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

First version: October 2023

Updated version: September 2025

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

Machine Learning Automated Change Detection (MLACD)

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*).

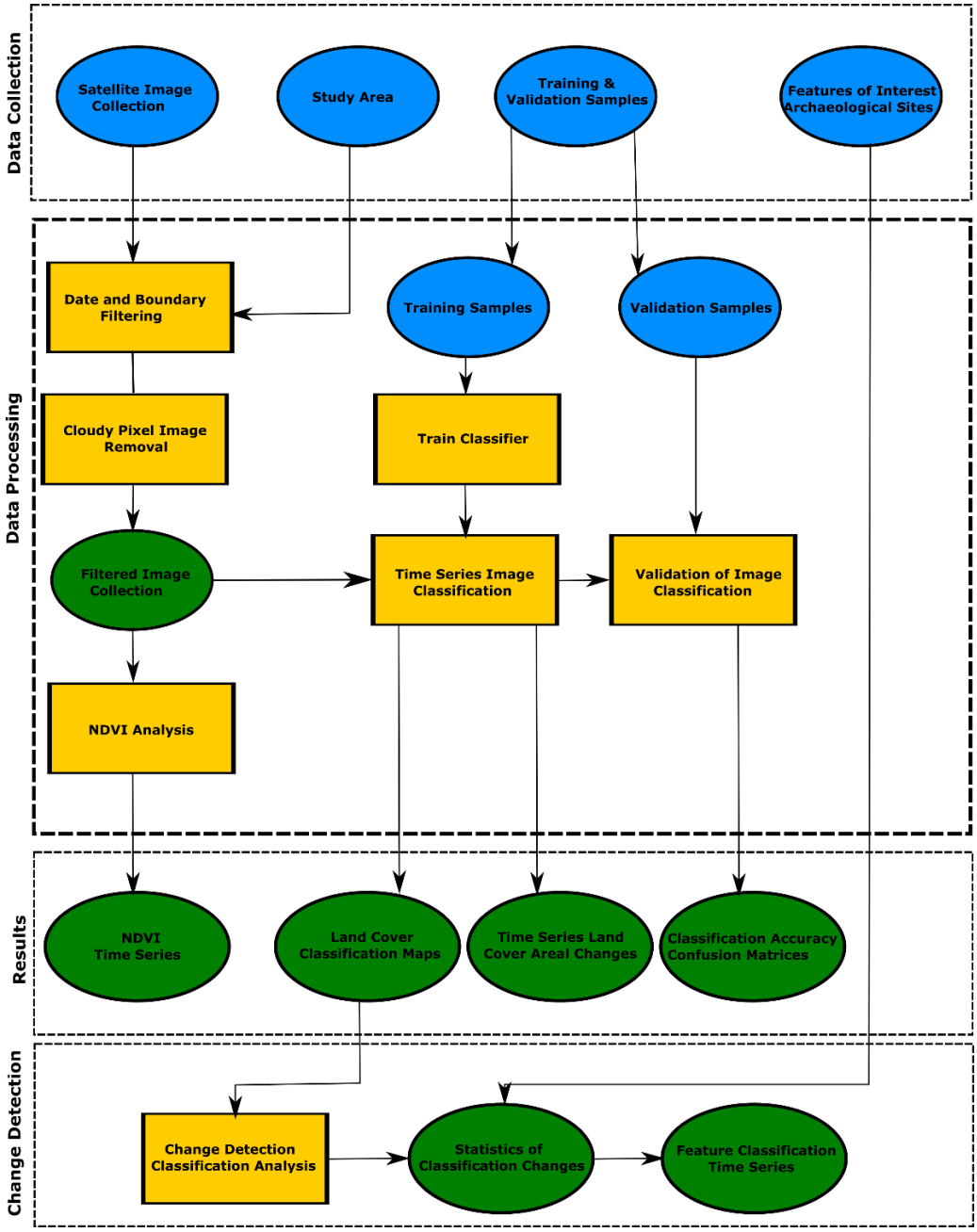


Figure 1. EAMENA MLACD Framework.

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

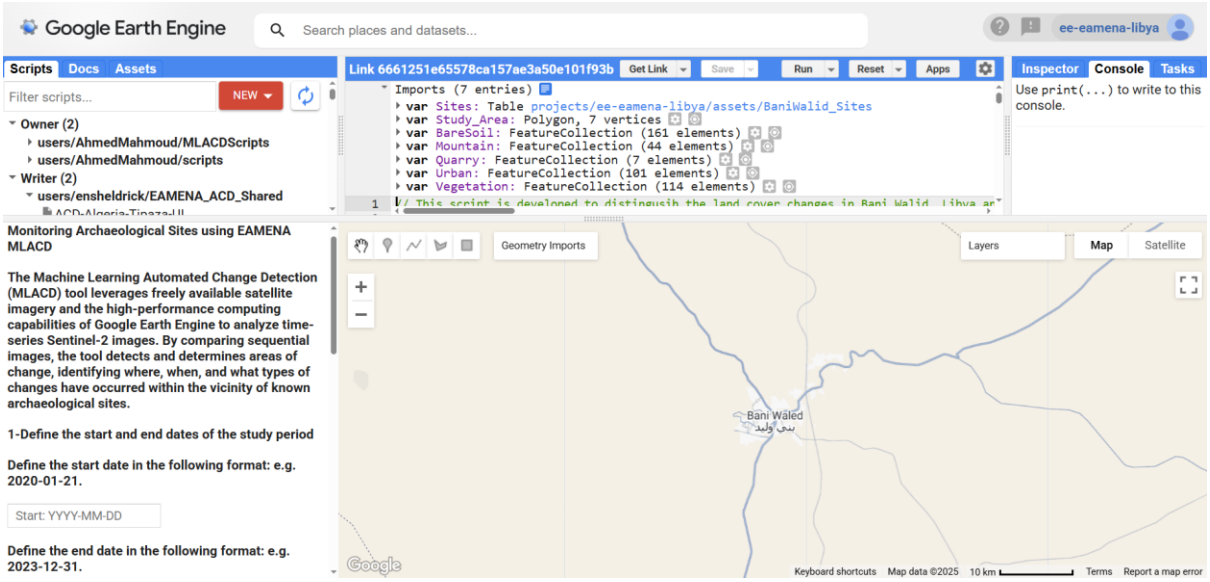
Once your application has been approved, go to https://code.earthengine.google.com/ to access the Code Editor where we will run the MLACD.

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

- Bani Walid: https://code.earthengine.google.com/22d556bc2f5aa5c822c4501de0ff2ba2.

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.

Note: if you want to modify the existing script and adapt it for a new case study, please refer to **Section 9 'Adaptation Stages of the MLACD to New Case Studies'**.

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. the visualisation parameters.
5. 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.

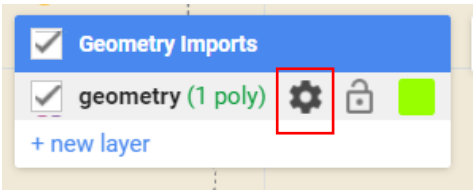
If you want to investigate a new area you will need to change this by deleting the existing Study_Area and entering your own study area using one of the two following methods.

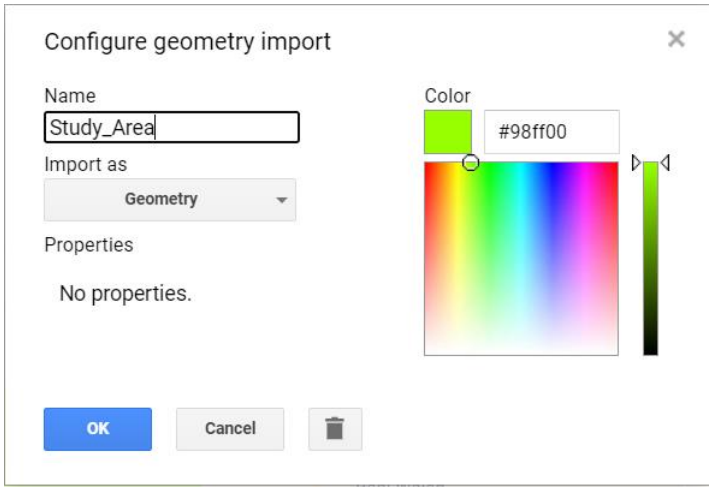
4.1.1 Method 1: Draw with the Geometry Tool

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.

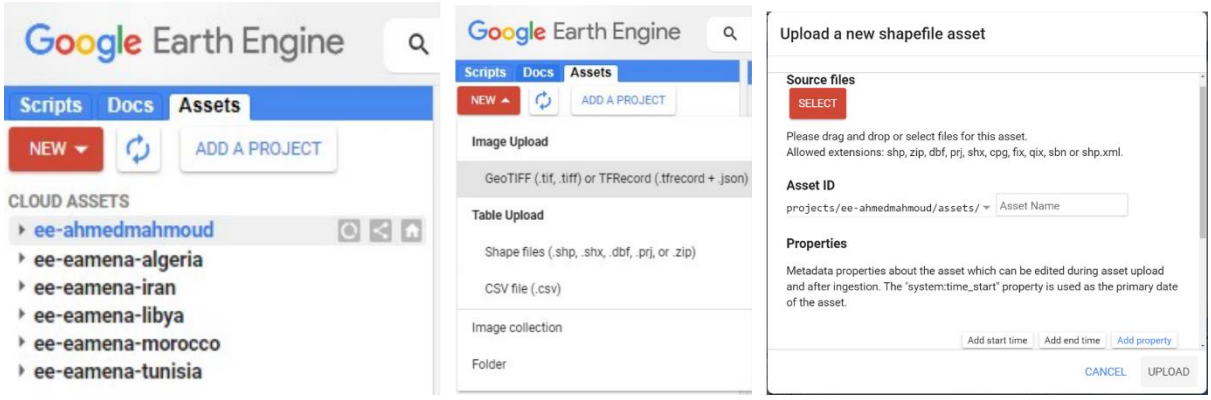
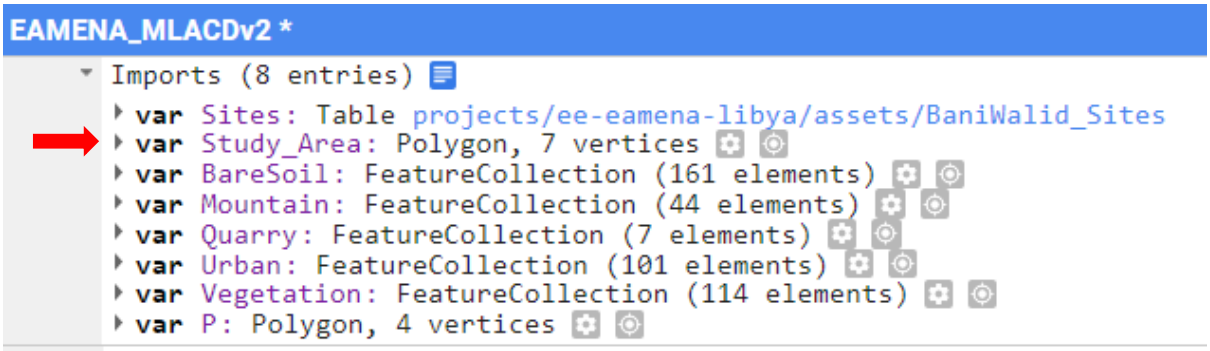


