EAMENA Machine Learning Automated Change Detection (MLACD)

Training Documentation

Developed and written by Dr. Ahmed Mahmoud

Revised by Dr. Nichole Sheldrick

Translated into Arabic by Dr. Ahmed Buzaian

October 2023

Glossary

EAMENA: Endangered Archaeology in the Middle East and North Africa

GEE: Google Earth Engine

ML: Machine Learning

ACD: Automated Change Detection

JV: JavaScript

TS: Training Sample

Script: A program code with a sequence of processes and functionalities

Automated Change Detection (ACD)

1 Introduction

The EAMENA machine learning automated change detection (EAMENA MLACD) is a tool developed by EAMENA researchers to rapidly monitor the changes and threats at and around archaeological sites using satellite images. For a comprehensive understanding of the tool, see the full research paper here: https://doi.org/10.1016/j.rsase.2024.101396.

The tool uses the cloud computing service Google Earth Engine (https://earthengine.google.com/). It was developed using JavaScript and machine learning algorithm (Random Forest) to process a time series of satellite images and create land classifications maps to detect the changes and threats in archaeological sites over time (Figure *1*).

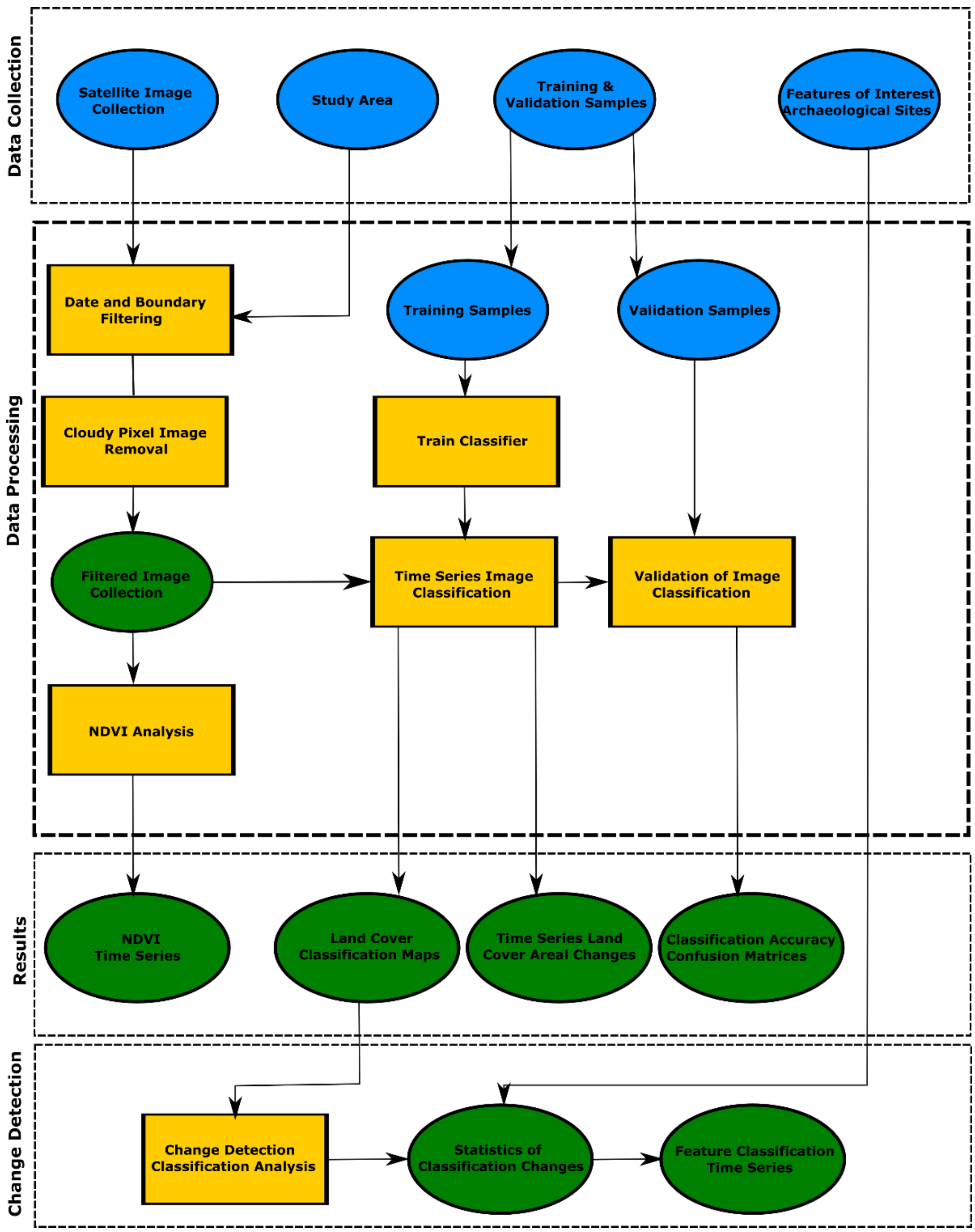


Figure 1. EAMENA MLACD Framework.

2 Pre-requisites

To use the EAMENA MLACD you will need a Google Earth Engine account.

- To sign up for Google Earth Engine, follow the instructions here: https://signup.earthengine.google.com/#!/
- It sometimes takes several days for applications to be approved.

3 Getting Started

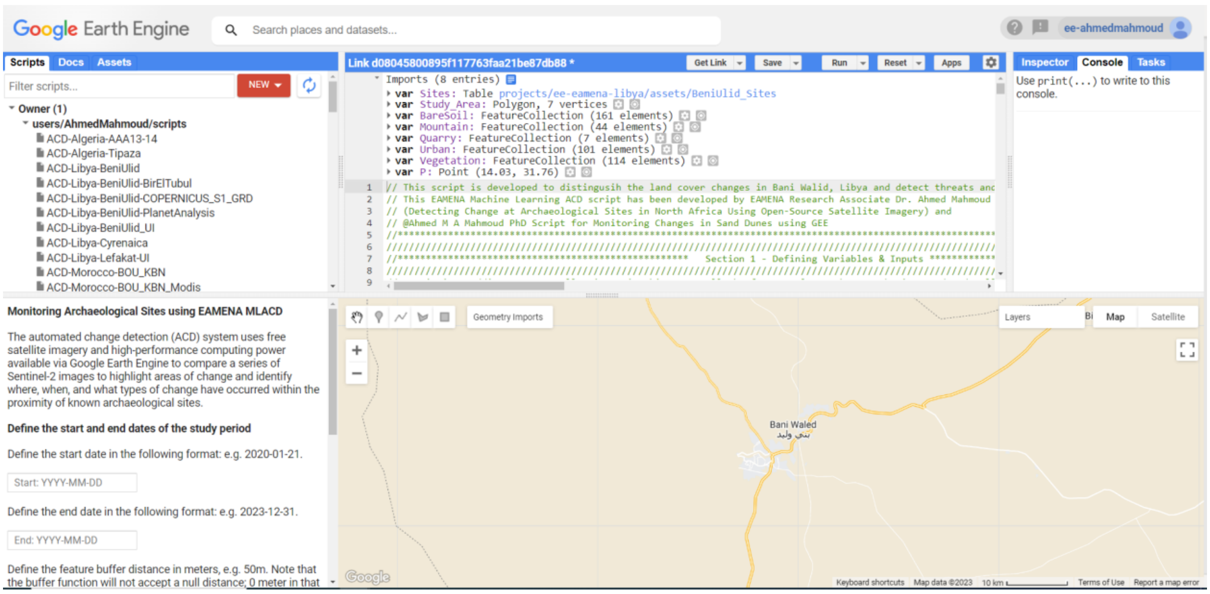
Once your application has been approved, go to https://code.earthengine.google.com/ to access the Code Editor where we will run the MLACD.

The EAMENA team has prepared pre-set MLACD case studies which you can access at the link below:

- Bani Walid: https://code.earthengine.google.com/26b14ceb8b13847515512d6a99909a98.

This documentation will use the Bani Walid case study to outline the steps of the tool.

- Click on the Bani Walid link to open the script.



- Click on 'Save' to save the script to your GEE repository.
- Give it a name like 'EAMENA_MLACD' so that you can return to it
  - You can always click on the link above again to recover the original version
  - You can also always find the most up to date version of the MLACD script on the EAMENA GitHub page: https://github.com/eamena-project , https://github.com/eamena-project/EAMENA-MachineLearning-ACD

To get started with using the tool there are several elements and parameters that have to be defined and edited in the EAMENA MLACD JavaScript code. The workflow is divided into three main stages:

1. Defining variables and inputs.
2. Image classification and analysis.
3. Identification of threats on archaeological sites.

4 Stage 1: Defining variables and inputs

In the first stage, you will define several things:

1. the study area.
2. the archaeological sites under investigation.
3. the training samples.
4. A location of interest.
5. the visualisation parameters.
6. the values and names for the charts and other outputs.

All these defined inputs and variables will facilitate the automation and generation of results in the next stages of the script.

4.1 Defining the Study Area

The pre-set scripts will come with the Study Area already defined.

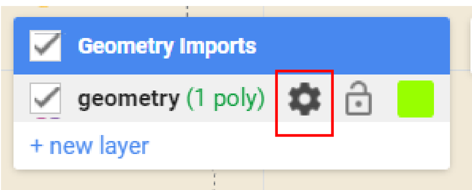
If you want to investigate a new area you will need to change this by deleting the existing Study_Area and entering your own study area using one of the two following methods.

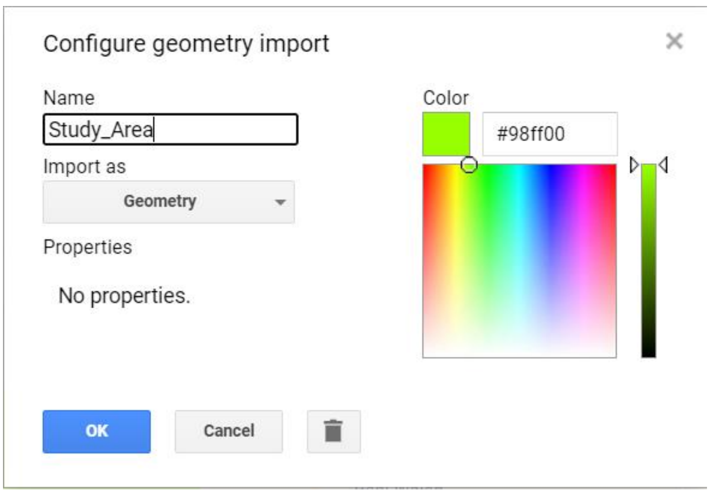
4.1.1 Method 1: Draw with the Geometry Tool

The Study Area can be drawn manually using the **Geometry tool** in the upper left corner of the GEE map panel.

- Create a new geometry layer by drawing a polygon around the study area using the rectangle or polygon tool

- Hover over the 'Geometry Imports' box and open the configure tool by clicking on the gear icon to change the properties of the new layer
- Change the Name of the geometry to '**Study_Area**'
- Keep the feature type for the study area under 'Import as' as 'Geometry'
- It will appear in your Imports at the top of the Script





4.1.2 Method 2: Shapefile Upload

If you have already defined your study area in a GIS software and you have it as a shapefile, you can also upload it directly to GEE.

- From the main GEE code editor panel, go to the Assets tab and select **NEW**
- Under **Table Upload** click on **Shapefiles (.shp, .shx, .dbf, .prj, or .zip)**.
- Follow the instructions to upload your shapefile and give it a name that you will remember
    - Make sure you select all the shapefile extensions or select a zip folder that contains all the supporting file extensions for the shapefile.

