 MB

Results are geotiff, with all the stacked bands
Projection is EPSG:4326 – Geographic WGS84

QGIS



Classification tool

The screenshot shows the QGIS Plugins window with the search bar set to 'dz'. The 'dzetsaka : Classification tool' plugin is selected and its details are displayed on the right. The left sidebar shows the 'All' tab and a list of filter buttons: 'Installed', 'Not installed', 'Upgradeable', 'Install from ZIP', and 'Settings'.

dzetsaka : Classification tool
Fast and Easy Classification plugin for Qgis

Plugin for semi-automatic classification with Gaussian Mixture Model, Random Forest*, and SVM* classifiers.
Very easy and fast to use.
*You need to install scikit-learn library to use these algorithms.
For more information on this tool check our github :
<https://github.com/lennepkade/dzetsaka/>

★★★★★ 21 rating vote(s), 22230 downloads

Category Raster
Tags classification,semi-automatic,gaussian,mixture,model,random forest,svm,knn,forest,processing
More info [homepage](#) [bug tracker](#) [code repository](#)
Author [Nicolas Karasiak](#)

Installed version 3.1
Available version 3.1


Changelog


- 3.1
 - * Replace scipy with numpy when possible
 - * Correct bug in predicting an already trained model
 - * Specify a way on Windows to install scikit-learn and use SVM/RF/KNN
- 3.0.3
 - * Add confirmation box if two different projections
 - * Correct bug when loading model
- 3.0.2
 - * Add progress bar for GUI
- 3.0.1
 - * Minor fixes (with icons)
- 3.0.0
 - * First version of dzetsaka for Qgis 3.
 - * TODO : Progress bar when using UI.
 - * TODO : Historical Map algorithms


[Upgrade all](#) [Uninstall plugin](#) [Reinstall plugin](#) [Close](#) [Help](#)

Classification tool


dzetsaka : classification tool



 1999_L5_543 [EPSG:4326]

 training1999 [EPSG:4326]

or ☐ Load model


 Name

Model


Classification. Leave empty for temporary file


...