Figure 2. Adding assets in GEE projects

- Once it is 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**'



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

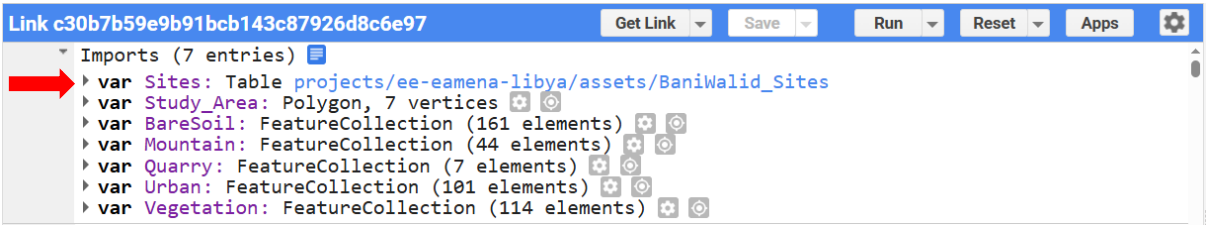
The pre-set scripts come loaded with a set of archaeological sites within the pre-defined study areas.

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

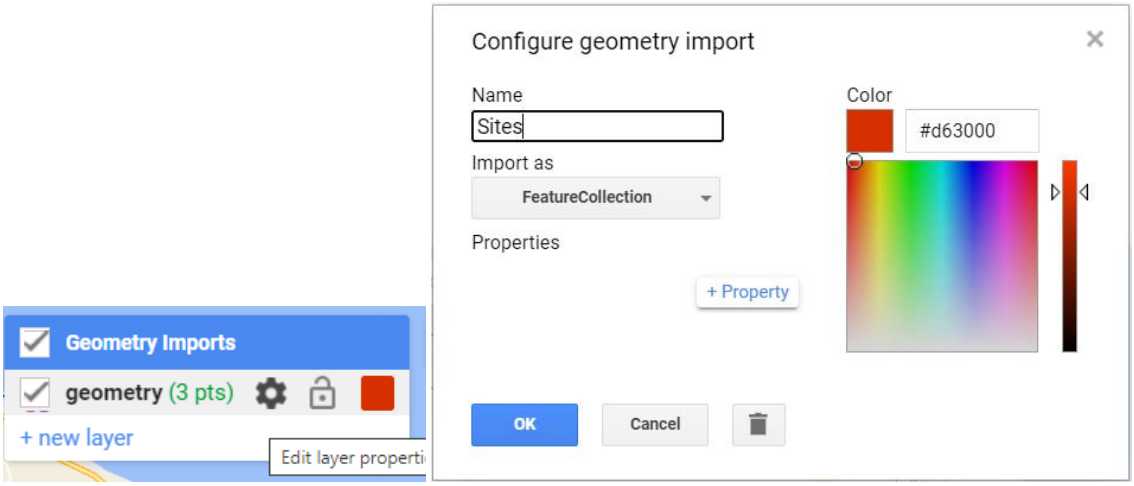
- Upload the shapefile as an asset and import it into your script using the same instructions you learned to upload the Study Area in Section 4.1.1
- Once you have imported the table to your script, you MUST change the name to '**Sites**'



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 use land cover 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 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 (Line 23-Line 29)**

- Training Samples can be created as polygons or points
    - **Polygons** are recommended here as they allow faster collection of large numbers of samples per 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 training samples define the **land cover classes** that the MLACD tool will use for classification. The first step is to decide, for your study area, **how many different classes of land cover are present** and which ones you want to classify. In the *Bani Walid* example, we defined **five classes**.

Each training sample must include:

1. A **dataset** of collected training samples; a FeatureCollection representing the land cover type (e.g., BareSoil, Mountain, Quarry, Urban, Vegetation). This can be:
    - A FeatureCollection imported into your script, or
    - A FeatureCollection created using the Geometry drawing tool in GEE.
2. A **colour**: a hex colour code (e.g., '#ffeec3') used for visualizing one class in the output maps.
    - Each code in the example above represents a colour, which have been pre-defined for the Bani Walid example
    - You can copy the colour name code from the geometry tool of each Training Sample Layer that you created, **see section 4.3.2.1**.
    - Or see https://en.wikipedia.org/wiki/Web_colors for more information on colours and examples.
3. A **label**: a descriptive name for the class (e.g., 'BareSoil', Mountain, Quarry, Urban, Vegetation). This label is used automatically in the classification and output legends.

```
22 // // Define the training samples datasets, colours and labels for each class in your case study
23 var trainingClasses = [
24   {TrainingSample: BareSoil,   color: '#ffeec3', label: 'BareSoil'},   // class 1
25   {TrainingSample: Mountain,   color: '#170821', label: 'Mountain'},   // class 2
26   {TrainingSample: Quarry,     color: '#c7c6c5', label: 'Quarry'},     // class 3
27   {TrainingSample: Urban,      color: '#cdc2a8', label: 'Urban'},      // class 4
28   {TrainingSample: Vegetation, color: '#118b29', label: 'Vegetation'}, // class 5
29 ];//Add and define additional training samples (e.g., TS: Water, color: '#2591ff', label: 'Water') or remove any class
```

- **How to define training samples in the script:**
  Training samples are organized in the trainingClasses array. For each class, add one entry using this structure:

```
{TrainingSample: Dataset1, color: '#HEXCODE', label: 'ClassName'},    // class 1
{TrainingSample: BareSoil,    color: '#ffeec3',  label: 'BareSoil'},  // class 1
```

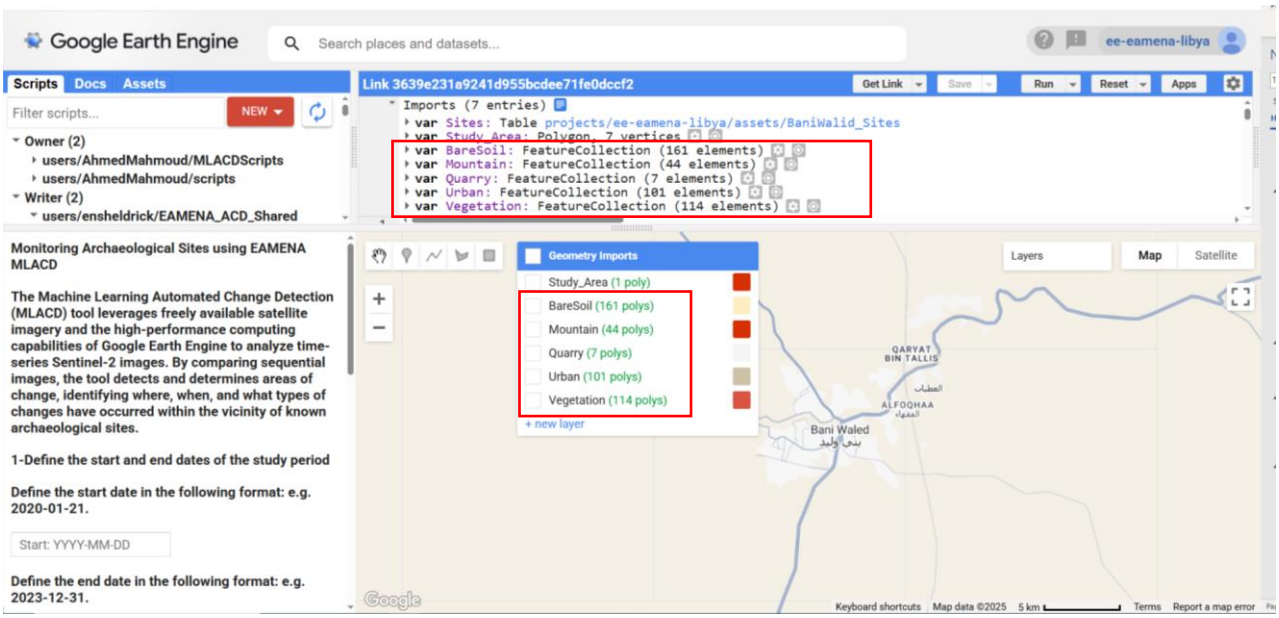
    - In this example, the BareSoil layer is defined as the first training sample dataset, with color '#ffeec3' and label 'BareSoil'.
    - You can add or remove classes depending on your case study.
    - To reduce the number of classes, simply remove a line (e.g., delete Class 5 if you only need four classes).
    - To add another class, copy a line and edit the dataset, colour, and label accordingly (e.g., create a Class 6 for 'Water').

- Once your training samples are defined, the script automatically generates the following arrays for classification:
    - **trainingSets** – contains all the FeatureCollections of the training samples datasets.
    - **classArray** – contains all class labels.
    - **classesPalette** – contains the visualization colours for each class.

4.3.2 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 dataset layers, that have the same names as defined above, e.g. BareSoil, Mountain, Quarry, Urban, Vegetation.



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.