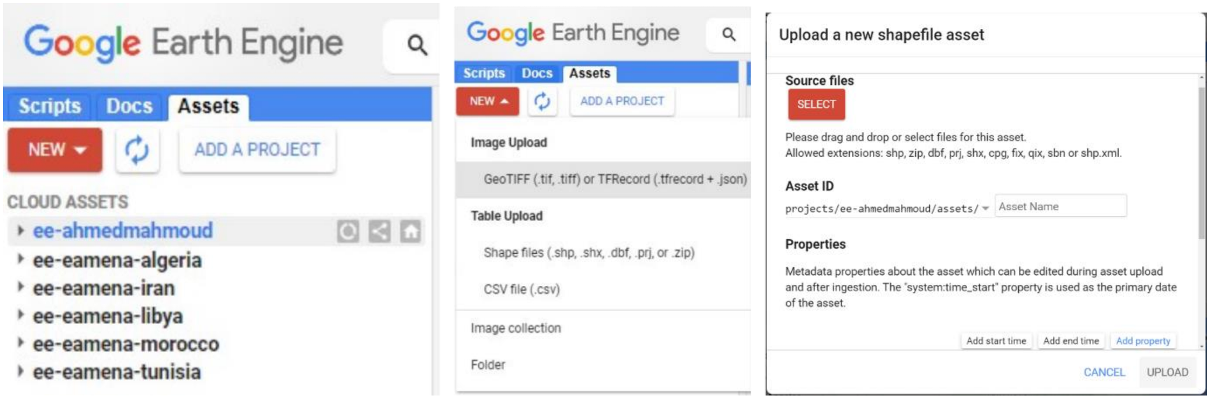
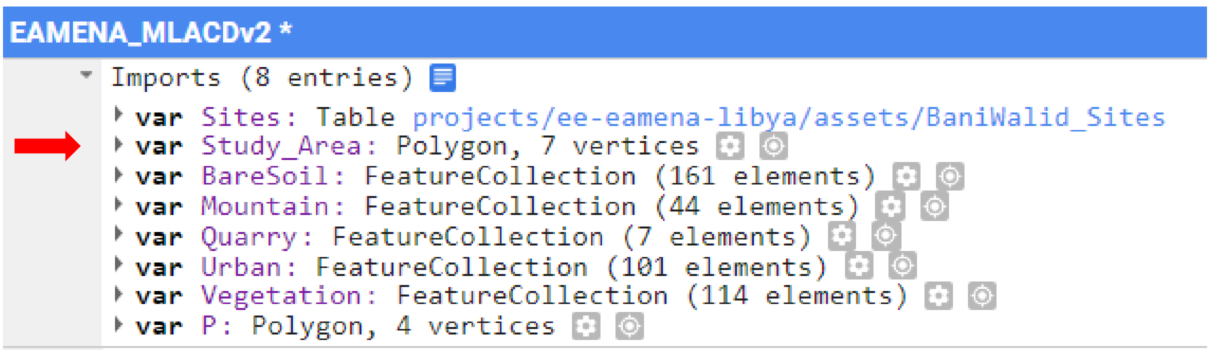


Figure 2. Adding assets in GEE projects

- Once it has uploaded to your Assets list, click on it and click 'Import' to import it into your script
- When you have imported it, you will see it appear at the top of the script named 'table'.
- You **MUST** change the name to '**Study_Area**'
- Then you must go to Section 1 of the Script, under 'Define the Study Area' and find the line in green which says Study_Area and delete the // at the beginning of the line so that it turns purple



Important Notes for both methods:

- You must label your feature for the area of interest as "**Study_Area**" as this label is used in EAMENA MLACD JavaScript code to execute other tasks.
- In order for the MLACD to work, the Study Area you define must cover ALL of the archaeological sites that you wish to analyse, which you will define in the next section.

4.2 Defining the Archaeological Sites under Investigation

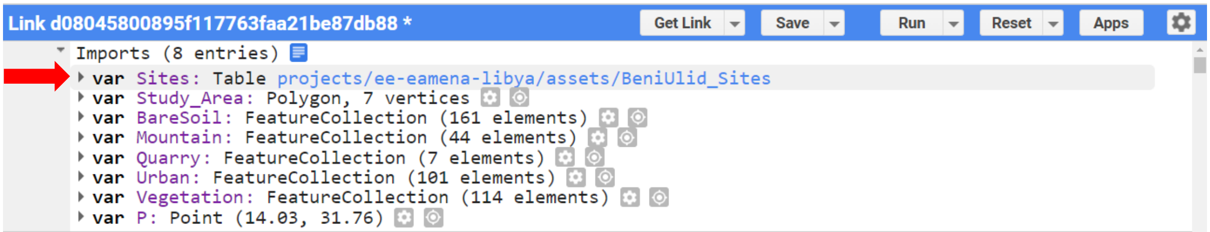
The pre-set scripts come loaded with a set of archaeological sites within the pre-defined study areas.

If you want to investigate a new dataset, you will need to change this by deleting the Import called 'Sites' at the top of the script and adding your own, using one of the two following methods.

4.2.1 Method 1: Shapefile Upload

Follow the exact same method you used to upload a Study Area shapefile in the last section

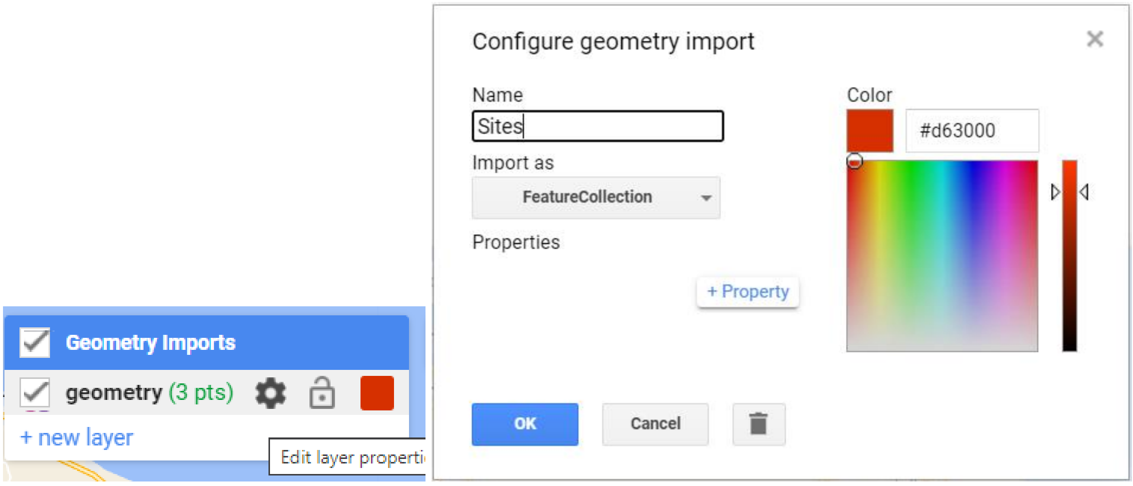
- Upload the shapefile as an asset and import it into your script using the same instructions you learned to upload the Study Area in Section 4.1.1
- Once you have imported the table to your script, you MUST change the name to '**Sites**'



4.2.2 Method 2: Draw with the Geometry Tool

Alternatively, the Site Locations can be drawn manually using the **Geometry tool** in the upper left corner of the GEE map panel.

- Use the Geometry tool to place a Point at each archaeological site that you wish to analyse
- When you have placed a point on every site, hover over the 'Geometry Imports' box and open the configure tool by clicking on the gear icon
- Change the Name to '**Sites**' and change the type under Import as to '**Feature Collection**'
- Click OK, and your Sites will appear under your Imports at the top of the script



Important Notes for both methods:

- You must label sites as "**Sites**" as this label is used in EAMENA MLACD JavaScript code to execute other tasks.

4.3 Defining the training samples

The MLACD script works by creating a series of land classifications maps for the area of interest and time period defined by the user.

- In order to create the land classification maps, we must first 'train' the machine learning model to recognise the different classes that we want to identify, for example, vegetation, bare soil, urban areas, etc. by creating Training Samples.
- Each Training Sample (TS) will be a representative location of the land cover class features that can be identified from a satellite image (e.g. urban, vegetation, etc).

To create the Training Samples, in the Code Editor script go to **Section 1: Defining Variables and Inputs**, and scroll down to the section called **Define Training Samples and Classification Variables**

- Training Samples can be created as polygons or points
    - **Polygons** are recommended here as it is a faster way to collect a large number of samples for each class.
    - We use the stratified random sampling approach to generate sample points from the collection of Training Sample polygons datasets provided by the user.

4.3.1 Prepare the variables in the script

The first step, is to decide, for your area, how many different classes of landcover are found there and which ones you want to define.

- You will see in the Bani Walid example, that we have defined 5 classes
    - The training samples has been given a generic label (**TS**) for instance the first training sample collected data for class one is TS1 = BareSoil
    - Each class is defined as a variable, by writing "**var**" in front of it so that no editing will be required after you execute the first stage of the processing.
    - You can delete a variable to make fewer classes, or add additional ones to add more classes.
- Each class must then also be given a label that you want to appear in the outputs that the script will generate later
    - Each label is also defined as a variable, with the general label (**C**), so that the label for Training Sample 1, is defined as C1 = bareSoil

```
// Define the training samples dataset of your case study
var TS1 = BareSoil; var TS2 = Mountain; var TS3 = Quarry; var TS4 = Urban; var TS5 = Vegetation; // Add
// Define variables for each land cover class to contain the names of the training datasets
var C1 = 'BareSoil'; var C2 = 'Mountain'; var C3 = 'Quarry'; var C4 = 'Urban'; var C5 = 'Vegetation';
```

Important Note: Make sure you have the same number of **TS** and **C** variables and that they are in the same order.

- Define all training datasets within a single array. This step is essential for automating the script. You can easily add or remove training sample variables based on your case study; for example, remove TS5 if you only have four training datasets, or add TS6 if you have six. The training sample variables should reflect the specific classes and characteristics relevant to your study area.

```
// Define all training sets in an array
var trainingSets = [TS1, TS2, TS3, TS4, TS5];
```

4.3.2 Define the Class Array and Class Values

In order for different functions later in the script to work properly, we must also define what are called the Class Array and the Class Values, which are found in the new few lines of the script

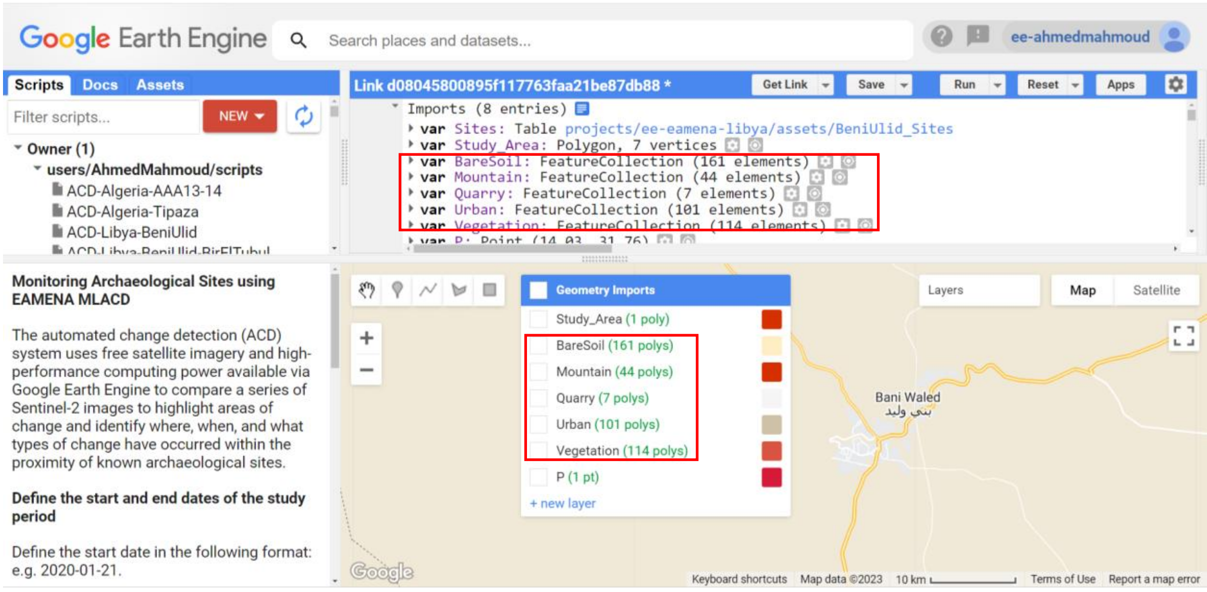
- Define the Class Array which contain the training samples variables names by adding each of the Class (C) variables you defined in the previous section to the list inside the square brackets where it says **var classArray** as in the example below.

```
// Define a Class Array which contain the training dataset names
var classArray = [C1,C2,C3,C4,C5];
```

4.3.3 Collecting Training samples

In this section we will learn how to collect the Training Samples by drawing polygons using the GEE Geometry Tool.

- For the script to work, there needs to be one layer for each landcover class that you wish to identify, each of which will have multiple polygons.
- In the Bani Walid example, as discussed in the previous section, we defined 5 classes, therefore, you can see that there are five training sample layers, that have the same names as defined above, e.g. TS1 = BareSoil



If you turn on each of the Training Sample layers one by one, you will see that each layer consists of multiple polygons, distributed across the study area.

- Each polygon encircles a representative area of the class that it is defining
- So for example, in the Vegetation layer, there are 114 polygons, each one encircles an area of vegetation, to 'teach' the machine learning model in the script what vegetation looks like.