Perform the classification




▼ Optional

 ☐ Mask

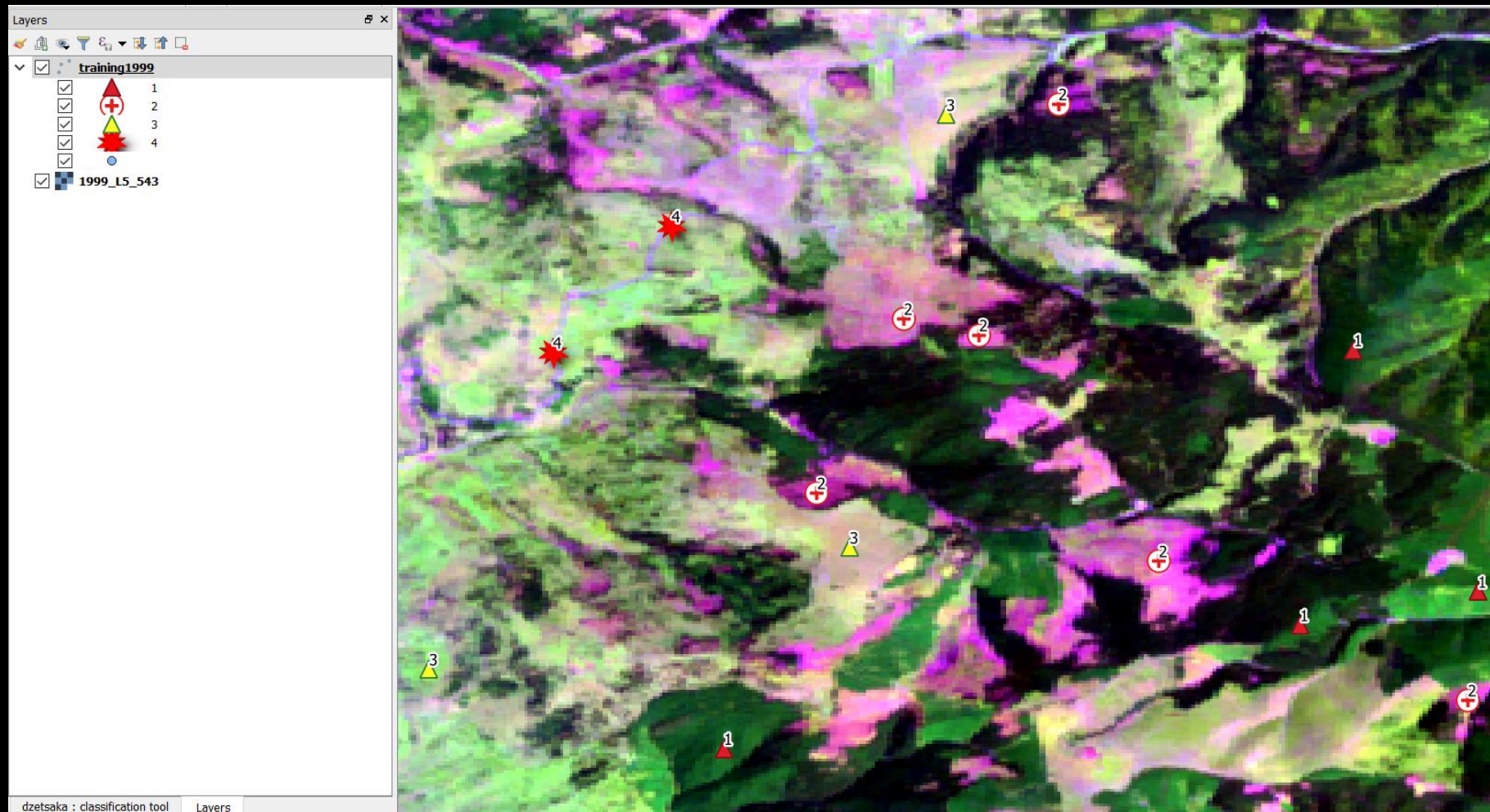
 ☐ Confidence map

 ☐ Save model

 ☐ Save matrix


Split (?)

Classification tool



Classification tool

dzetsaka : classification tool



1999_L5_543 [EPSG:4326]

training1999 [EPSG:4326] or ☐ Load model

Name Model

Classification. Leave empty for temporary file

Perform the classification

▼ Optional

☐ Mask Automatic find filena...