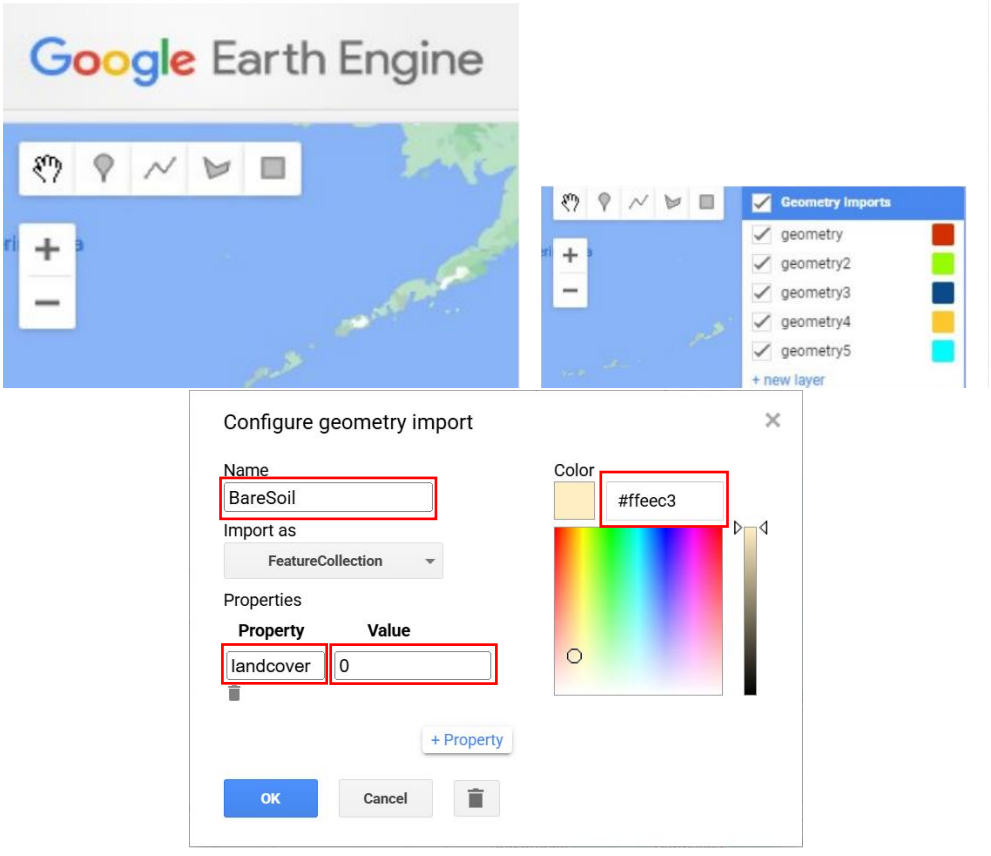
4.3.2.1 Collecting Training samples using the Geometry Tool in GEE

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 import tab 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 training sample dataset in the previous section, e.g. if **class 1** = BareSoil, your layer must also be named 'BareSoil'
  - You will need to make a separate layer for each Class dataset (e.g., BareSoil, Mountain, Quarry, Urban, Vegetation,..etc).
- Under **Import as**, select **'FeatureCollection'**, since your layer will contain multiple features.
- Click the **+Property** button and set:
  - Property name: '**landcover'** (use lowercase letters, as this name will be referenced later in the script).
  - Class Value: assign a numeric code to each feature (e.g., 0, 1, 2, 3, 4 …). For example: if 'BareSoil' is Class 1, assign it the value 0. Class 2 = 1, Class 3 = 2, and so on.
- Choose a different colour for each class
- Click OK



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

- There is no fixed rule for the number of polygons required – it depends on the size of your study area and how much of it is likely to be classified under that class.
  - For example, in the Bani Walid case, Class 2 (Mountain) has only 44 polygons, while Class 1 (Bare Soil) has 161.

Tips for drawing polygons:

- Aim to capture several *representative examples* of the class you are defining.
- Vary polygon sizes, but avoid making them too large—every pixel within a polygon should clearly belong to the chosen class.

- Distribute polygons across the entire study area to ensure good spatial representation of the training sample.
- Use high-resolution imagery in GEE that corresponds with the Sentinel-2 imagery year you selected for training data, and cover *all* target classes.
- Remember that the quality of training samples strongly influences classification accuracy. Careful collection and validation will result in more reliable outcomes.

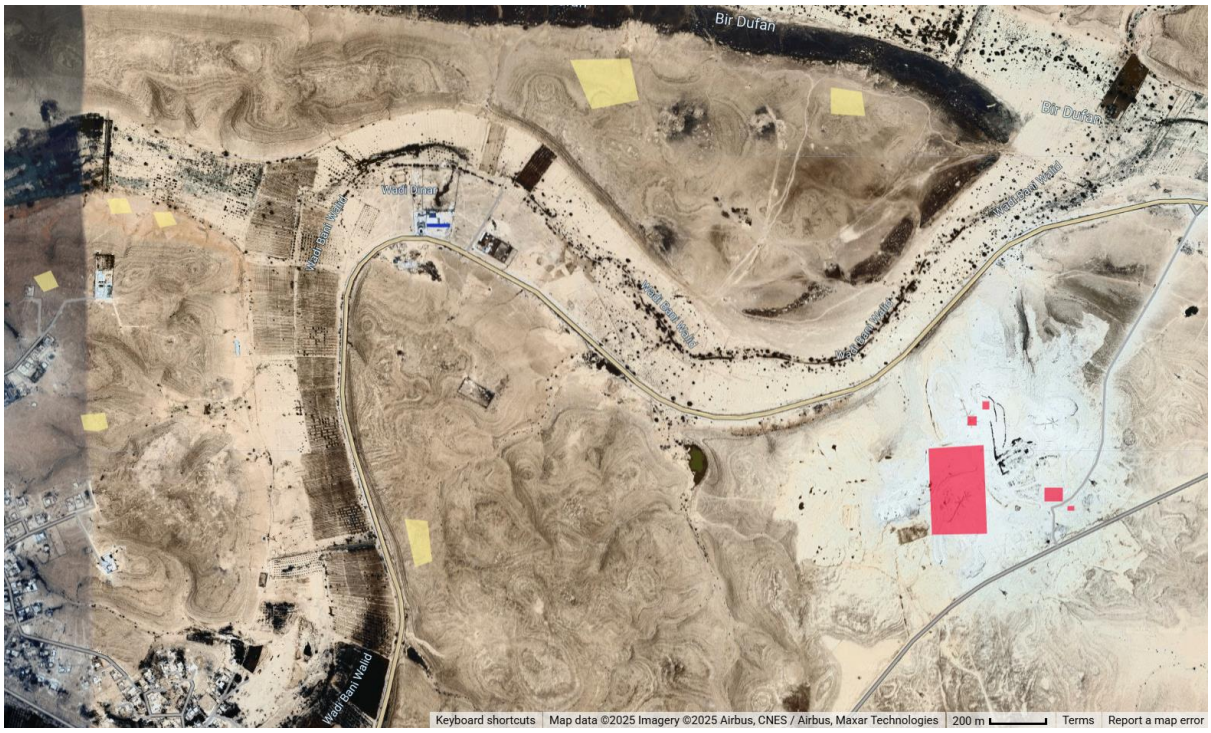


Figure 3. Example of collected training samples for BareSoil (yellow) and Quarry (red) areas representing the classes of Bare Soil and Quarrying activity in Bani Walid region in Libya.

- To improve the accuracy of your collected training sample datasets, you can export each dataset created in GEE (see **Section 8, *Exports***) and refine it using high-resolution imagery in Google Earth Pro. Using the historical imagery panel to visually verify each polygon and adjust any that do not correctly represent the intended land-cover class.
- The accuracy check process involves:
  1. Importing the exported shapefile of each training sample dataset into Google Earth Pro.
  2. Editing polygon boundaries through the Properties option.
  3. Saving the edited dataset for each class as a new KML file.
  4. Opening the KML file in QGIS.
  5. Re-exporting it as a shapefile in QGIS.
  6. Uploading the corrected shapefiles back into Google Earth Engine Assets.
  7. Replacing the original GEE-created training sample datasets with the verified versions.

4.3.2.2 Collecting Training samples using Google Earth Pro

An alternative way to collect training samples is to draw polygons in **Google Earth Pro** using high-resolution archived imagery, which can improve classification accuracy.

- Save the polygons for a single land cover class in a folder and export the folder as a .kml
- Import the **.kml** file for that class into **QGIS** and re-export it as a **shapefile**.
- In QGIS, create a new field named '**landcover'** to store the class value for each polygon.

- Example: if working on the *BareSoil* shapefile as your first class, assign the value **0** to all polygons in the '**landcover'** field. Class 2 would have the value **1**, Class 3 = **2**, and so on.
- Upload the shapefile of your training samples to your **GEE Assets** using the method described in Section 4.2.1.
- Import the shapefile into your script and rename the table in the script **Imports** according to the training sample dataset name (e.g., *BareSoil*, *Vegetation*).
- The name must match exactly how it was defined in Section 4.3.1.
- Repeat this for each individual class.
- Note: Training samples imported this way will **not appear as visible geometry layers on the Map** in GEE, but they will still function correctly in your workflow.

4.3.2.3 Collecting Training samples on the ground

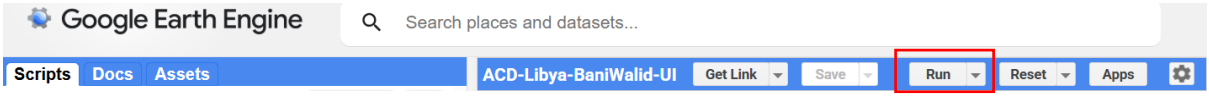
A third way is to collect training samples for each feature class on the ground using terrestrial land surveying techniques such as Differential Global Navigation Systems (DGNSS) or a GPS navigator.

- Collect points or polygons which are representative of each land cover 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.

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 to execute your script.



A user interface will appear on the left side of the map, where you need to define the start and end dates for the study period, as well as a buffer value for the archaeological sites (if required).

The tool's user interface is organized into eight main steps that users must follow to effectively assess threats and monitor the predefined archaeological sites.

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

Step 1: 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 MLACD is designed to process **atmospherically corrected, harmonized Sentinel-2 Level-2A images**. The processing is limited by data availability in your study area and cannot extend earlier than **2017.**

5.1.2 Define a buffer for the sites

Step 2: 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.