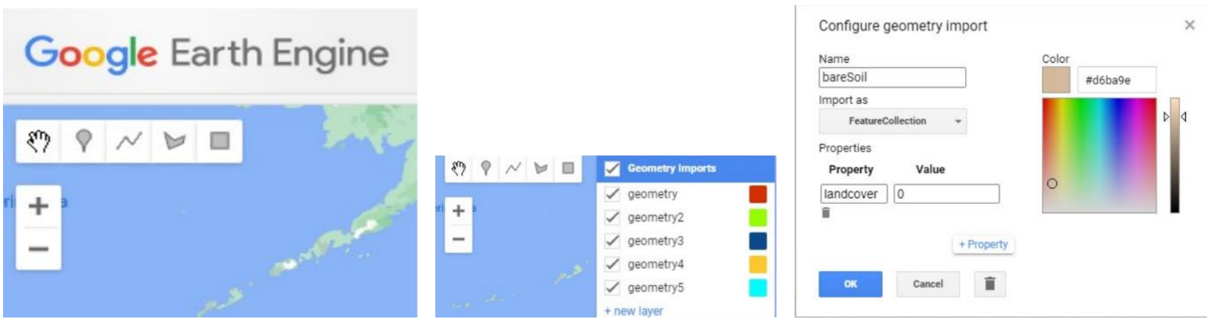
If you want to investigate a different study area, you will need to delete the existing Training Sample layers and create new ones in your new Study Area to train the script for your area of interest. You can do this by different methods described below.

4.3.3.1 Collecting Training samples using the Geometry Tool in GEE

This is the method that was used to create the Training Samples for the Bani Walid example.

To create a new Training Sample layer:

- Hover over the geometry box and click '+ new layer'
- Hover over your new geometry layer and click the gear icon to 'Edit layer properties'
- Rename the Name to the Class that you want to define
    - This must be **identical** to the way that you have identified the TS variable in the previous section, e.g. if **var** TS1 = BareSoil, your layer must also be named 'BareSoil'
    - You will need to make a separate layer for each Class
- Under Import as; select 'FeatureCollection', because your layer will contain multiple features
- Click the +Property button
    - Set the Property to '**landcover**' in small letters as this text name will be used later on the script to execute some other processes.
    - Set the Class Value to match those defined in the previous section. This should be the **TS number minus 1**.
        - So for TS1 Value = 0; for TS2 Value = 1; for TS3 Value = 2, etc.
- Choose a different colour for each class
- Click OK



Once you have created your layer, you can start drawing polygons

- There is not set rule for how many polygons you need – it is dependent on the size of your study area, and how much of the study area you think might be classified as this class
    - For example, you can see in the Bani Walid example that Class 3 Quarry only has 7 polygons, while Class 1 BareSoil has 161

Here are some tips to drawing your polygons:

- The goal is to find and create polygons around several representative examples of the class you are defining.
- Polygons can be various sizes. They should not be too big, because you want to ensure that every single pixel within the polygon you are drawing belongs to the class you are trying to identify.
- Spread out your polygons across your study area to ensure well representation of the training sample.
- To ensure accuracy, collect training samples from high-resolution imagery in GEE that aligns with your study period and covers all target classes.
- High-quality training samples directly influence classification accuracy. Invest time in careful collection and validation to ensure reliable results.

Example of collected training samples for vegetation areas representing the vegetation class in Tipaza region in Algeria.

4.3.3.2 Collecting Training samples using Google Earth Pro

A different way to collect your training samples is to draw your polygons in Google Earth Pro using high-resolution archived imagery which must improve your classification results accuracy.

- Save the polygons for a single class in a folder and export the folder as a .kml
- Import the .kml to QGIS and then re-export it as a shapefile.
- If you collect the training samples using the Google Earth Pro you will have to import each class dataset into QGIS first and create a new field containing the class land cover type label, which should be named '**landcover**' and you have to add the landcover type label (BareSoil, Vegetation, Urban, etc) to each feature. Also, you should create another field and name it '**Value**' and define the class value for each feature (i.e., 0, 1, 2, 3, 4…) for example the value for class BareSoil is "0".
- Upload the shapefile of your Training Samples to your GEE Assets using the same method described in Section 4.2.1
- Import your shapefile into the script and rename the table with the TS name, e.g. BareSoil, Vegetation, etc.
    - The name must match exactly how you defined it in Section 4.3.1
- Repeat this for each individual Class.
- ** If you import your training samples in this way, they will not appear on the Map as a visible geometry layer, but don't worry, they will still work!

4.3.3.3 *Collecting Training samples on the ground*

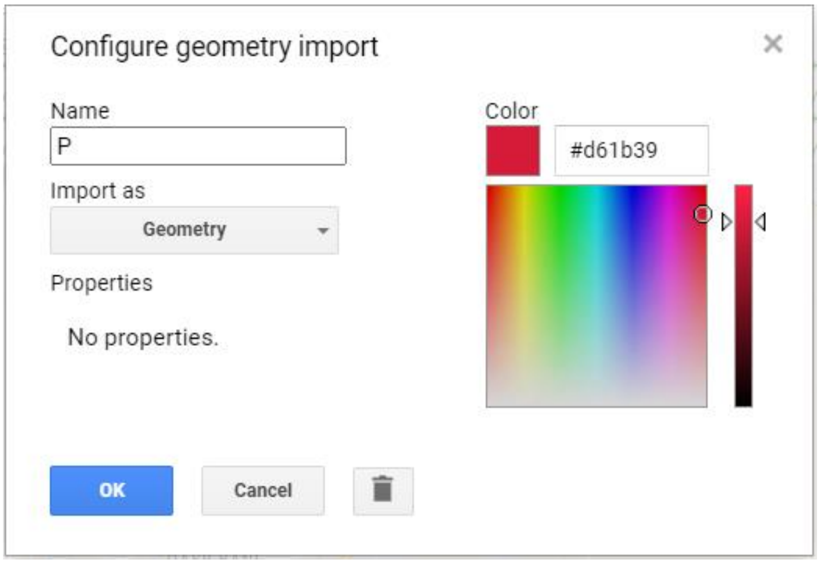
A third way is to collect training samples for each feature class on the ground using terrestrial land surveying techniques such as Differential Global Navigation Systems (DGNSS) or a GPS navigator.

- Collect points or polygons which are representative of each Class that you want to define on the ground with your preferred tool.
- Export the features for each class as a shapefile and follow the instructions in the previous section to upload them into GEE and your script.

4.4 Define location of Interest (P)

The EAMENA MLACD allows the user to define a specific location of interest to get further detailed results on its condition.

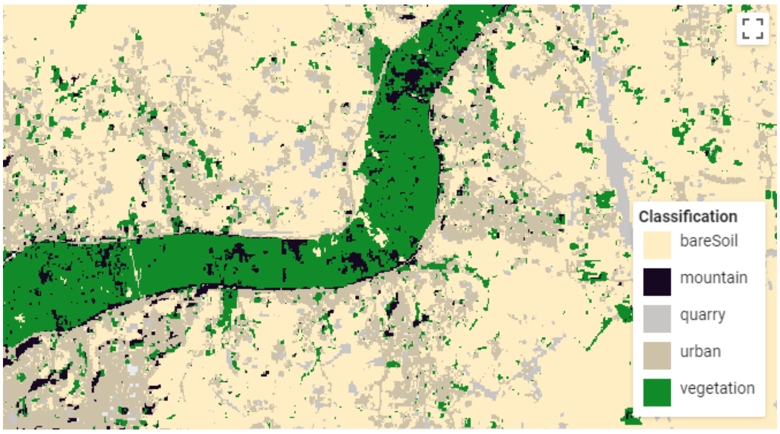
- You must create a new geometry layer using the Geometry tool "**P**".
- Use the Geometry drawing tools to draw one point or a polygon on the boundaries of the site of interest.
- This layer must contain one geometry feature, with the type of "**Geometry**".
- This defined location of interest can be any site or part of an archaeological site that requires a more detailed understanding of the changes and threats affecting it.



4.5 Visualisation Parameters: Colour Display

Our goal is to make land cover classifications which look something like below, with each class appearing as a different colour.

- In order to do that, we need to choose the colour visualisation for each class and set them in the next lines of the script.

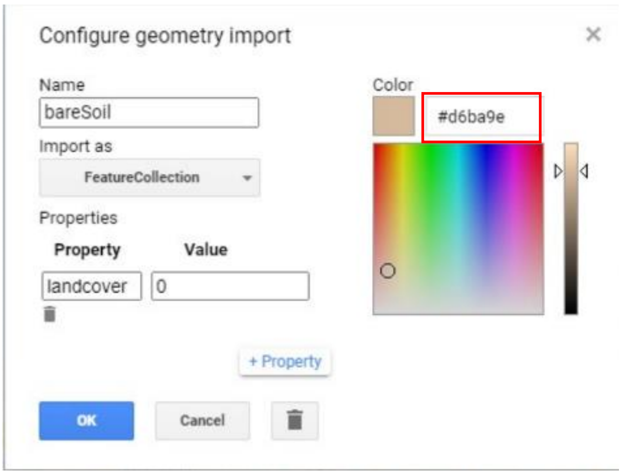


Find the following lines in Section 1 of the script, under 'Define Training Samples & Classification Variables'

```
// Define the color palette for each class
var classesPalette = ['#ffeec3','#170821', '#c7c6c5', '#cdc2a8', '#118b29'];
```

The colour for each Class is defined inside the brackets, in the same order as defined in previous sections, from Class 1 to Class 5.

- Each code in the example above represents a colour, which have been pre-defined for the Bani Walid example
  - See https://en.wikipedia.org/wiki/Web_colors for more information and examples
- You can also copy the colour name code from the geometry tool of each Training Sample Layer that you created.
- Add more colours, separated by a comma, if necessary depending on how many classes you have.



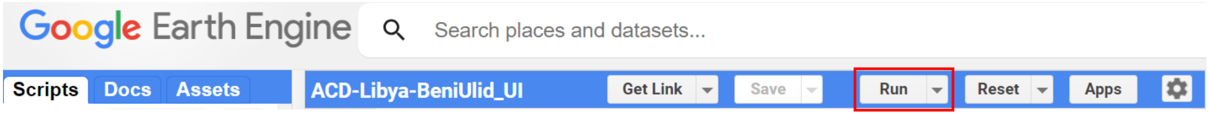
When you have set your colours, move to the next lines in the code.

- The visualization colours are automatically set based on the training samples colour identified in the 'classesPalette', so that the same visualisation colours used in the classification map are used for the chart outputs.
- In the Bani Walid example, these have already been defined.

5 Run the ACD

Once you have set all your variables, you can run the MLACD.

- Press the ‘Run’ button from the main code editor panel



A user interface will appear on the left side of the map where you must define the start and end date for the study period, in addition to a buffer value for the archaeological sites; if required.

5.1.1 Defining the start and end dates of the period under study:

Define the start and end date for your study period of interest in the following format: Year-Month-Day, e.g. 2020-01-21.

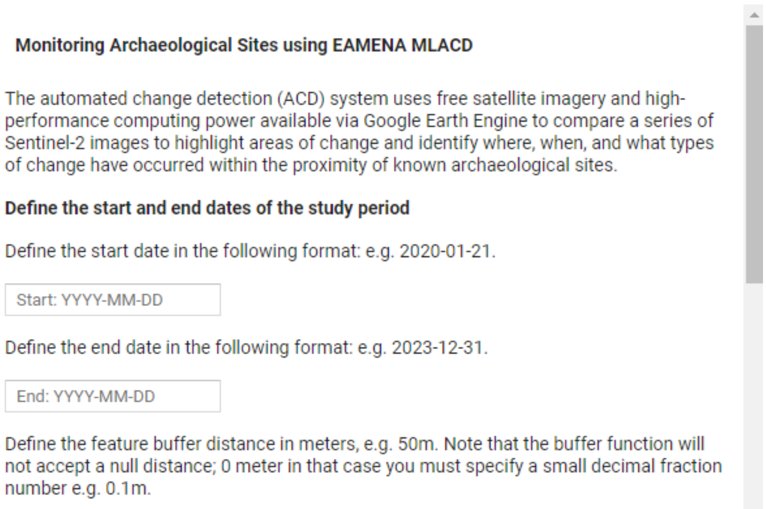
Important Note: The ACD is developed to process the atmospherically corrected harmonized Sentinel-2 Level-2A images so, the processing can only go back depending on the availability in the study area.

5.1.2 Define a buffer for the sites

Define the distance around each site where you want to run the analysis.

- Enter the distance in metres, e.g. 50 or 100
- Note that the buffer function will not accept a null distance, i.e. 0.
  - If your sites are already polygons and you don’t want a buffer, you must specify a small decimal fraction number e.g. 0.1.

Once the dates and buffer are defined you should press the first “**Run**” button in the user interface to execute the first stage of processing.