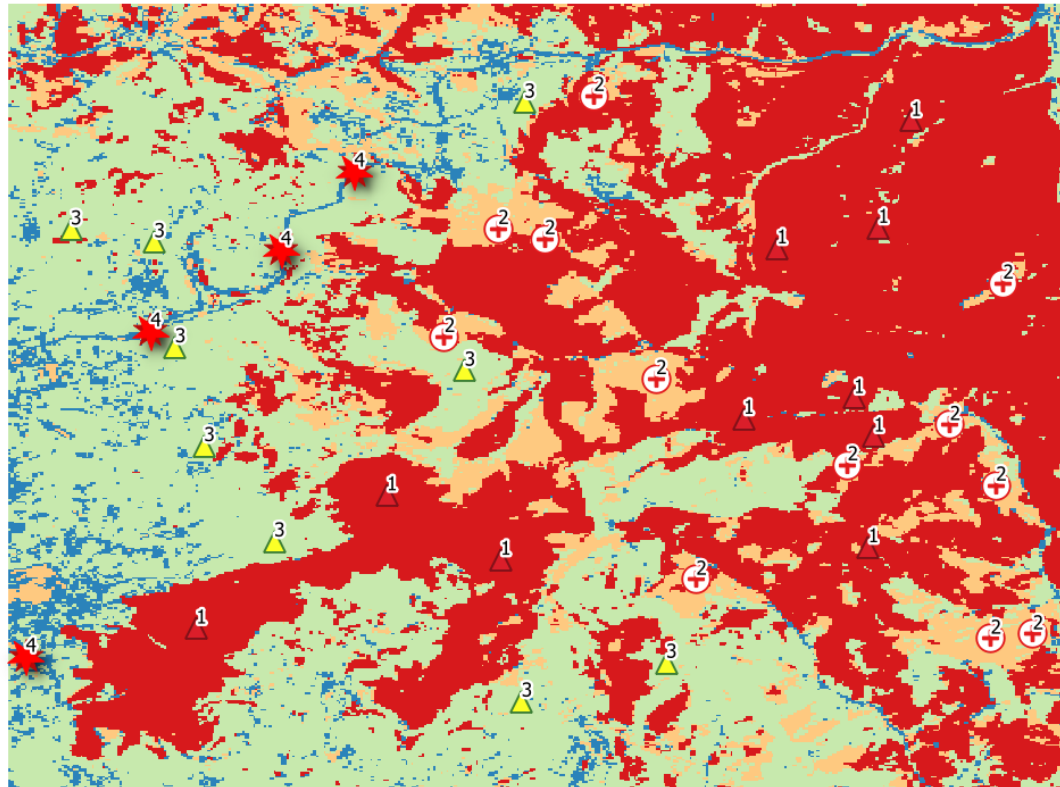
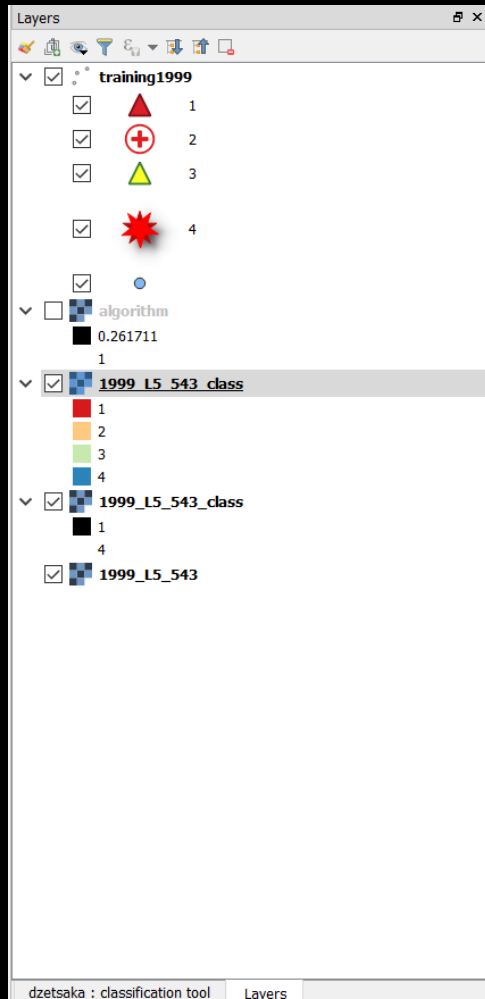
☒ Confidence map TAT/temp/algorithm.tif

☐ Save model To use with another i...