Step 3: 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.

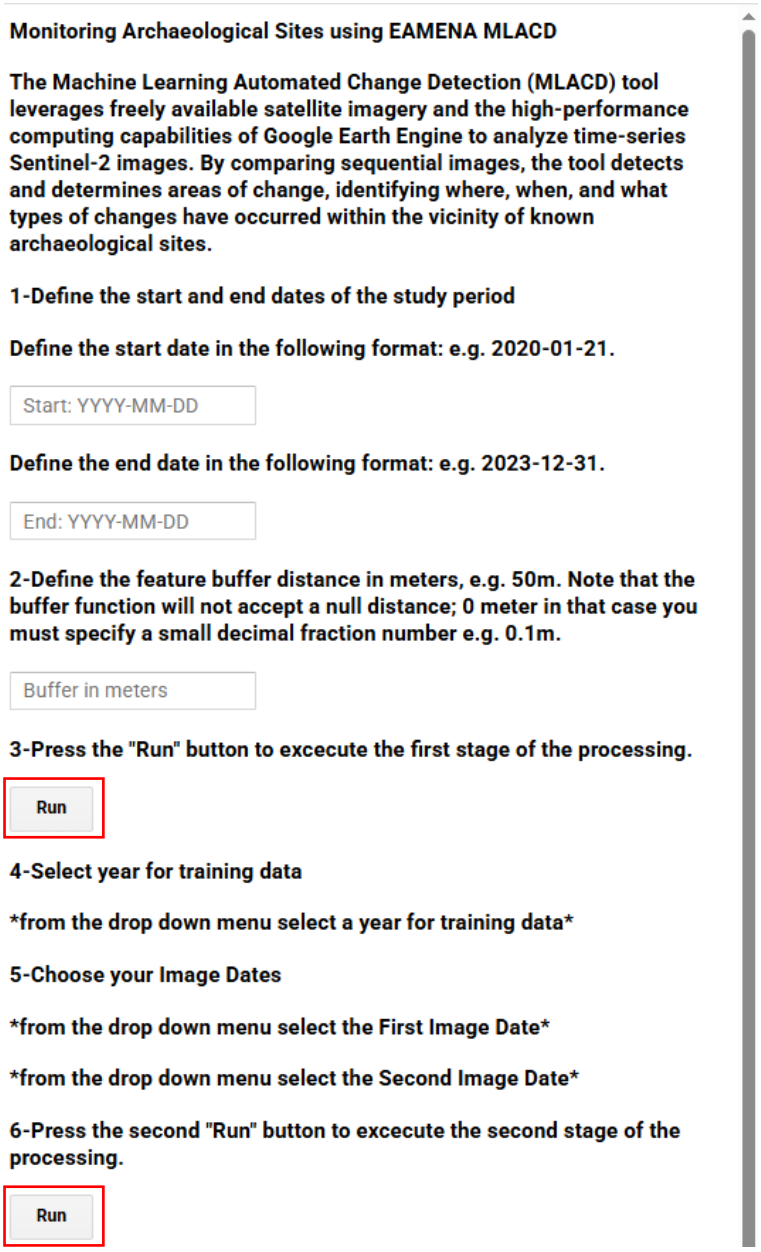


Figure 4. EAMENA MLACD User Interface

5.1.3 Select year for training data

Since the classifier trains on both the collected training samples and the image collection for a specific year, the user must select the year corresponding to when their training samples were collected.

Step 4:

- A dropdown menu of years [2017, 2018, 2019, 2020, 2021, …, 2023] will appear.
- Click **'Select a value'** and choose the relevant year (e.g., *2021*). This will automatically filter Sentinel-2 images from that year as inputs for training the Random Forest classifier.
- This step is critical to ensure that the training samples and the image collection belong to the same year, which improves classification accuracy and reduces errors.

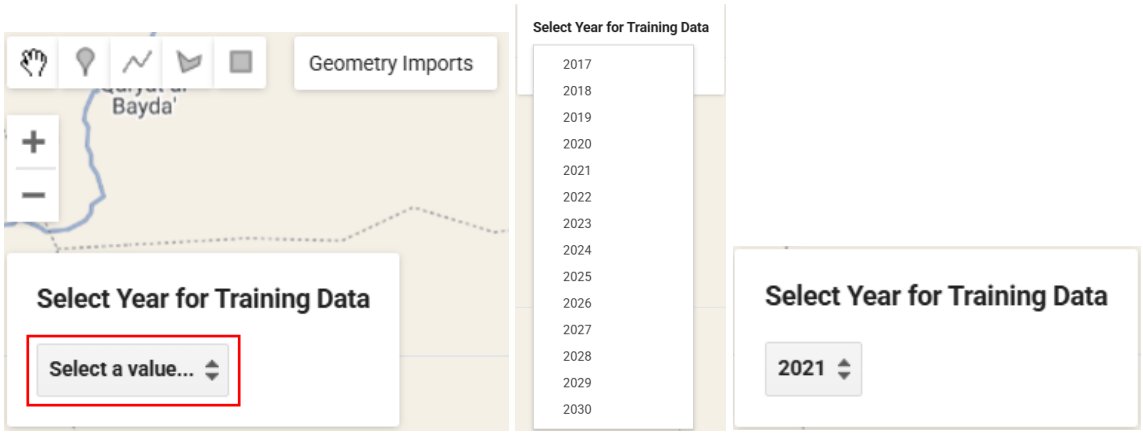


Figure 5. Select year of training data.

5.1.4 Select first and second images to be compared

After running the first stage, two dropdown menus will appear, showing the available Sentinel-2 image dates based on the start and end dates you specified in **Section 5.1.1**.

Step 5: To monitor changes between two specific dates of interest, **Select First Image** and **Select Second Image** from the dropdown menus.

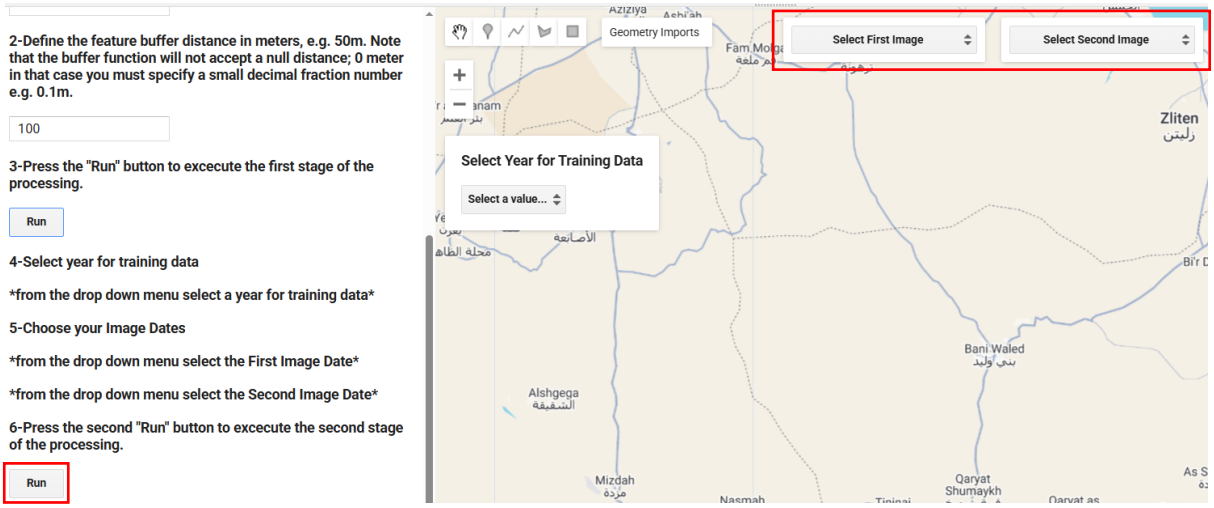


Figure 6. Select two images to compare for change detection

Once both dates are selected, the MLACD has all the necessary inputs to perform the change detection analysis.

Step 6: Click the second **'Run'** button on the MLACD user interface to execute the second stage of the processing (see Figure 6).

You can redefine your inputs by clicking on the '**Reset**' button.

6 Stage 2: Image classification and analysis

The MLACD produces several different outputs in the form of visual and statistical data. These includes:

- 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 that are automatically generated by the MLACD tool (Figure 7).

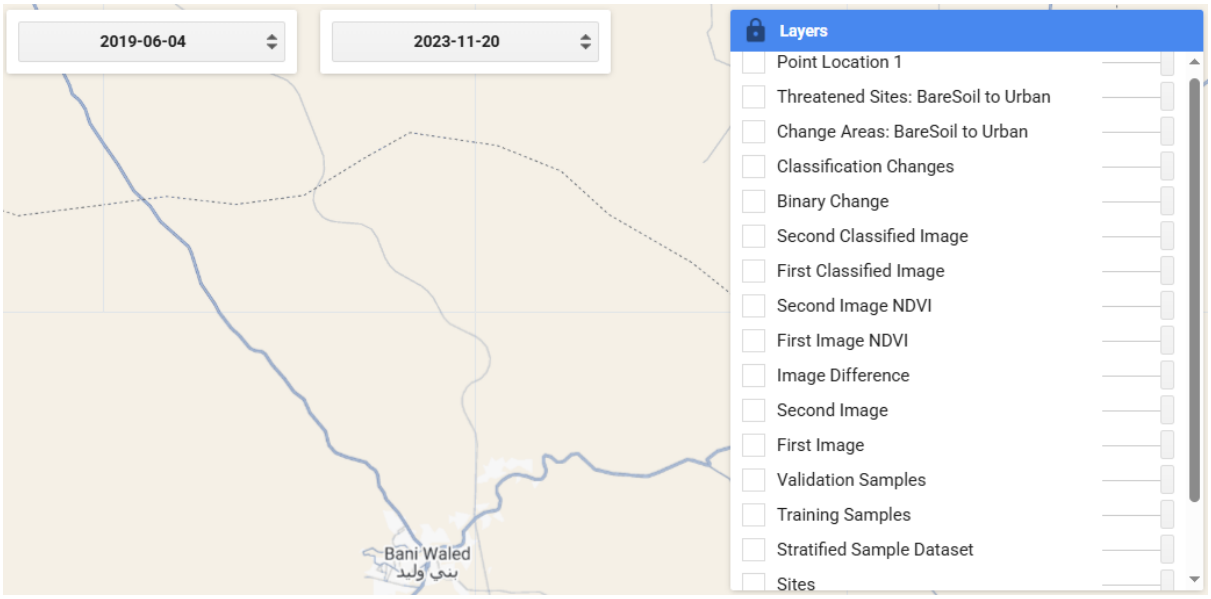


Figure 7. MLACD Output Layers on the map panel

This includes:

- the Sentinel-2 images for the two images you selected in Step 5 (Figure 8).
- classification maps for the two images you selected in Step 5 (Figure 9).
- a classification difference map between the two selected images (Figure 10).
- 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 11).
    - These Change Class Values are calculated by assigning a new value to each possible combination of classes, as illustrated in the table below, and are automatically calculated by the script.