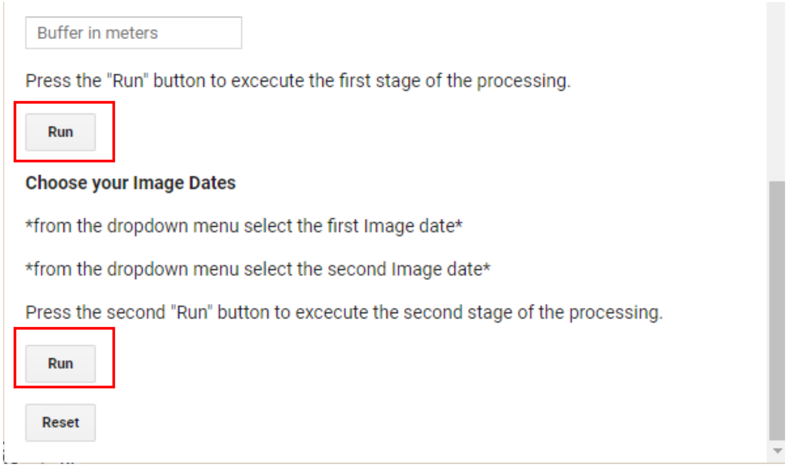


Figure 3. EAMENA MLACD User Interface

5.1.3 Select first and second images to be compared

Once you run the first stage, two drop down menus will appear that includes dates of the Sentinel-2 images acquired based on the start and end dates you have specified in the previous step (**Section 5.1.1**).

To monitor the changes between two specific dates you have to select the first image date and second image date.

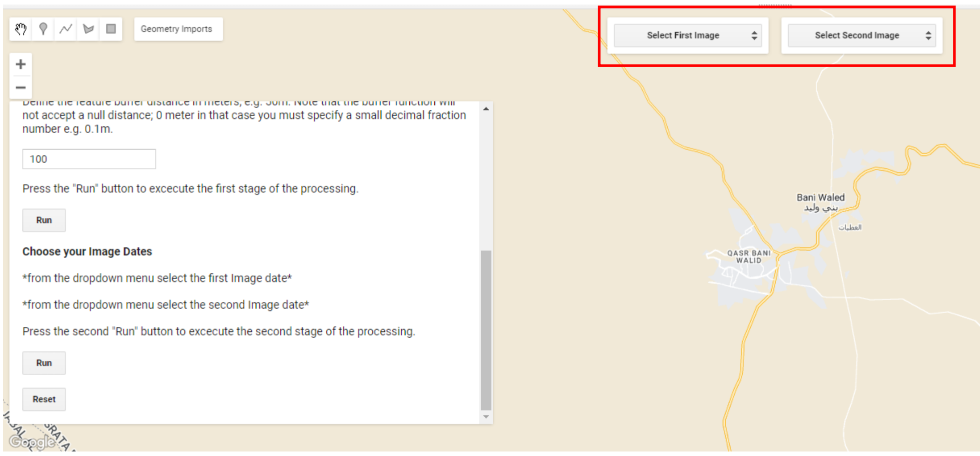


Figure 4. Select two images to compare for change detection

Once you have chosen the two images you want to compare for change detection, you must click on the second 'Run' button on the ACD user interface under 'Choose your Image Dates' (Figure 3).

6 Stage 2: Image classification and analysis

When you run the image classification and analysis you will get many different results and outputs including:

- land classification maps.
- change detection maps.
- classification time series.
- statistics on archaeological sites under threat and changes.

These results are displayed and viewed in the Layers tab and in the Console.

6.1 Results in the Layers Tab

From the layers tab in the Map panel, you can activate and view the different layers generated by the script (Figure 5).

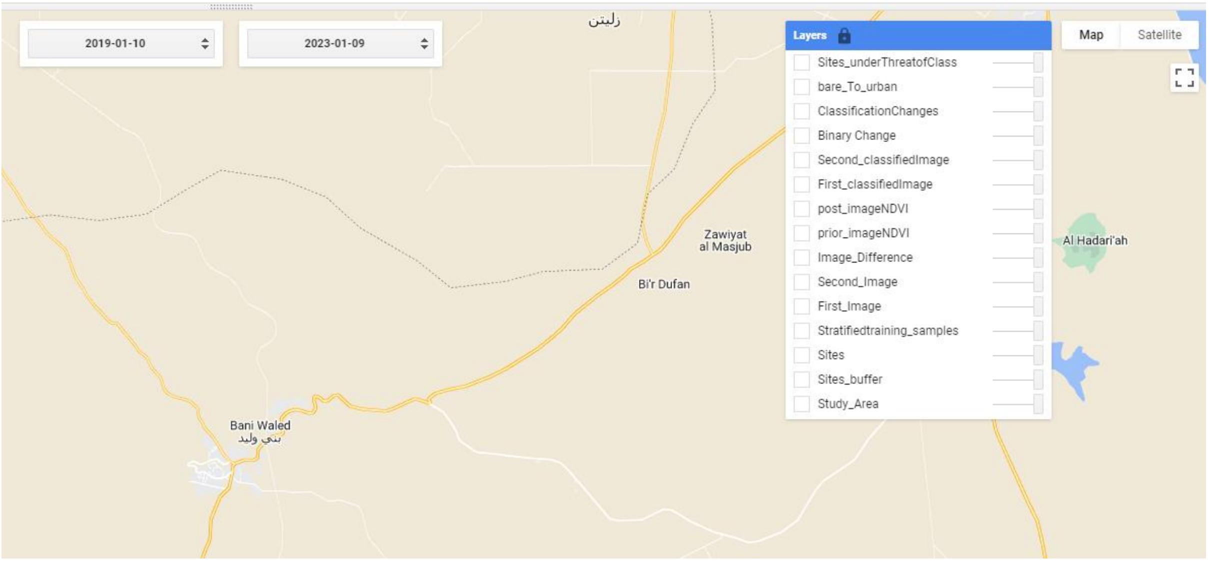
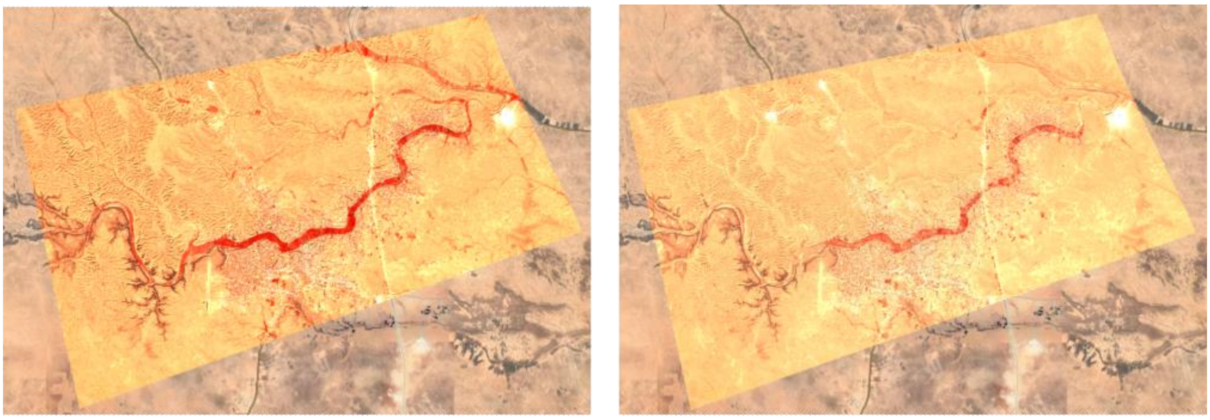


Figure 5. Layers on the map panel

This includes:

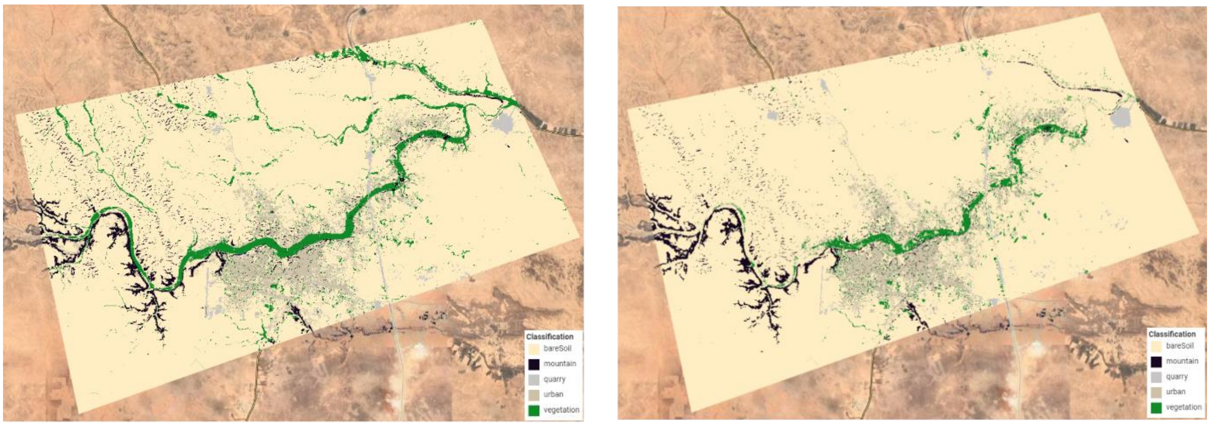
- the Sentinel-2 images for the two images you selected in the last section (Figure 6).
- classification maps for the two images you selected in the last section (Figure 7).
- a classification difference map between the two selected images (Figure 8).
- a classification change map, which assigns a new Change Class Value for each type of change from one class to another that is identified, e.g. a change from bareSoil to Vegetation (Figure 9).
    - These Change Class Values are calculated by assigning a new value to each possible combination of classes, as illustrated in the table below

Class in Image 1	Class in Image 2	Change Class Value	Change Class Label
C1	C1	1	bareSoil_to_bareSoil
C1	C2	2	bare_To_mountain
C1	C3	3	bare_To_quarry
C1	C4	4	bare_To_urban
C1	C5	5	bare_To_vegetation
C2	C1	6	mountain_To_bare
C2	C2	7	mountain_To_mountain
C2	C3	8	mountain_To_quarry
…	…	…	…



Sentinel-2 Image 10 Sep 2019 **Sentinel-2 Image 09 Sep 2023**

Figure 6. Sentinel-2 Images selected to detect the changes between them – First_Image and Second_Image in the Layers panel. For the visualization of each image the near infrared band (B8), Green band (B3) and Blue band (B2) were used, this explains the vegetated areas appearing as red.



Land Cover Classification Map 10 Sep 2019 **Land Cover Classification Map 09 Sep 2023**

Figure 7. Land cover classification maps for two selected dates *(*First_classifiedimage, Second_classifiedImage).

- **Note**: You might see gaps in your classified images as the image collection passes through a mask filter step that removes any cloudy and shadowy pixels from the image to reduce any misclassification.

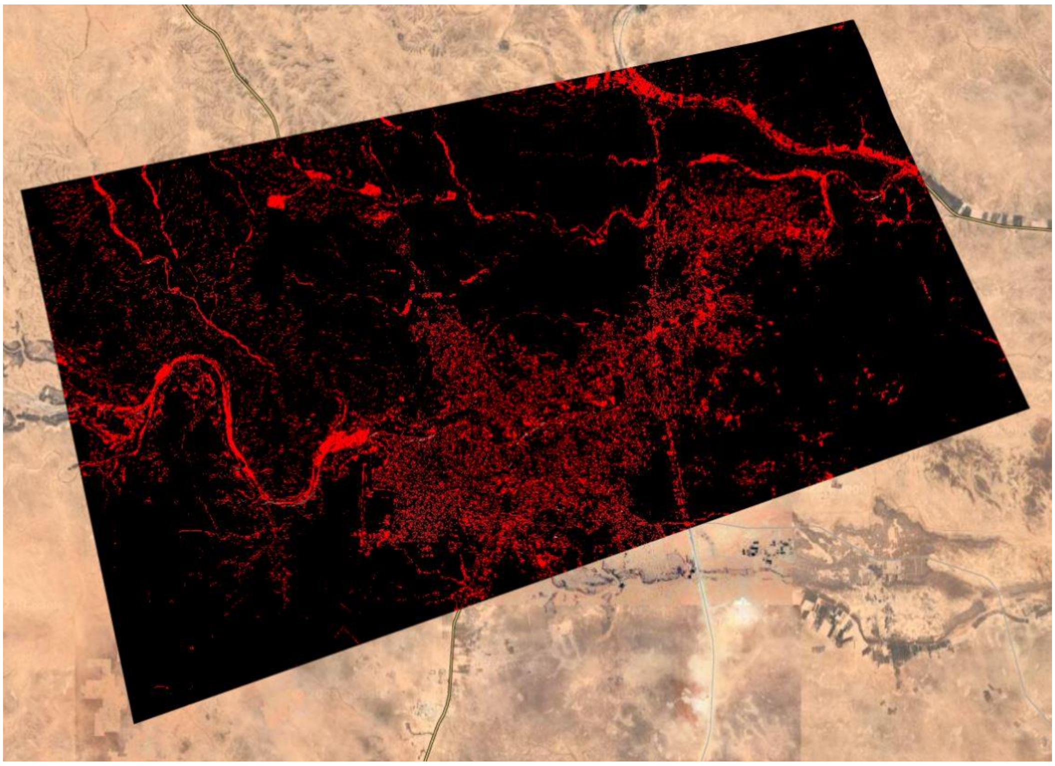


Figure 8. Binary change map. Black represents areas with no change and a binary value of (0), while red colour represents areas with a change with value of (1).