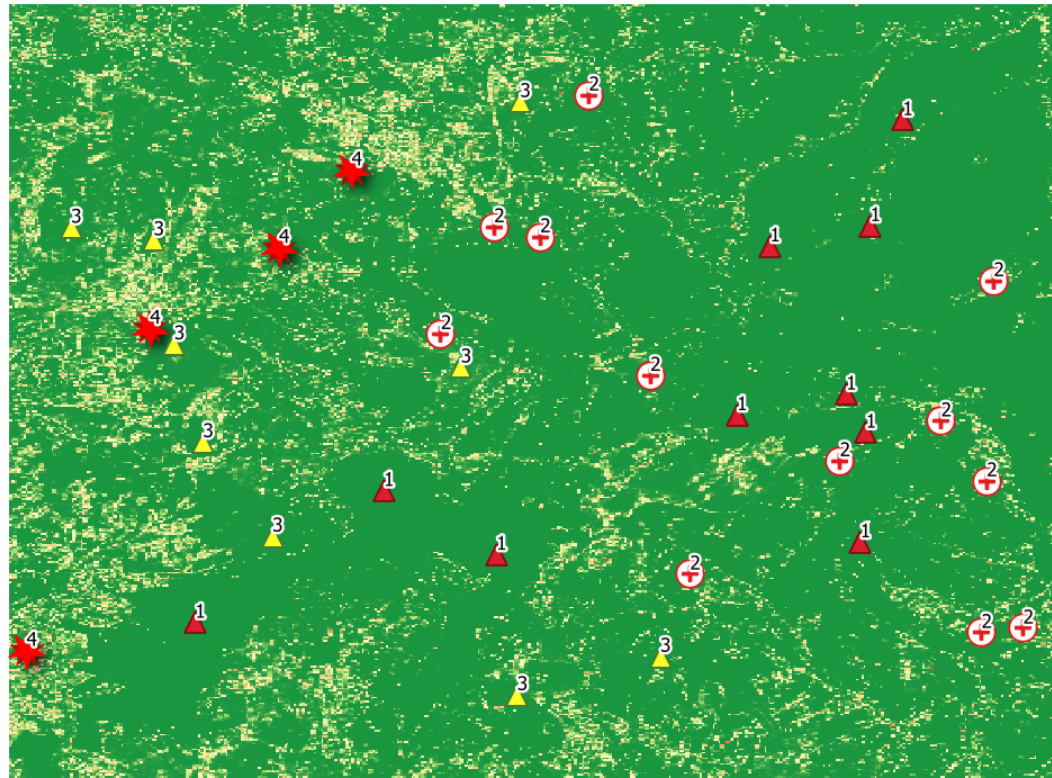
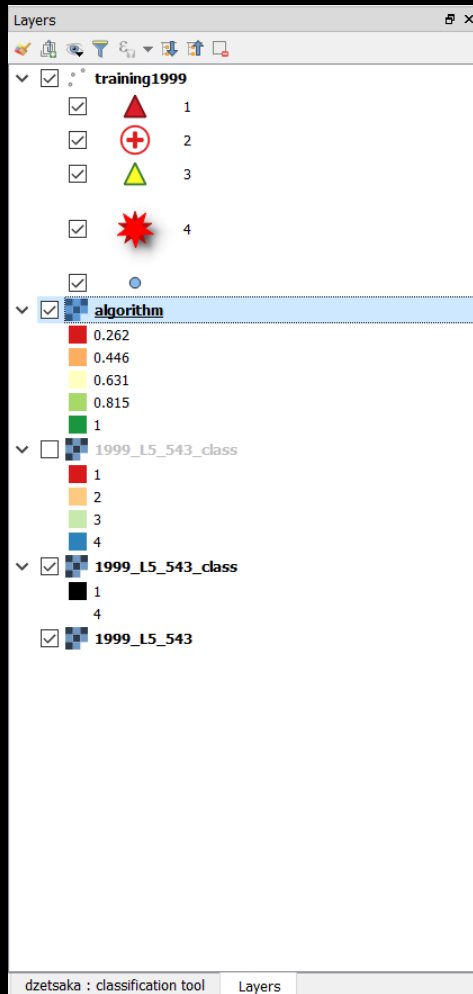
☐ Save matrix Save confusion matrix

Split (?) 100%

Results



Results



Python - geextract

← → ↻

www.loicdutrieux.net/landsat-extract-gee/install.html

🏠 geextract

0.3.2

Search docs

Introduction

USER GUIDE

☰ Installation

Quick install

☰ Install from scratch

Processing

geextract API

Command line interface

EXAMPLES

Examples

Docs » Installation

[View page source](#)

Installation

You must have a Google Earth Engine account. If you don't yet have an account, you can request for it [here](#)

Once you have an account, the package can be installed using `pip`, preferably within a virtual environment. If you're new to python and/or virtual environments, read the [Install from scratch](#) section.