Class in Image 1	Class in Image 2	Change Class Value	Change Class Label
C1	C1	1	BareSoil_to_ BareSoil
C1	C2	2	BareSoil_To_Mountain

C1	C3	3	BareSoil_To_Quarry
C1	C4	4	BareSoil_To_Urban
C1	C5	5	BareSoil_To_Vegetation
C2	C1	6	Mountain_To_BareSoil
C2	C2	7	Mountain_To_Mountain
C2	C3	8	Mountain_To_Quarry
…	…	…	…

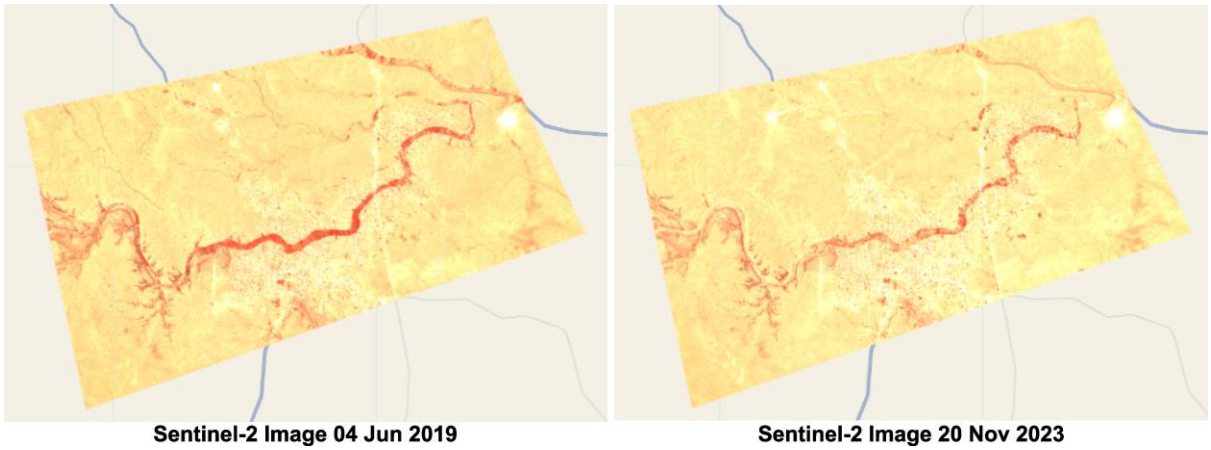


Figure 8. Sentinel-2 Images selected to detect the changes between them – First Image and Second Image in the Layers panel. For the visualization of each image the near infrared band (B8), Green band (B3) and Blue band (B2) were used, this explains the vegetated areas appearing as red.

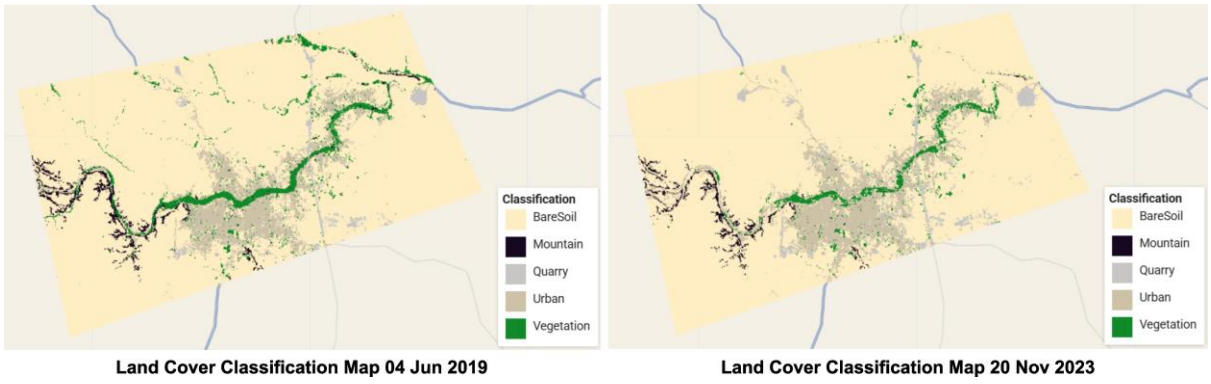


Figure 9. Land cover classification maps for two selected dates *(*First Classified Image, Second Classified Image).

- **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 misclassification.

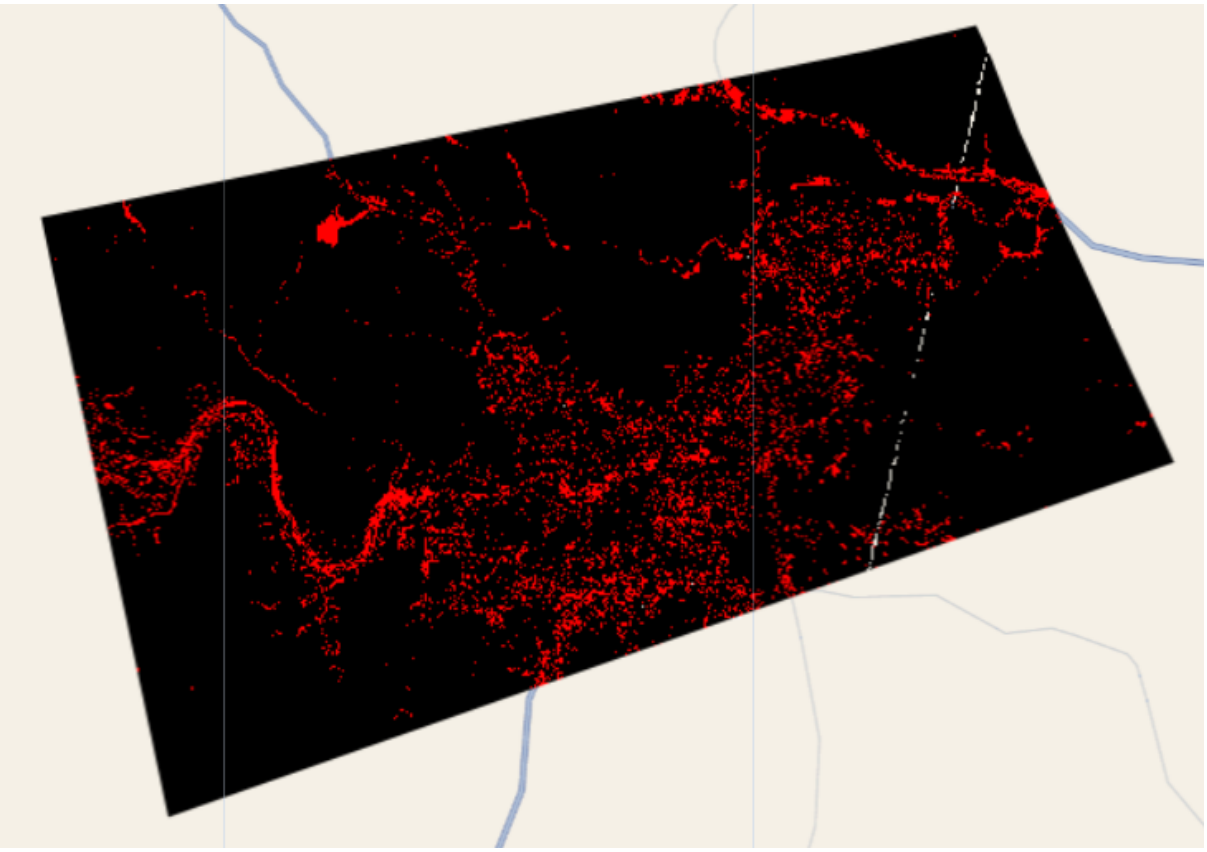


Figure 10. Binary change map between the selected First Image and Second Image. Black represents areas with no change and a binary value of (0), while red colour represents areas with a change with value of (1).