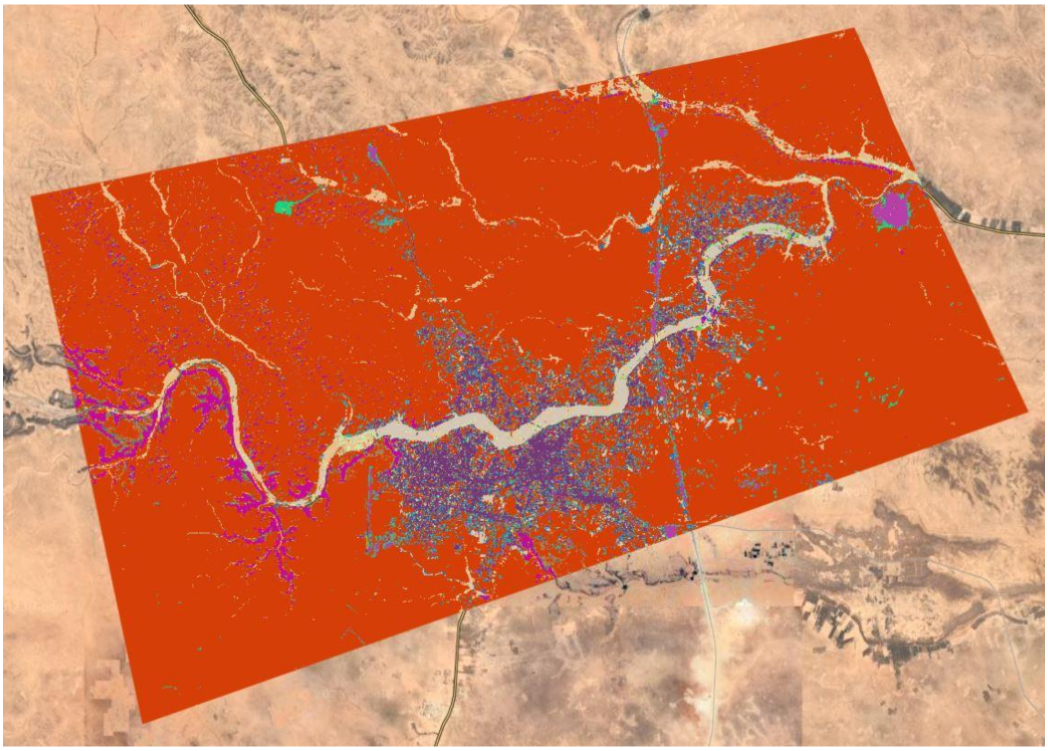


Figure 9. Classification Changes map demonstrating the changes in land cover features between the two selected images.

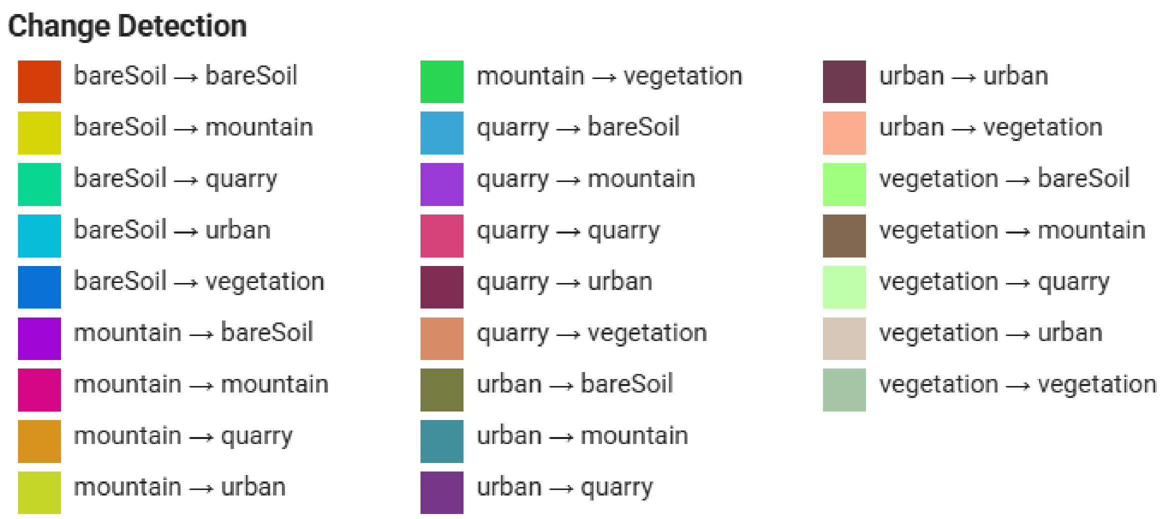


Figure 10. Legend for the change detection classification, showing the class transition labels and corresponding colours used to interpret the classification results. Each label represents a specific type of land cover change.

The script will also generate different raster layers which isolate different individual types of change or lack of change. For example, the image below presents a map of only those areas that were classified as vegetation in both images.

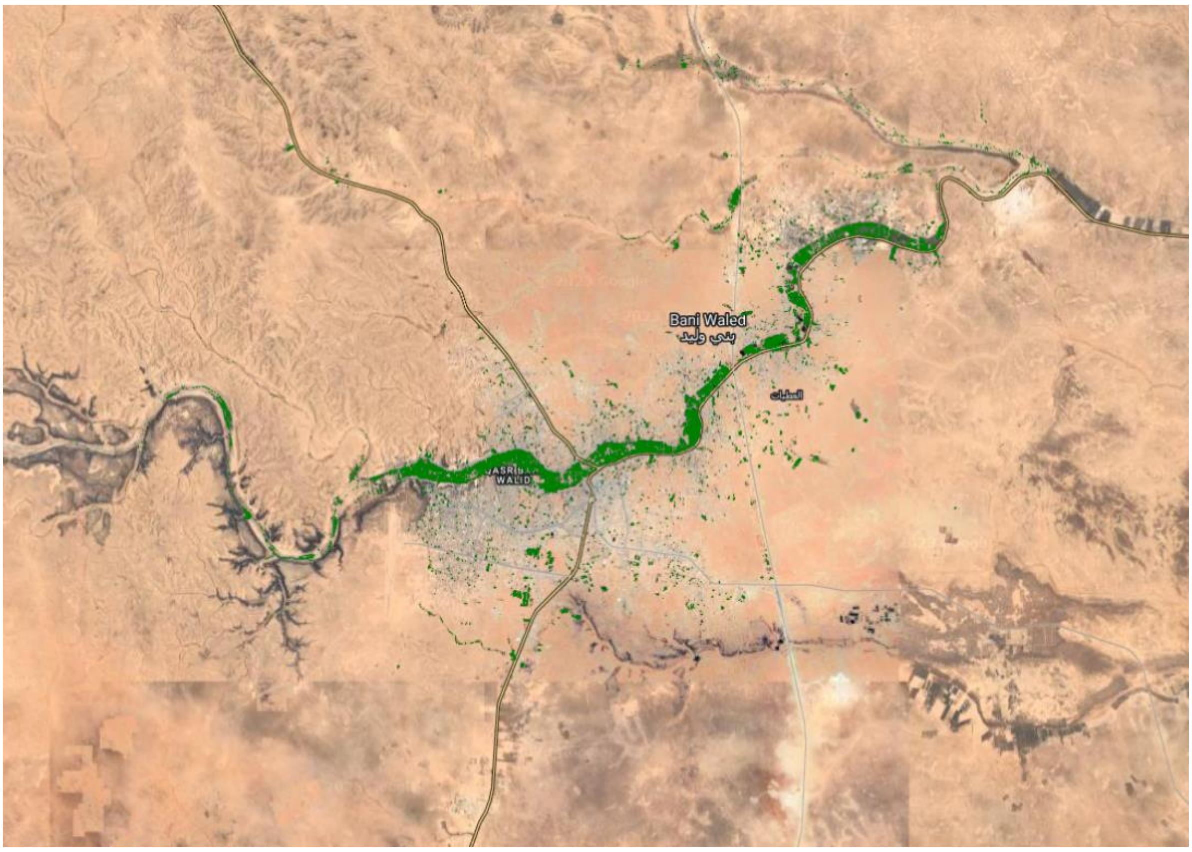


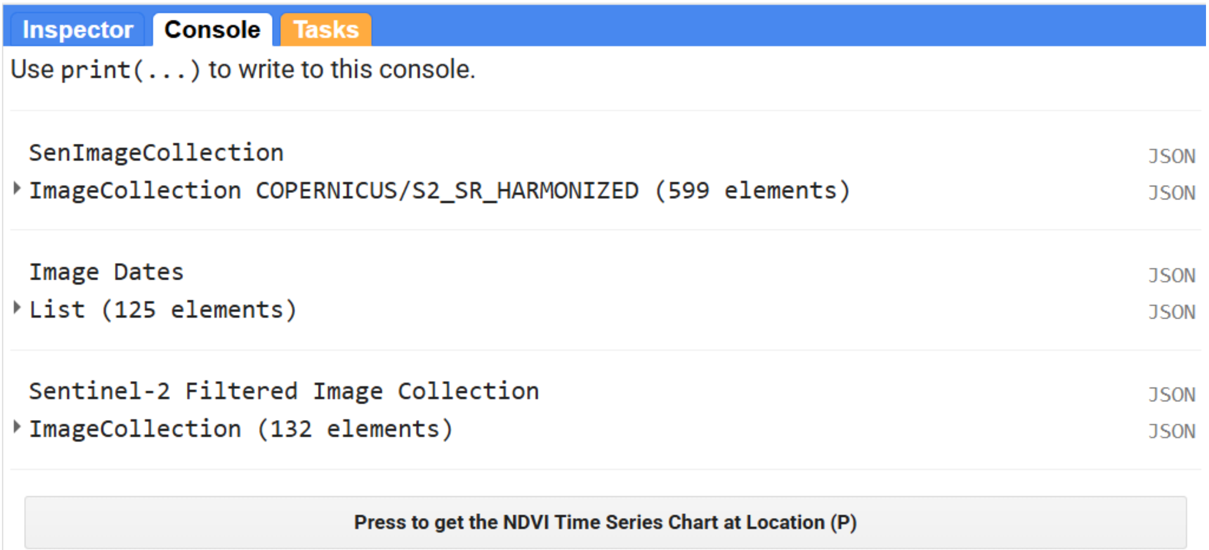
Figure 11. A raster layer that shows areas that remained as vegetation between the two selected dates.

6.2 Results on the Console

When you have run the script, several other buttons and information will appear on the console.

SenImageCollection lists all of the Sentinel-2 images and their properties which overlap with the study area and the time period specified.

Image Dates and **Sentinel-2 Filtered Image Collection** provides information about the date of acquisition and properties of the images used in the analysis, which have been filtered for quality and to make sure they cover the whole study area.



Press the **NDVI Time Series Chart** button to get an NDVI time series chart for a location of interest (P).

- Scroll to the bottom of the console to see a preview of the chart
- Click on this button ( ) at the top right of each chart and it will open a full view of the chart in a sperate Chrome Tab. This will provide a full view of the chart.
- You can click on any of the buttons at the top right of the chart in order to download the chart in different formats; CSV, SVG or PNG.