Quick install

```
pip install geextract
```

If you're using the gee API for the first time on your machine, you'll have to run:

```
earthengine authenticate
```

which will open a google authentication page in your browser, and will give you an authentication token to paste back in the terminal.

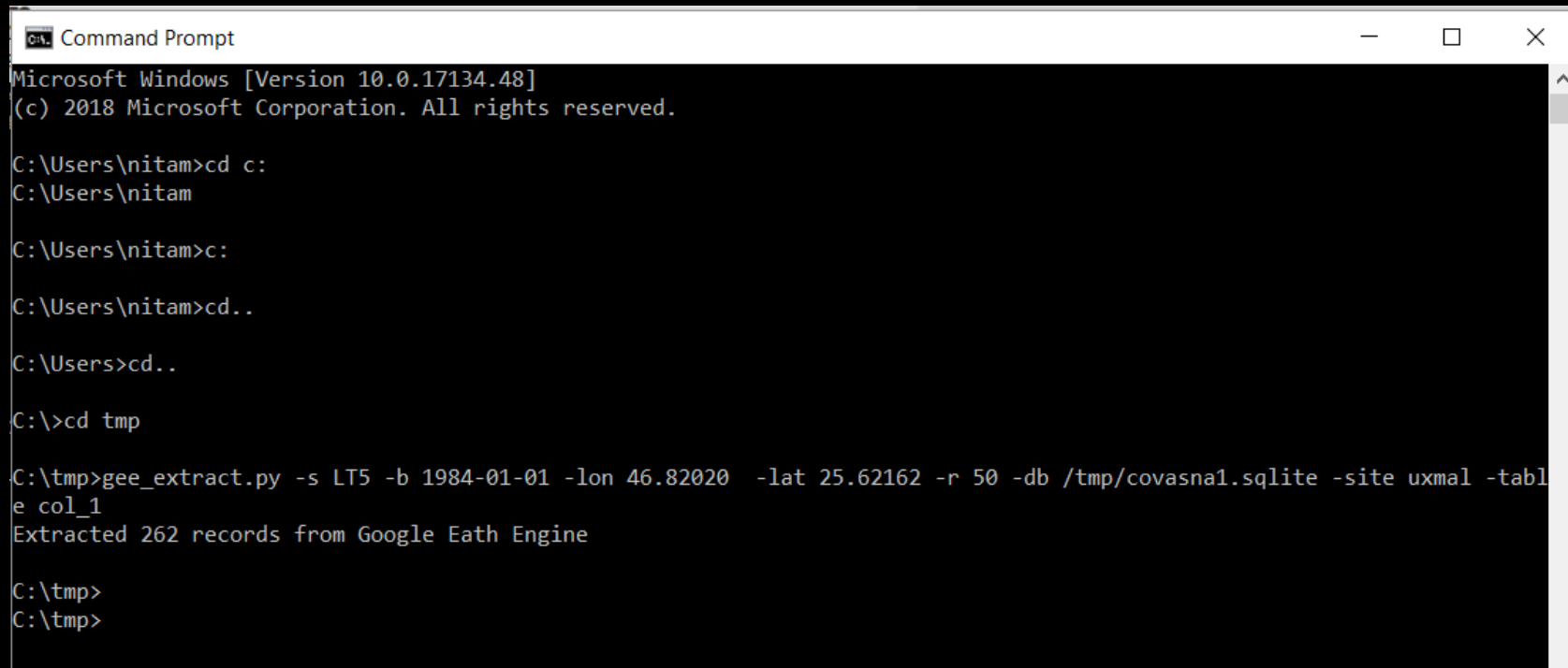
You can check that the authentication process was successful by running.

```
python -c "import ee; ee.Initialize()"
```

If nothing happens, it means that things are working... You can go ahead and use the `geextract` API and command line.