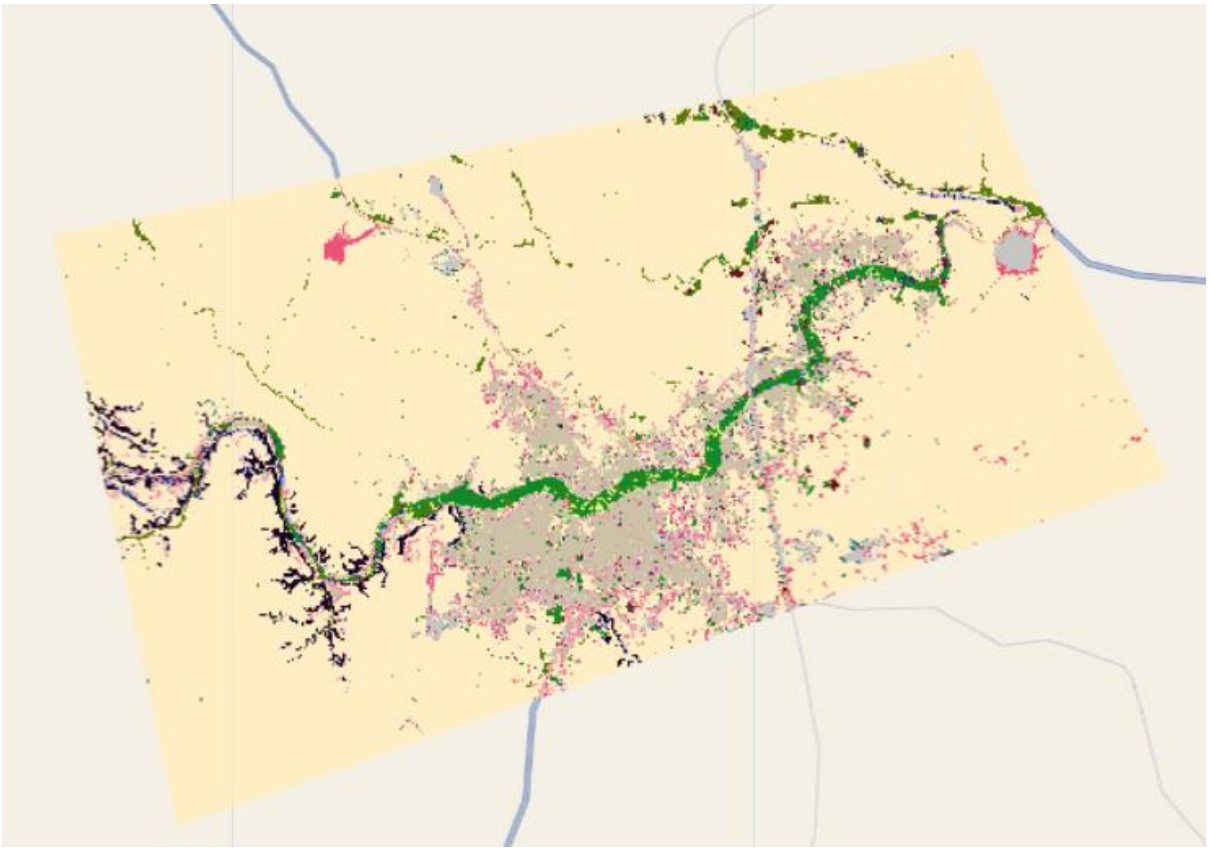


Figure 11. Classification Changes map demonstrating the changes in land cover features between the two selected images.

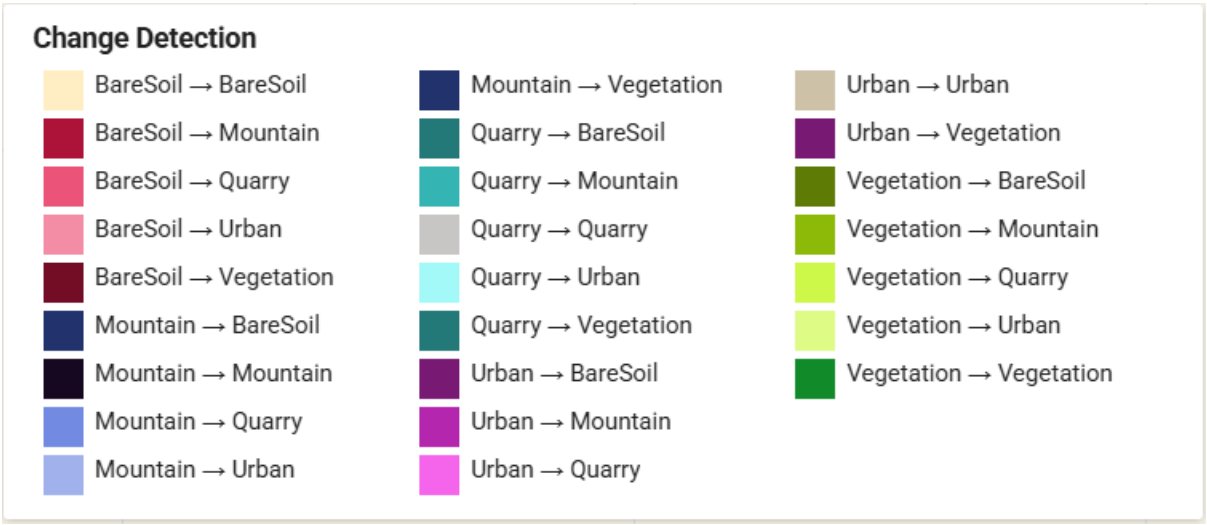


Figure 12. Legend for the change detection classification, showing the class transition labels and corresponding colours used to interpret the classification results. Each label represents a specific type of land cover change.

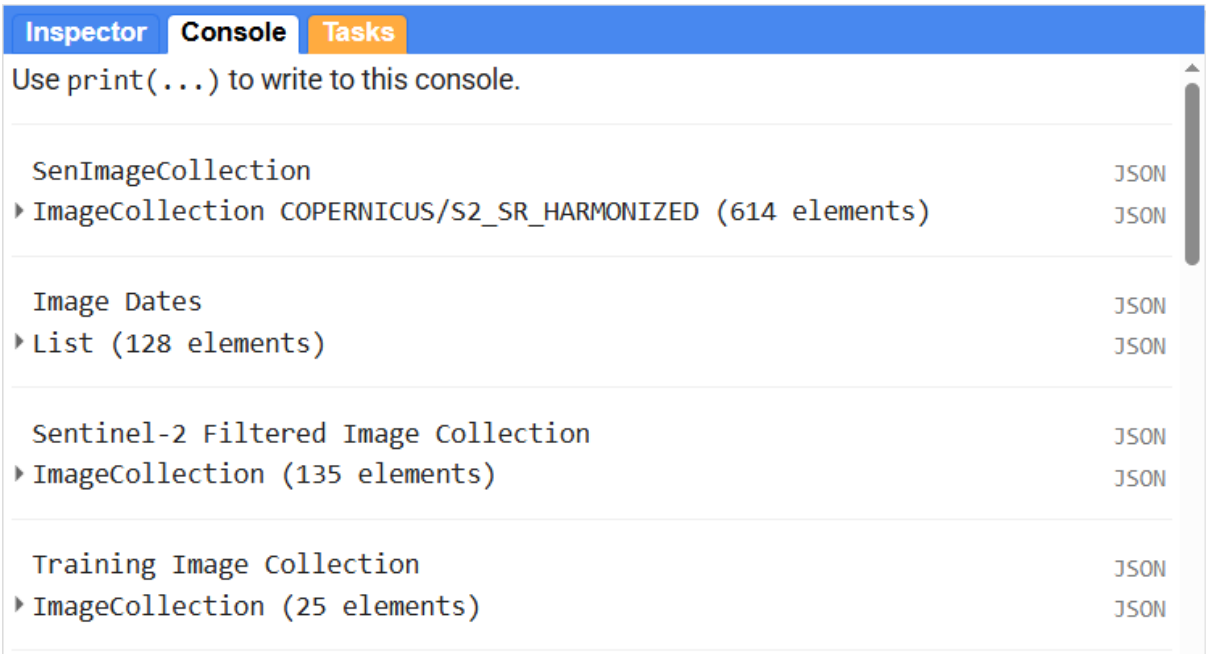
6.2 Results on the Console

When you have run the script, several other output results 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.

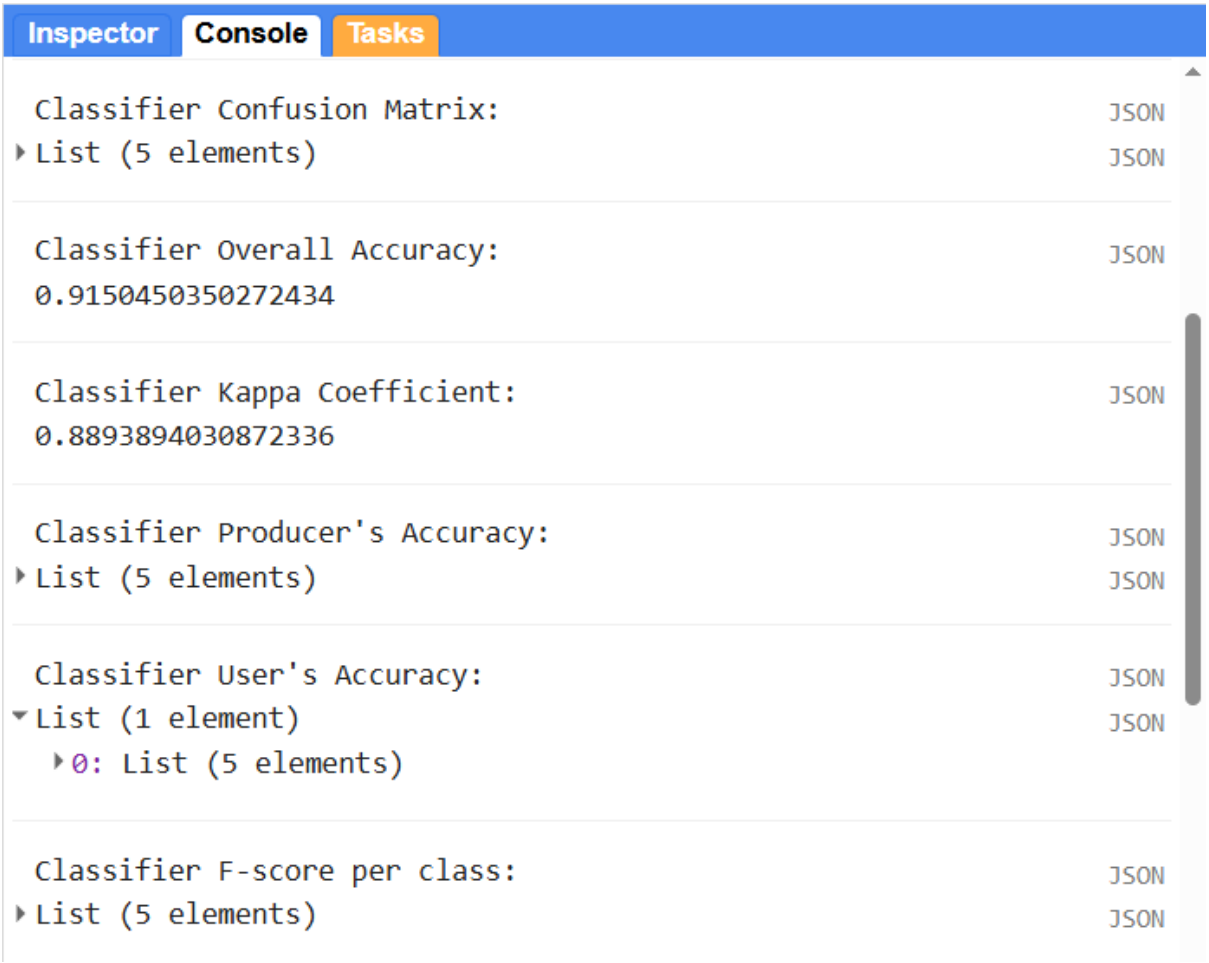
Image Dates and **Sentinel-2 Filtered Image Collection** provides information about the date of acquisition and properties of the images used in the analysis, which have been filtered for quality and mosaiced to ensure they cover the whole study area.