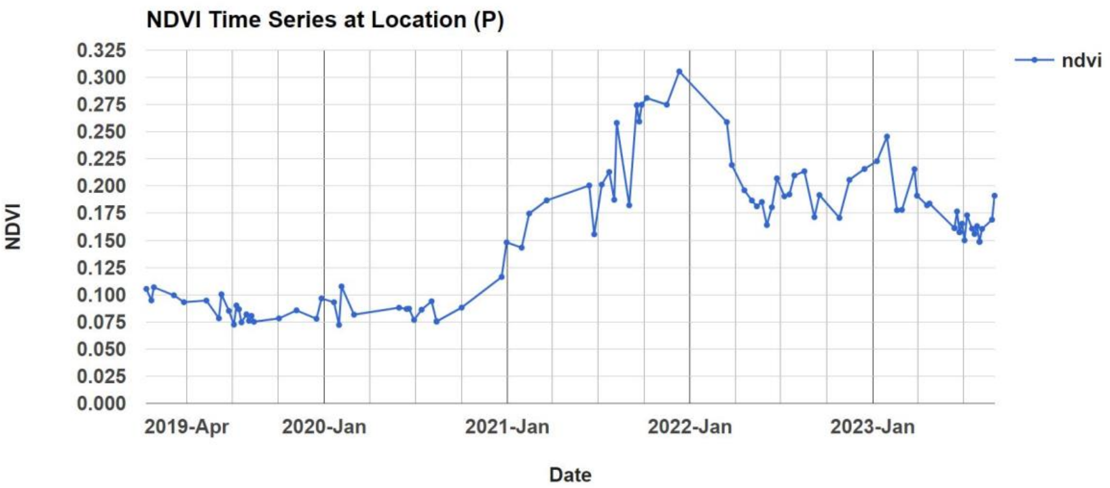
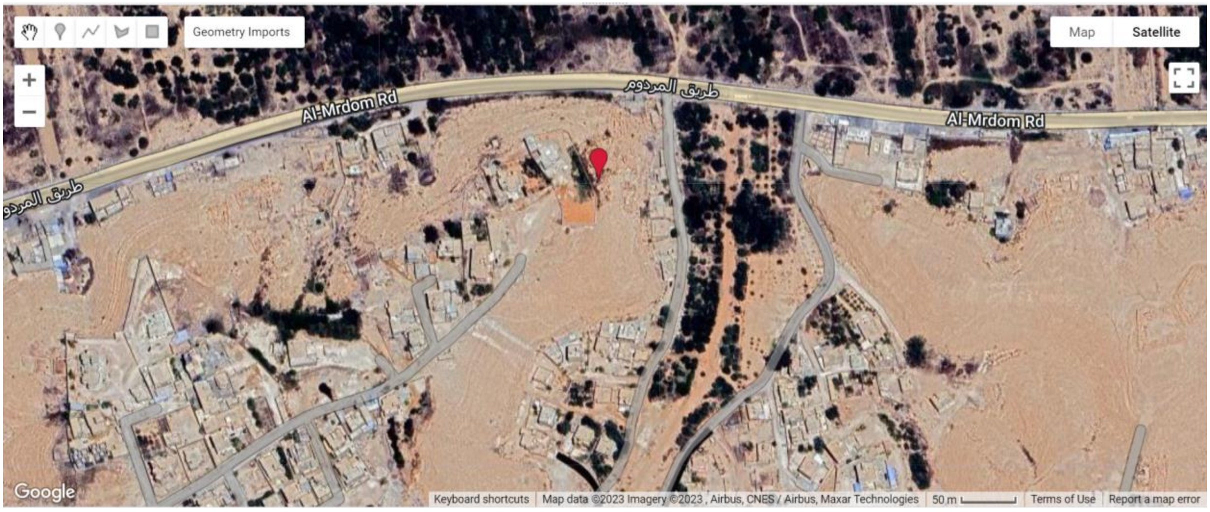


Figure 12. NDVI time series of site EAMENA-0189408 showing vegetation growth.

The location of point (P) has been pre-set in the Bani Walid example as a Geometry.

- Move it and re-run the script to get the results for a different location.
- This location could be either a point or a polygon depending on the area of interest. However, it Must be only one feature.



To assess and evaluate the performance of the supervised classification algorithm (i.e. Random Forest) several accuracy results are also generated on the console for the first (Prior) and second (Post) images selected for change detection:

- A confusion matrix is a table that summarizes the performance of a classification algorithm by comparing its predicted classifications to the actual ground truth. It consists of four key metrics: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).
- Overall accuracy which is a measure of how well a classification algorithm correctly identifies all classes within a dataset. It is calculated by dividing the total number of correctly classified pixels (TP + TN) by the total number of pixels in the dataset (TP + TN + FP + FN).
- User accuracy (Consumer accuracy) also known as precision, which measures the probability that a pixel or object classified as a specific class by the algorithm actually belongs to that class in reality. Meaning from the User perspective, how many of the pixels labelled as class X are actually class X. It is calculated as TP / (TP + FP) and assesses the accuracy of the classifier from the user's perspective. High user accuracy means you can trust that what is classified as for example "vegetation" actually is vegetation.
- Producer accuracy also known as recall, which assesses how well the classification algorithm correctly identifies a specific class in relation to the actual occurrences of that class in the ground truth data. Meaning from the Producer (classifier) perspective, how many of the actual class X pixels were correctly classified. It is calculated as TP / (TP + FN) and evaluates the accuracy of the classifier from the producer's perspective. High producer accuracy means most actual vegetation areas were correctly labelled.
- The F1 Score (or F-score) is the harmonic mean of User Accuracy (precision) and Producer Accuracy (recall). It provides a balanced measure that accounts for both omission and commission errors, offering a single indicator of classification performance for a given class.

```
First Classified Image Validation Overall Accuracy:                JSON
0.9195710455764075

First Classified Image Producer Accuracy:                          JSON
▸List (5 elements)                                                 JSON

First Classified Image Consumer Accuracy:                          JSON
▸List (1 element)                                                  JSON

First Classified Image Confusion Matrix:                           JSON
▸List (5 elements)                                                 JSON

First Classified Image F-Score:                                    JSON
▸List (5 elements)                                                 JSON

Second Classified Image Validation Overall Accuracy:               JSON
0.8698060941828255

Second Classified Image Producer Accuracy:                         JSON
▸[[0.6],[0.98],[1],[0.8958333333333334],[1]]                       JSON

Second Classified Image Consumer Accuracy:                         JSON
▸List (1 element)                                                  JSON

Second Classified Image Confusion Matrix:                          JSON
▸List (5 elements)                                                 JSON

Second Classified Image F-Score:                                   JSON
▸List (5 elements)                                                 JSON
```

Press to get the Land Cover Classification Areal Time Series

https://earthengine.googleapis.com/v1/projects/earthengine-legacy/vide… JSON

Press to get the Classification Time Series Chart for the Feature of Interest

Press the **Land Cover Classification Areal Time Series** button to see a chart which shows the change in area of different land cover classes over time.

- This chart provides statistical information about changes in areas covered by each class over time. This is valuable when measuring the expansion of urban areas or agricultural areas within the study area.

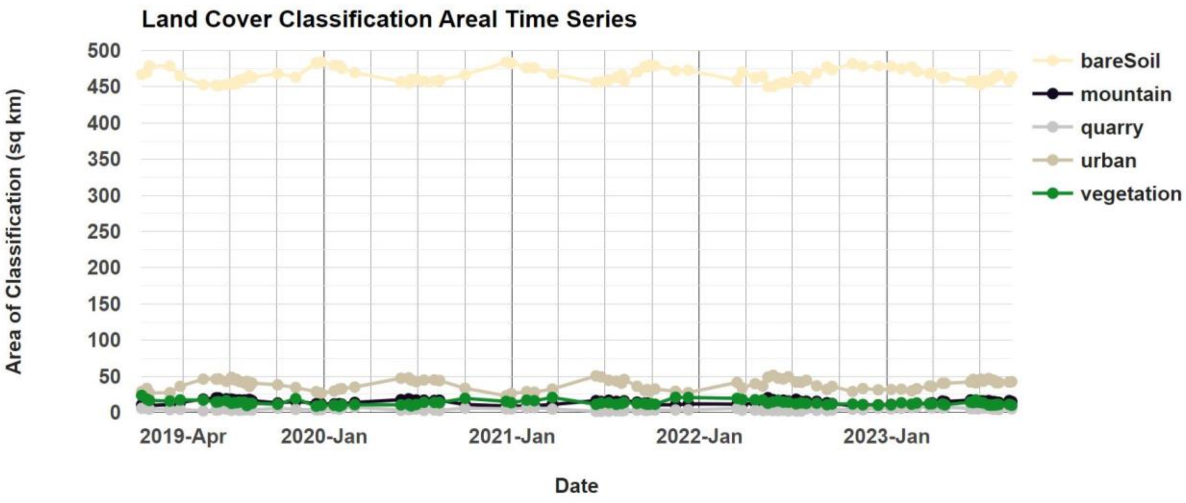


Figure 13. Land cover classification changes time series for Bani Walid

Click the **URL link** on the console to open an animated gif map showing all of the classified images over time: https://earthengine.googleapis.com/v1/projects/ee-ahmedmahmoud/videoThumbnails/99120ed9162b0e2115a397728e638e63-6f61b5deddfb02150b5a45336a47e625:getPixels.

Press the **Classification Time Series Chart for the Feature of Interest** button to print to the Console a chart that illustrates the classification time series for a location or feature at point (P) (discussed above) identifying the land cover class for that point and any changes that have occurred over time.

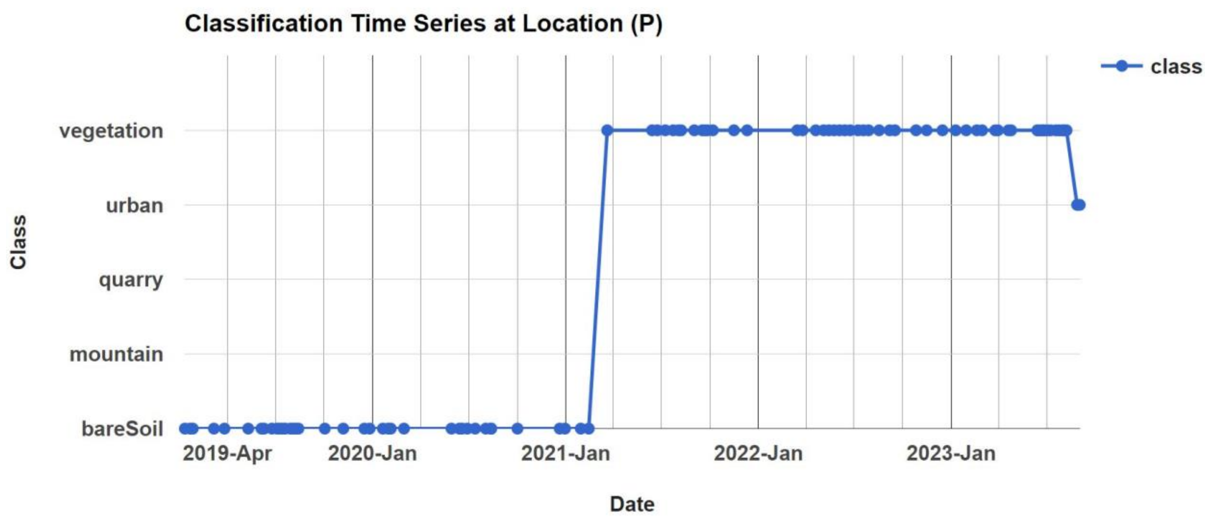
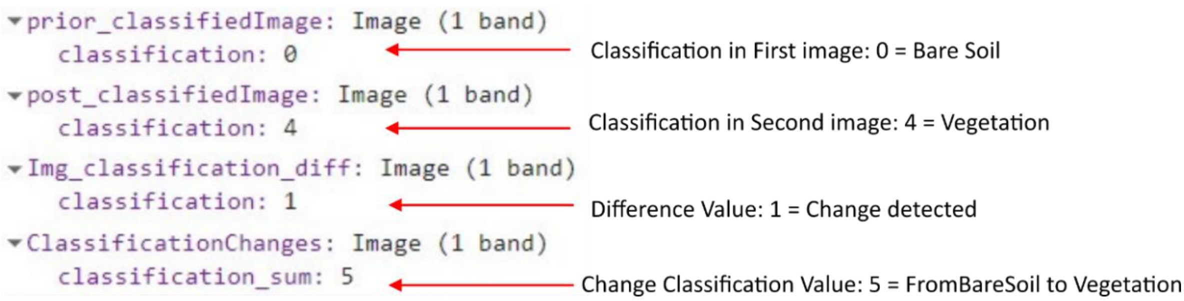


Figure 14. Classification time series of site (EAMENA-0189408) where (probably unintentional) vegetation expansion from a neighbouring farm was detected by EAMENA ACD

In the example above, we can see that sometime after Jan-2021, the classification at location (P), changed from bareSoil to Vegetation.