Pixel based Analysis

```
gee_extract.py -s LT5 -b 1984-01-01 -lon 25.62162 -  
lat 46.82020 -r 50 -db /tmp/cv_tst.sqlite -site uxmal -  
table col_1
```



```
Command Prompt
Microsoft Windows [Version 10.0.17134.48]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\nitam>cd c:
C:\Users\nitam

C:\Users\nitam>c:

C:\Users\nitam>cd..

C:\Users>cd..

C:\>cd tmp

C:\tmp>gee_extract.py -s LT5 -b 1984-01-01 -lon 46.82020 -lat 25.62162 -r 50 -db /tmp/covasna1.sqlite -site uxmal -table col_1
Extracted 262 records from Google Eath Engine

C:\tmp>
C:\tmp>
```

Pixel based Analysis

Layers

covasna1 col 1

Processing Toolbox

Search...

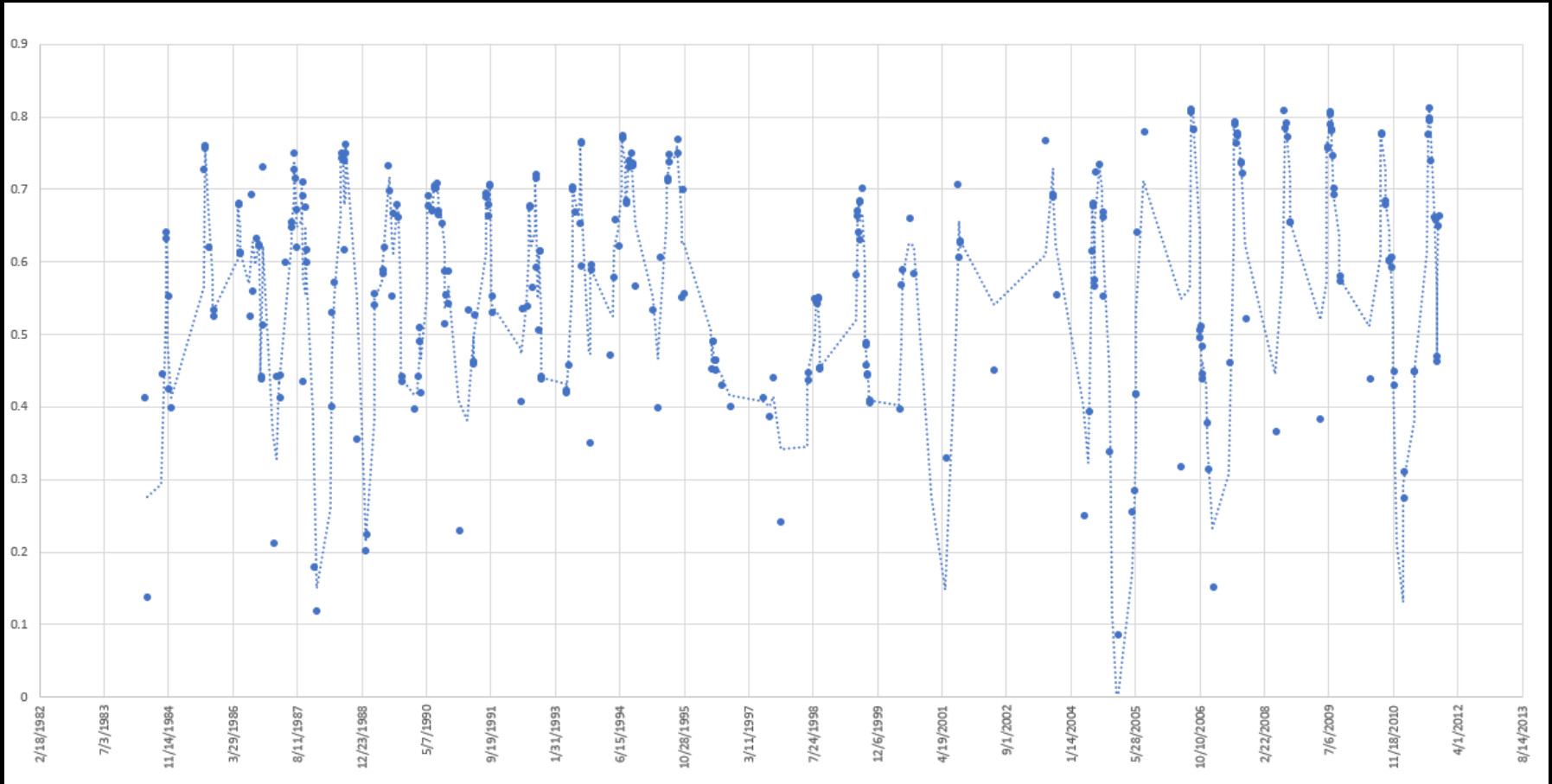
Recently used

covasna1 col_1 :: Features Total: 188, Filtered: 188, Selected: 0

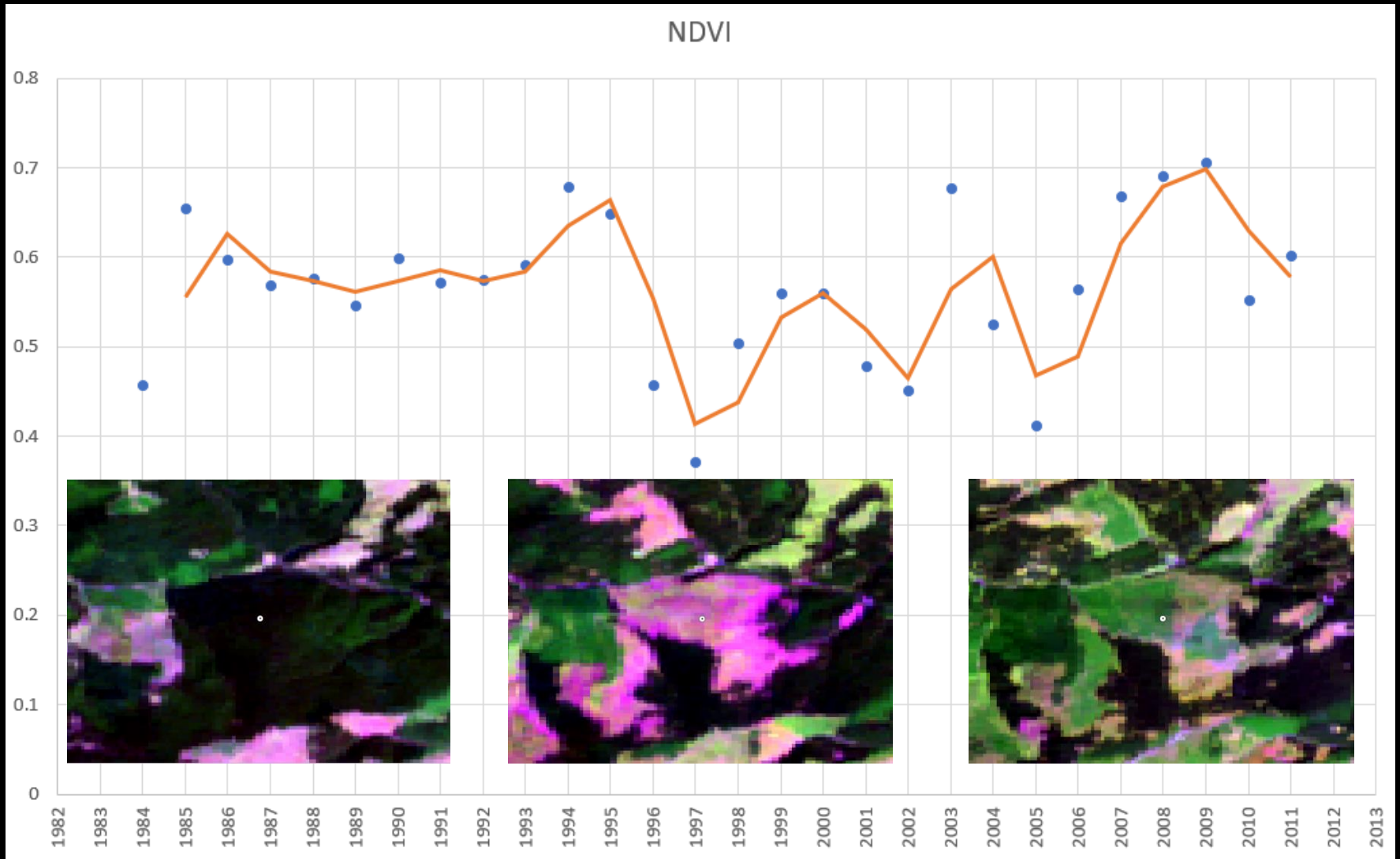
	rowid	index	blue	green	id	nir	red	swir1	swir2	time	sensor	site
1	1	0	1467.6715328467...	2494.175638686131	LT05_165042_198...	4725.334854014598	3801.594434306569	6031.58394160584	5404.284215328467	1984-09-23	LT5	uxmal