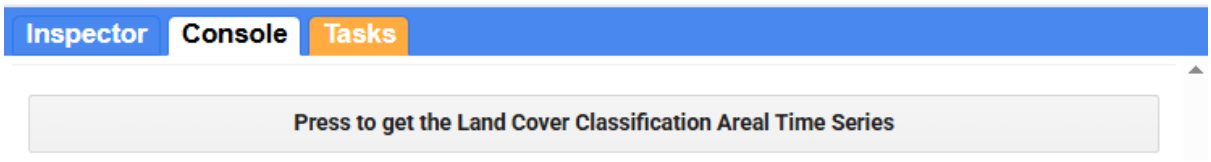
Training Image Collection which lists the images that are used in training the random forest classifier based on the year of training data selected by the user.



To assess and evaluate the performance of the supervised classification algorithm (i.e. Random Forest) several accuracy results are also generated on the console:

- 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.





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. Indeed, the accuracy of classification is very important here.

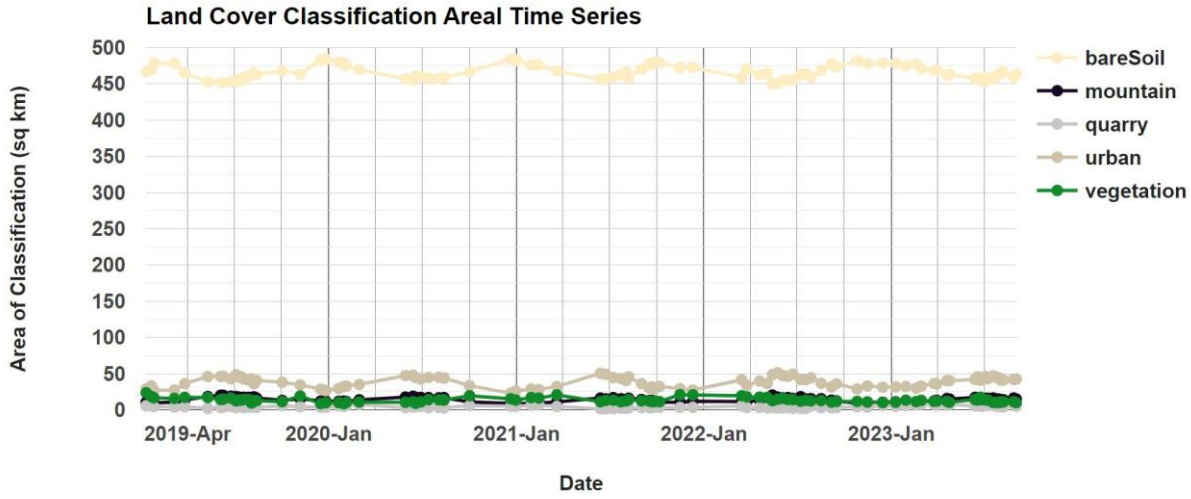
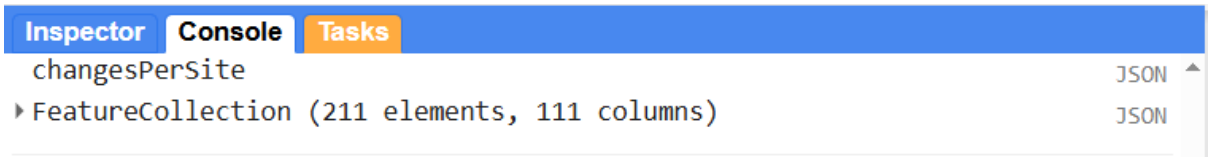


Figure 13. Land cover classification changes time series for Bani Walid

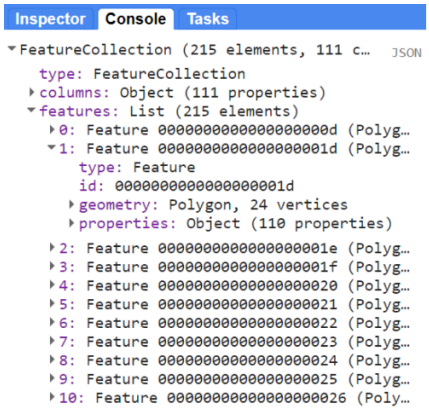
7 Stage 3: Identification of threats on archaeological sites

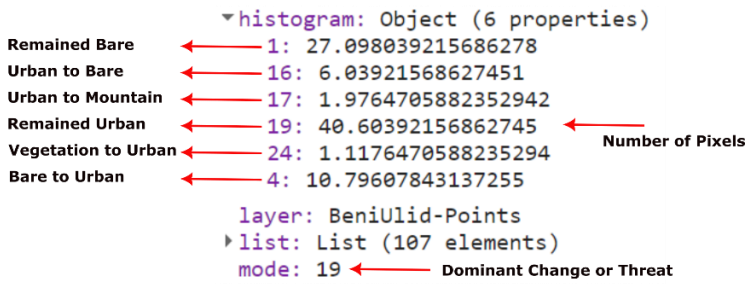
The next results in the console relate to statistical information about the pixel changes at each site. This change computation is the result of the classification change layer masked to each buffer site.



Under **changesPerSite: Feature Collection** 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”.





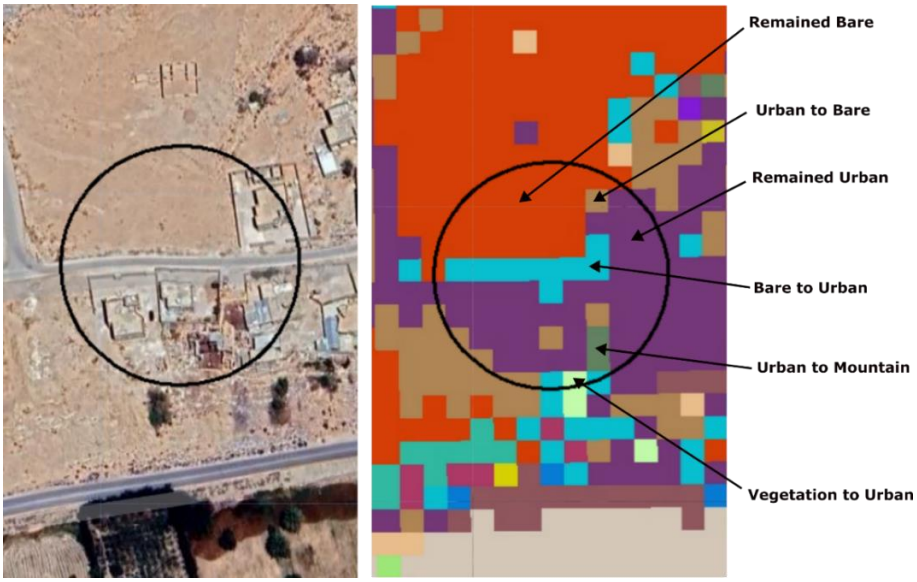


Figure 3. Classification change map in archaeological site EAMENA-0087287 in Bani Walid.

A histogram chart will also be generated on the Console to display the most frequent classification change found within each site buffer, and counts the number of sites by most common change in in land cover.

- This provides a general view of the most commonly detected threat in each of your investigated sites.

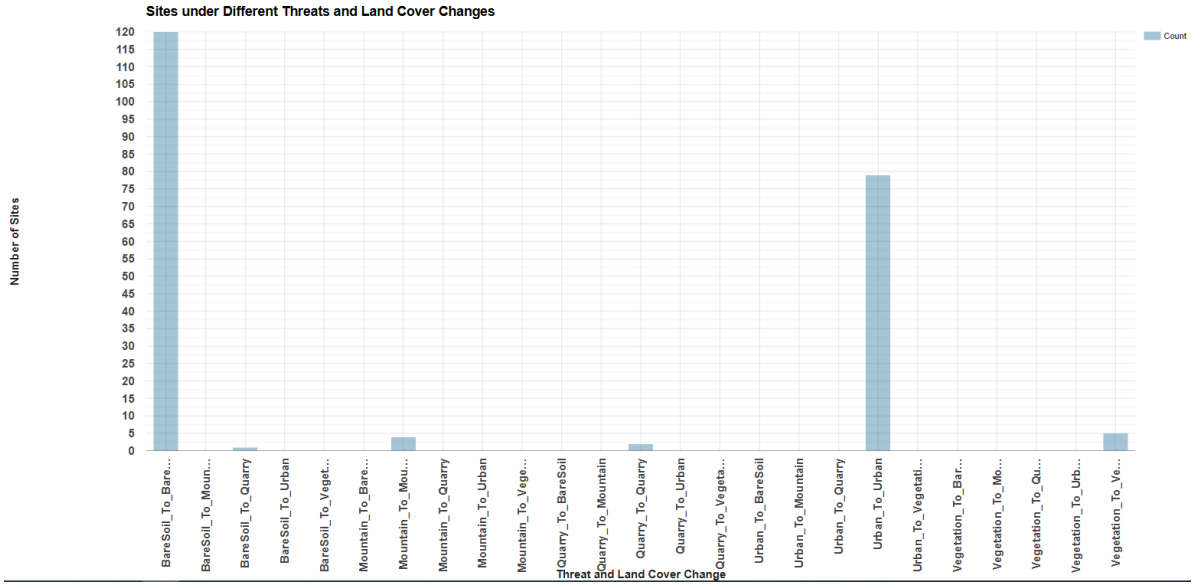
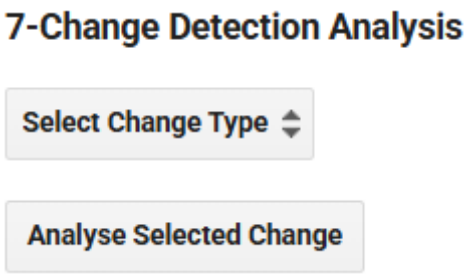


Figure 14. A histogram displaying the most frequent classification change found within each site buffer

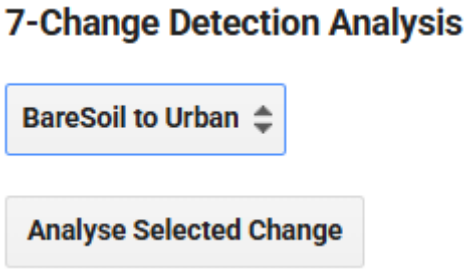
7.1 Change Detection Analysis:

Most importantly, the tool includes a dedicated feature for performing change detection analysis, which is activated by executing **Step 7: 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.