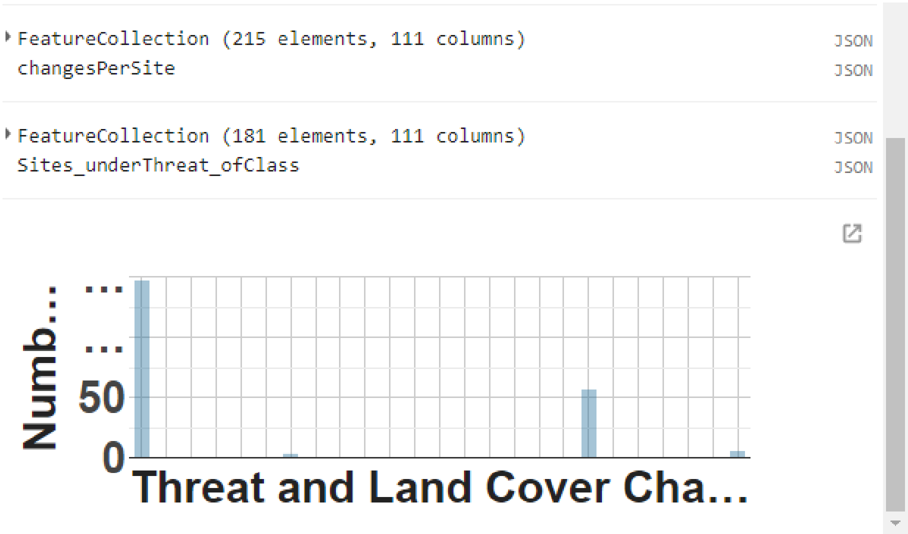
You can also view the statistical results of the generated layers by using the "Inspector tool" and clicking on the map on location (P) to inspect the layers' values in the chosen location.

For example, the results from the previous chart are as follows:



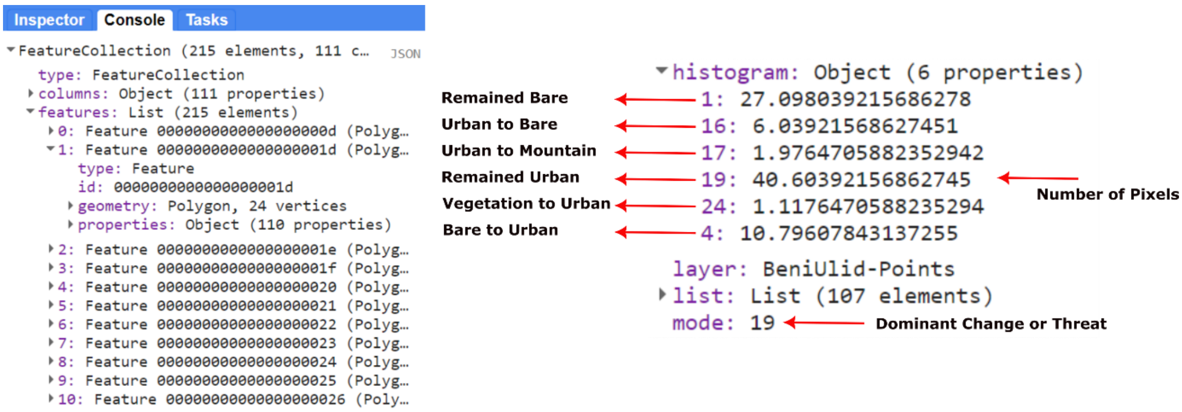
7 Stage 3: Identification of threats on archaeological sites

The next results in the console relate to statistical information about the pixel changes at each site. This change computation is the result of the classification change layer masked to each buffer site.



Under **Feature Collection: changesPerSite** you will see a list of each site analysed.

- For each site, under properties, you will find a histogram that shows you which changes have been detected within the boundary of the buffered archaeological site, and how frequently this change was identified within the buffer zone, measured in number of pixels.
- In addition, the most dominant change or threat within the buffer site will be identified by computing the "mode".



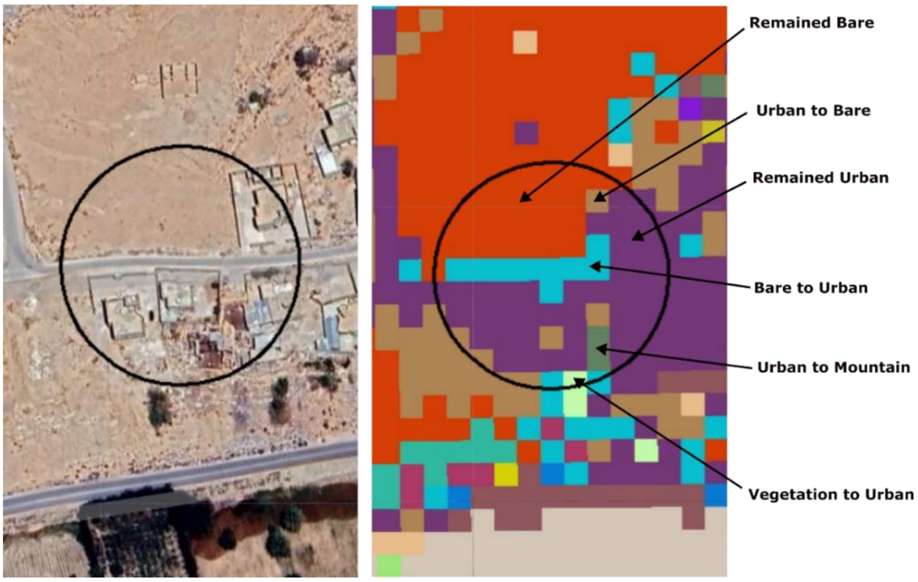
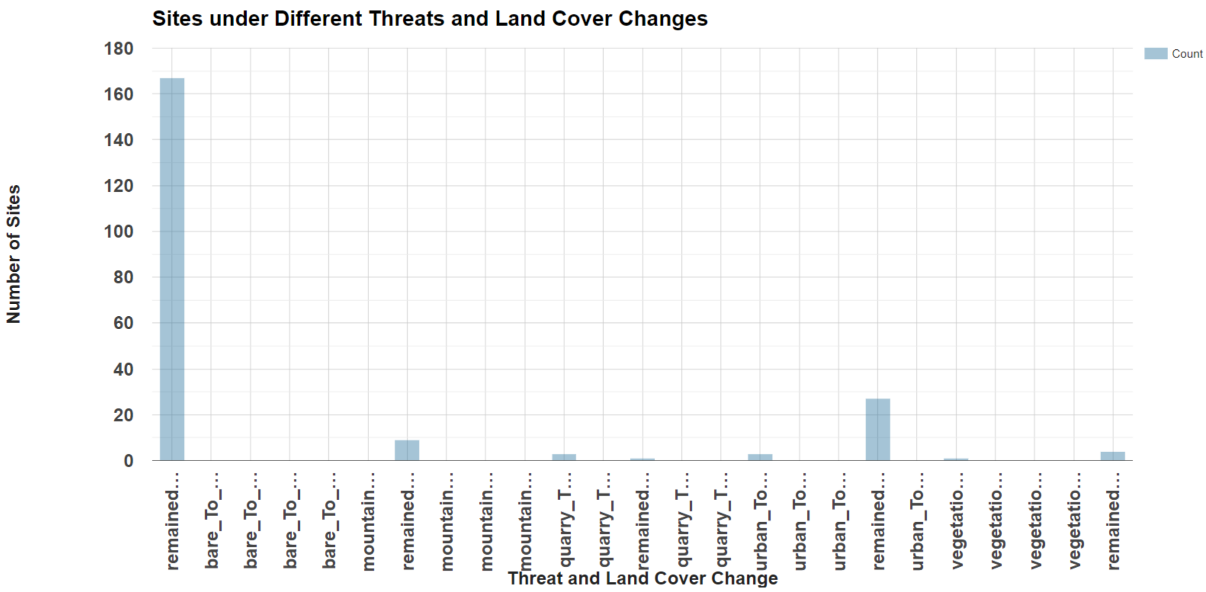


Figure 15. Change classification map in archaeological site EAMENA-0087287 in Bani Walid.

A histogram chart will also be generated on the Console which finds the most frequent Change Class found within each site buffer, and counts the number of sites by most common Change Class.

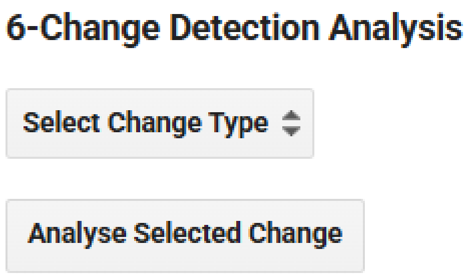
- This provides a general view of the most commonly experienced threat in all of the sites.



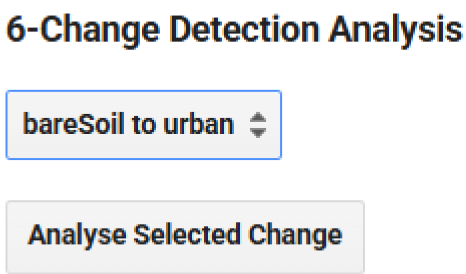
Sites by most common change category

Most importantly, the tool includes a dedicated feature for performing change detection analysis, which is activated by executing **Step 6: Change Detection Analysis**. After clicking the second Run button, a new widget will appear in the user interface, enabling users to select specific change types of interest. This functionality allows for a more focused investigation into how particular changes have impacted the archaeological sites.

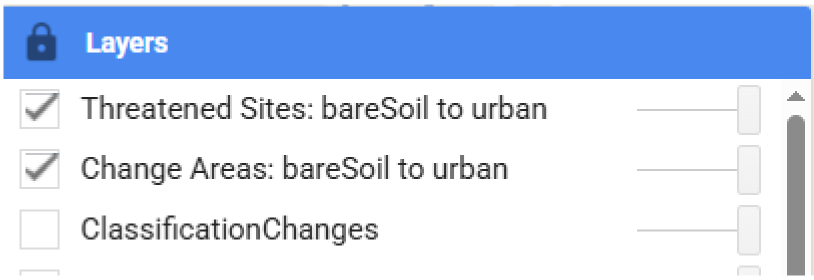
- Click on the '**Select Change Type'** icon and from the dropdown menu select the change type you want to investigate its threat; then click on the '**Analyse Selected Change'** button to execute the analysis.



The outcome is a masked image highlighting only the pixels corresponding to the selected type of change. Additionally, a separate layer is generated to display the archaeological sites affected by the specified land cover change or threat, based on the user's selection. For example, a user might choose the change type "bareSoil to urban" to identify areas that changed from bare soil to urban and assess how many archaeological sites were impacted by this change.



A change detection image displaying only the areas or pixels where the classification shifted from bareSoil to urban will appear in the Layers tab. In addition, a corresponding layer showing the archaeological sites affected by this specific change will also be added to the Layers tab. This layer can be used to examine the locations of impacted sites and support further on-the-ground analysis.



The number of sites affected by this change will be displayed in the Console panel. Click on FeatureCollection: Threatened Sites (bareSoil to urban) to view a list of all sites where at least one pixel of the selected change was detected within the site's buffer zone.

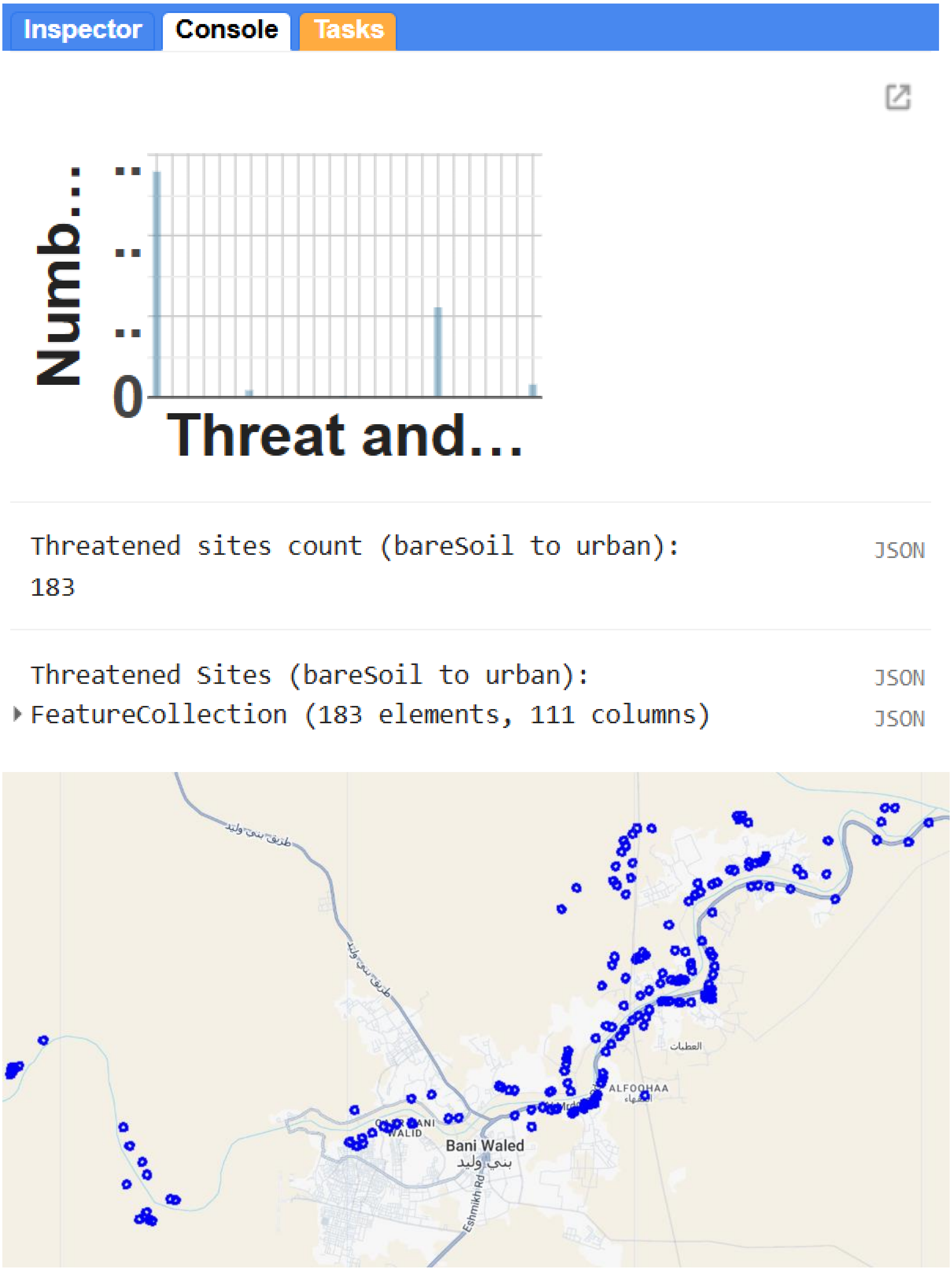


Figure 16. Archaeological sites with pixels values that changed from bare soil class to the urban class with classification change value (4).