2	93	137	1268.1765510948...	2318.405565693431	LT05_166042_198...	4330.140054744525	3539.50045620438	5689.849452554744	5036.958485401459	1985-01-04	LT5	uxmal
3	2	1	1241.7302631578...	2195.2236842105...	LT05_165042_198...	4173.776315789473	3357.8526315789...	5529.7171052631...	4991.993421052632	1985-01-13	LT5	uxmal
4	94	138	1553.7093978102...	2558.790145985401	LT05_166042_198...	4415.804744525547	3656.8759124087...	5694.245894160585	5021.738594890511	1985-01-20	LT5	uxmal
5	3	2	1118.7390510948...	1713.9206204379...	LT05_165042_198...	3079.1610401459...	2540.768248175183	3763.0209854014...	3151.1382299270...	1985-01-29	LT5	uxmal
6	95	139	1400.3302919708...	2427.9069343065...	LT05_166042_198...	4261.923357664234	3562.757755474453	5585.661040145986	4986.251368613139	1985-02-05	LT5	uxmal
7	96	140	1311.8202554744...	2414.009580291971	LT05_166042_198...	4571.097171532847	3723.3385036496...	5961.852645985402	5299.275547445255	1985-03-09	LT5	uxmal
8	97	141	1526.7244525547...	2557.501824817519	LT05_166042_198...	4510.155565693431	3715.200729927008	5816.835766423358	5250.703010948905	1985-03-25	LT5	uxmal
9	98	143	1370.5009124087...	2368.6783759124...	LT05_166042_198...	4266.3722627737...	3575.647810218978	5682.212591240876	4990.953923357663	1986-02-08	LT5	uxmal
10	4	4	1348.6145072992...	2166.568886861314	LT05_165042_198...	3696.751824817519	3143.929744525547	4487.349908759124	3983.22947080292	1986-02-17	LT5	uxmal
11	99	144	1339.1386861313...	2464.445711678832	LT05_166042_198...	4679.2002737226...	3781.089872262774	6059.001368613138	5530.923813868613	1986-02-24	LT5	uxmal
12	100	145	1292.8243613138...	2380.9393248175...	LT05_166042_198...	4504.930656934306	3712.86496350365	5951.085766423359	5323.421532846716	1986-02-24	LT5	uxmal
13	101	146	1350.3736313868...	2386.2262773722...	LT05_166042_198...	4546.49908759124	3705.4835766423...	5836	5414.451186131388	1986-07-02	LT5	uxmal

Show All Features

Pixel based Analysis



Pixel based Analysis



Practice