To identify sites threatened by a different type of change, simply select a new change category in the Change Detection Analysis step and click the 'Analyse Selected Change' button again. The updated results will be displayed in the Layers tab and the Console, just as with the previous analysis.

8 Exports

From the Tasks tab on the GEE Code Editor, you can export all the features and rasters generated from processing; study area boundary, training datasets, training samples, test samples, classification maps, change classification maps.

- Once the export tasks appear on the Task Manager, press Run on each feature or raster you want to export.

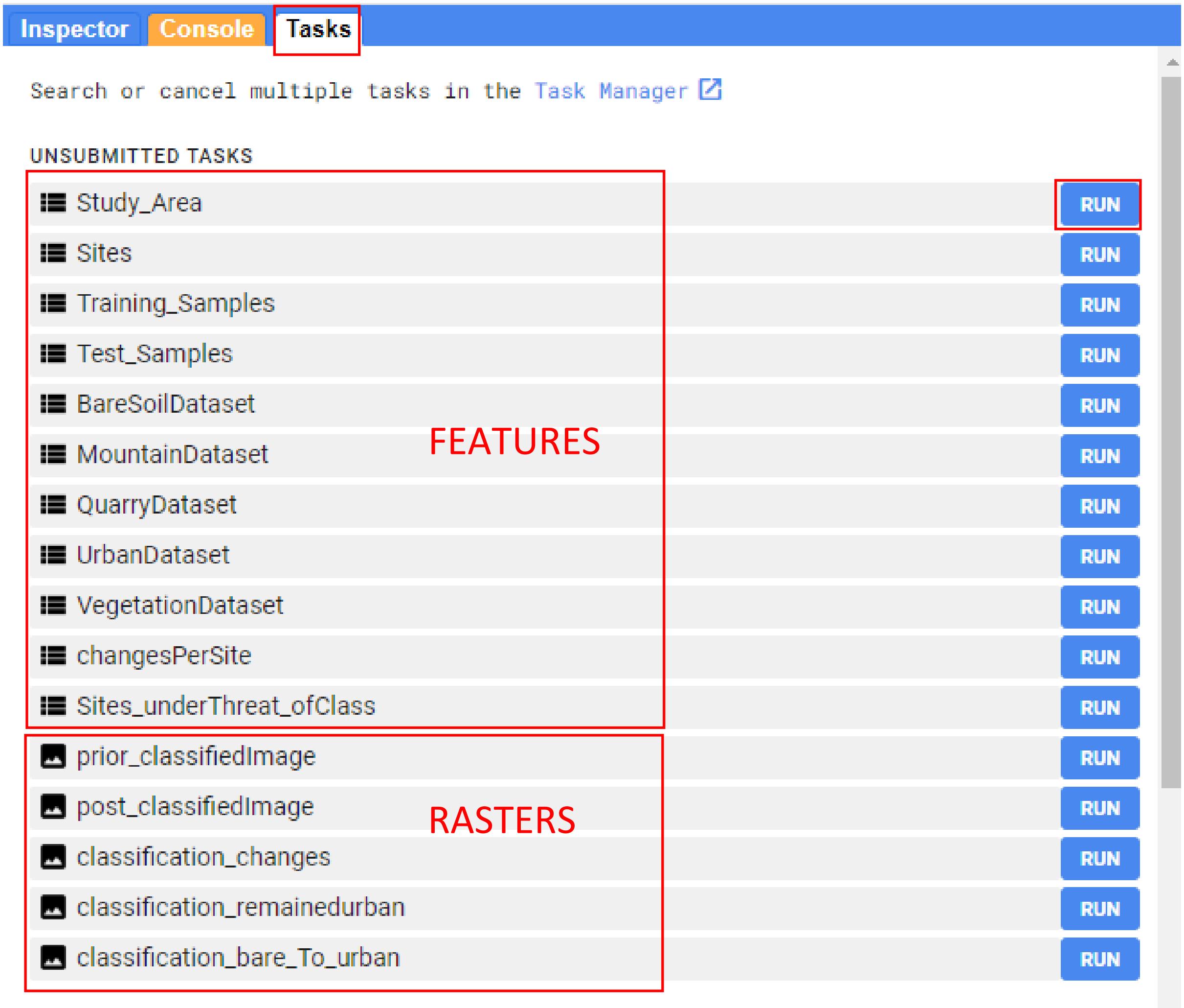
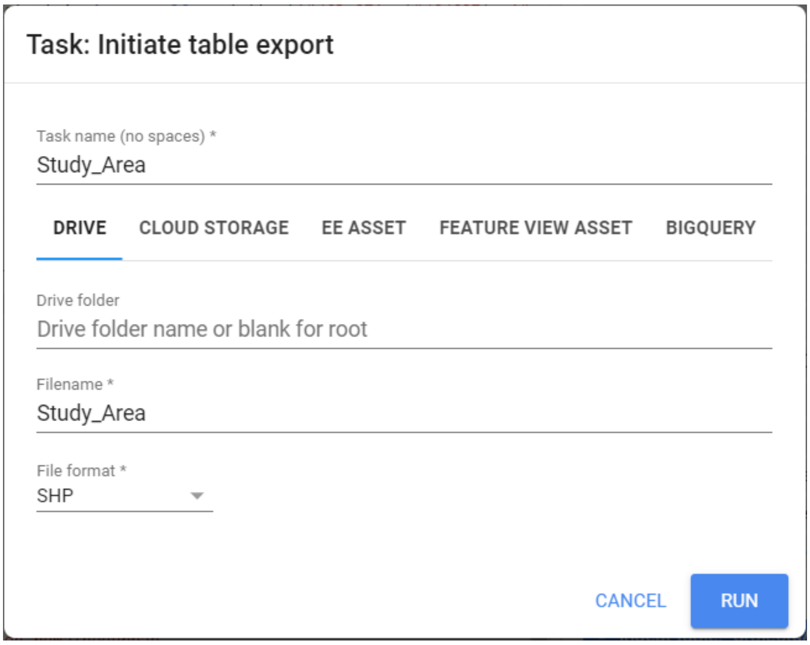


Figure 17. Export datasets from the code Editor Task Manager.

You can save your exported dataset in different locations (Google Drive, Cloud storage, EE-Asset…etc) depending on the work you would like to execute and the storage capacity permitted.

- You should specify the file export format if it is not already specified, then press Run.
- More detailed instructions for exporting files from GEE can be found in the ACD v1 Tutorial



9 Adaptation Stages of the EAMENA MLACD to New Case Studies

9.1 First stage of Adaptation: (Define Imports and Inputs)

1. Open the EAMENA MLACD Documentation.
2. In **Section 3** (**Getting Started**) click on the Bani Walid **URL code** to open in your browser.
3. Delete all the imports in code editor import tab from the Bani Walid Case Study.
4. Save the script as a new script with the name for your case study.
5. There are four inputs must be defined and imported so that the script can run properly which are the study area, archaeological sites, training datasets for all the land cover classes, and the location of interest (P).
6. Upload a shapefile of your **Study_Area** boundary or create a new one using the Geometry Tool in GEE. Follow the same instructions in **Section 4.1** to create the new study area.
7. Upload a shapefile for the archaeological **Sites** or use the Geometry Tool to create a new layer for the sites and its geometry must be defined as '**FeatureCollection'**. Follow the same instructions in **Section 4.2** to create the new study area.
8. As a preparation step you can visually inspect what type of land cover feature you can distinguish in your new study area, and then identify the different classes or features in your study area.
9. Add or collect the training samples for your case study (e.g., Bare, Buildings, Trees, Sand, Water). Follow the same instruction in **Section 4.3.3** to define the new training samples.
10. Create or add a geometry layer for the location of interest (**P**). Follow the instruction in **Section 4.4**.

9.2 Second Stage of Adaptation (Adaptation of the Variables in the Script)

11. Modify and edit the training samples and classification variables definition in the main script, following the same instructions from **Section 4**.

```
// Define the training samples dataset of your case study
var TS1 = BareSoil; var TS2 = Mountain; var TS3 = Quarry; var TS4 = Urban; var TS5 = Vegetation; // Add
// Define variables for each land cover class to contain the names of the training datasets
var C1 = 'BareSoil'; var C2 = 'Mountain'; var C3 = 'Quarry'; var C4 = 'Urban'; var C5 = 'Vegetation';
// Define all training sets in an array
var trainingSets = [TS1, TS2, TS3, TS4, TS5];
// Define a Class Array which contain the training dataset names
var classArray = [C1,C2,C3,C4,C5];
// Define the colour palette for each class
var classesPalette = ['#ffeec3','#170821', '#c7c6c5', '#cdc2a8', '#118b29'];
```

- Now your new script is ready to be executed by following the steps from **Section 5** forward.

10 Advanced Editing on the Script

There are many more edits that expert remote sensing users can make to adapt the MLACD script to their area of interest, for instance:

- **Editing the cloud percentage**: you can change the cloud coverage assessment value to increase the number of accessed images. The smaller the value the fewer images collected, but increasing the cloud coverage value will result in selecting images with large amount of cloud coverage which can generate misclassification.

```
var SenImageCollection = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
                         .filterDate(startDateValue, endDateValue)
                         .filterBounds(Study_Area)
                         .filter(ee.Filter.lt('CLOUD_COVERAGE_ASSESSMENT',0.1));
```

- **Optimizing the Random Forest Classifier:** To enhance the accuracy of your training model and classified maps, adjust the Random Forest classifier's hyperparameters: Number of trees, Minimum samples per leaf node, Subsampling ratio.

```
// Train the Random Forest classifier using the combined training samples
var trainedClassifier = ee.Classifier.smileRandomForest({
  numberOfTrees: 100,
  variablesPerSplit: null,
  minLeafPopulation: 5,
  bagFraction: 0.6,
  maxNodes: null,
  seed: 0
}).train({
  features: monthlyTrainingSamples,
  classProperty: 'landcover',
  inputProperties: finalBands
});
```

- **Optimizing the quantity and distribution of training samples** across all land cover classes can significantly enhance classification accuracy by ensuring robust representation of spectral variability within each

```
//*********************************************************************************************************
// Merge the training samples into one feature collection
var sampleDatasetFC = ee.FeatureCollection(trainingSets).flatten();
// print('sampleDatasetFC', sampleDatasetFC);
// Convert the training samples feature collection into an image
var sampleDatasetFCImage = ee.Image().byte().paint(sampleDatasetFC, 'landcover').rename('landcover');
// Collect 100 training samples for each class using the stratified sample function
var StratifiedSampleDataset = sampleDatasetFCImage.stratifiedSample({
  numPoints: 1000,
  classBand: 'landcover',
  region: Study_Area,
  scale: 10,
  classValues: [0,1,2,3,4,5,6,7,8,9],
  classPoints: [100, 100, 100, 100, 100, 100, 100, 100, 100, 100],
  dropNulls: true,
  geometries: true
});
// print (StratifiedSampleDataset, 'StratifiedSampleDataset');
Map.addLayer(StratifiedSampleDataset, {}, 'StratifiedSampleDataset',false);
// Training/Validation Split
var StratifiedSampleDatasetRandom = StratifiedSampleDataset.randomColumn();
```