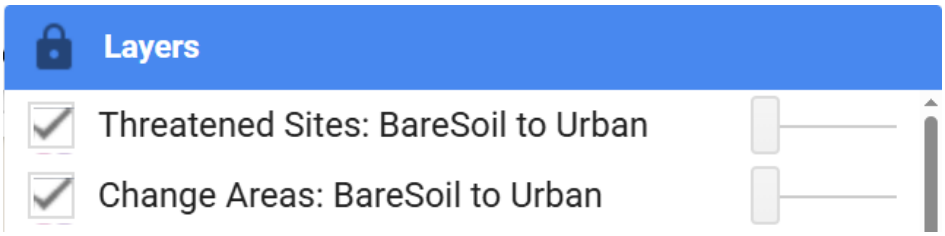
- Click on the '**Select Change Type'** icon and from the dropdown menu select the change type you want to investigate its threat; then click on the '**Analyse Selected Change'** button to execute the analysis.



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.



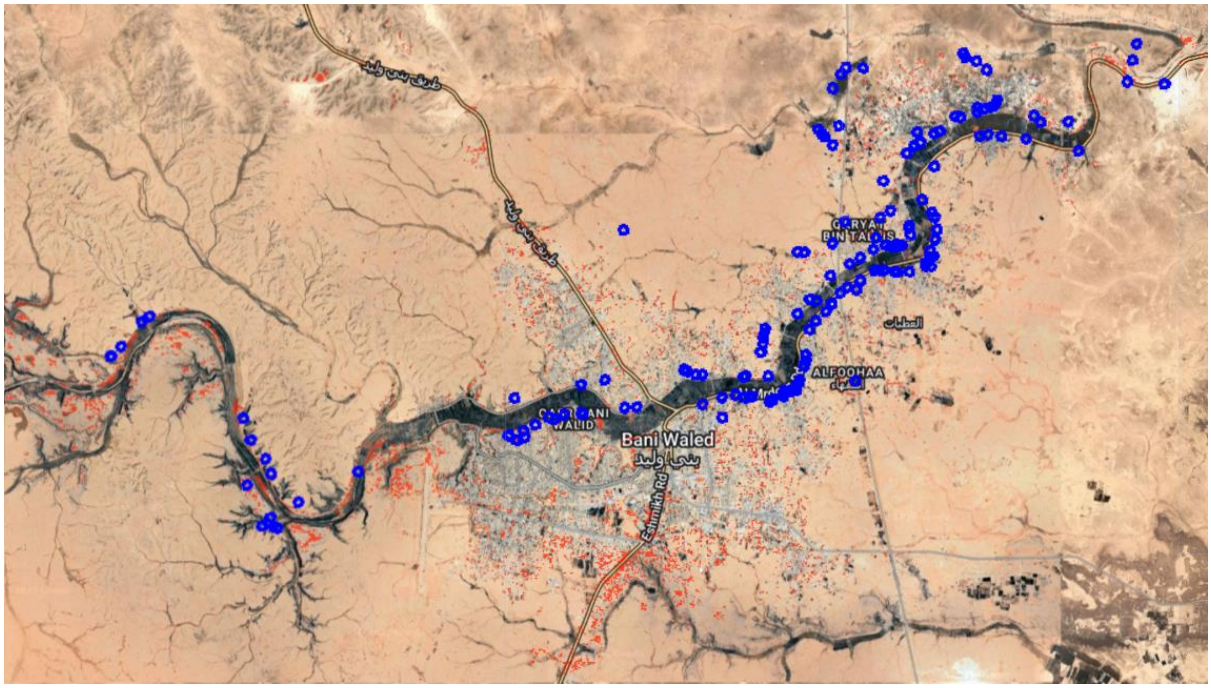
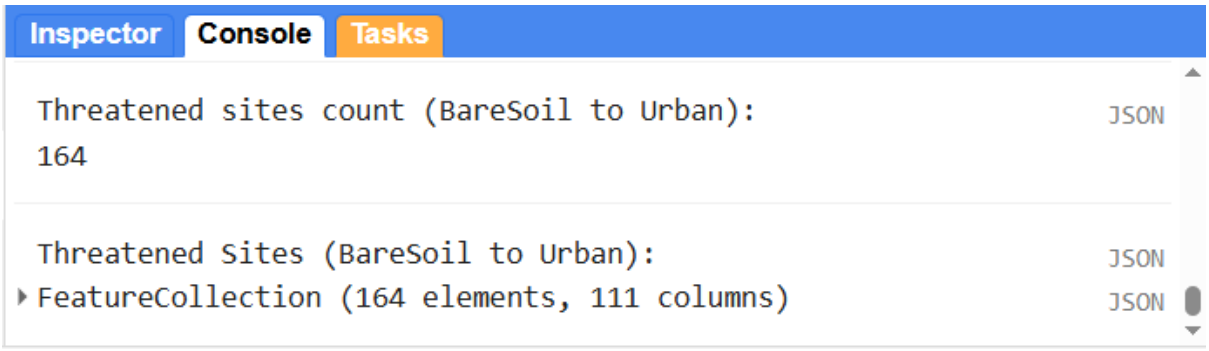
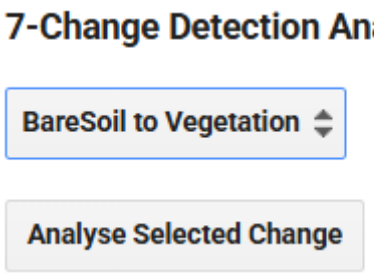


Figure 15. Classification change map showing areas that changed from BareSoil to Urban between the two selected dates and archaeological sites that intersects with these changed classification pixels.

The number of sites affected by this change will also be displayed in the Console panel. Click on **Threatened Sites (BareSoil to Urban)** Feature Collection to view a list of all sites where at least one pixel of the selected change has been detected within the site's buffer zone.



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 both the **Layers** panel and the **Console**, as before. For example, to detect land cover changes from ***Bare Soil*** **to** ***Vegetation*** and identify the sites impacted by this change, select **'BareSoil to Vegetation'** and view the corresponding results in the Console and Layers panel.



7.2 Classification time series analysis

The MLACD also enables users to **investigate specific locations of interest** to obtain more detailed results on their condition and land cover changes. These locations can include entire archaeological sites or particular areas within them that require a closer assessment of the changes and threats.

Step 8: The user can now hover over the study area and click on any location of interest to generate a classification time series chart for that point location. This chart, displayed in the Console, shows the classification changes or potential threats over time. Additionally, an NDVI chart for the same location is produced to illustrate vegetation changes and seasonal trends. At the same time, a new point layer is added to the Layers panel to mark the exact location selected for time series analysis.

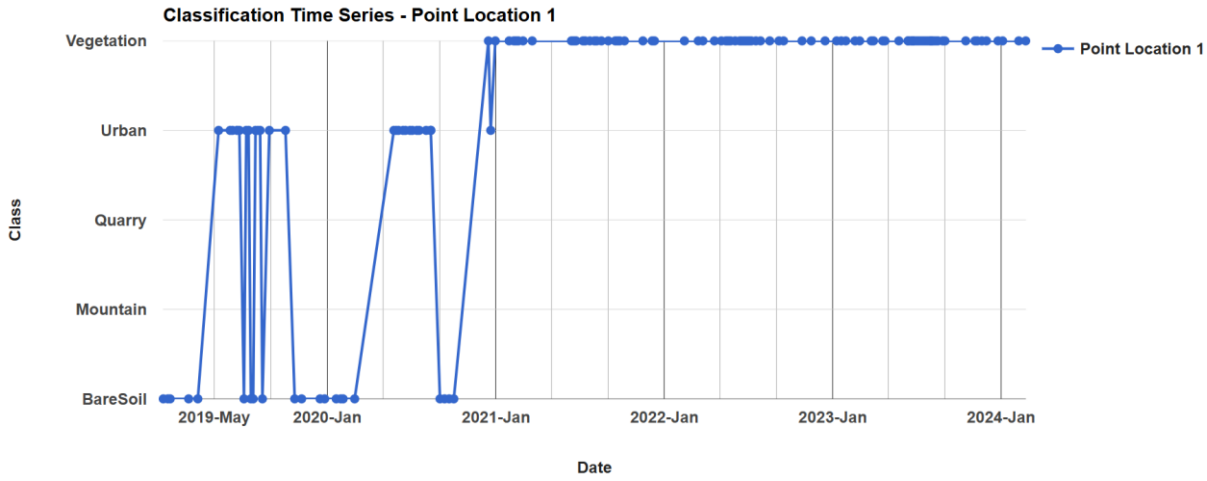
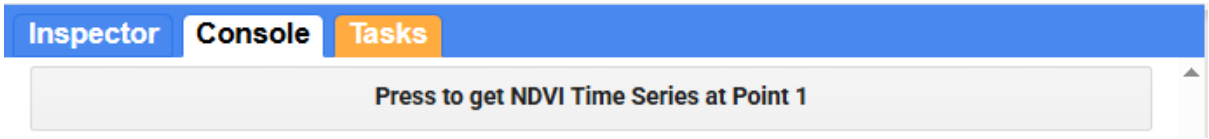


Figure 16. Classification time series of site (EAMENA-0189408) where (probably unintentional) vegetation expansion from a neighbouring farm was detected by the MLACD.

On the Console press on NDVI Time Series Chart button to get an NDVI time series chart for the location of interest.



- 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.

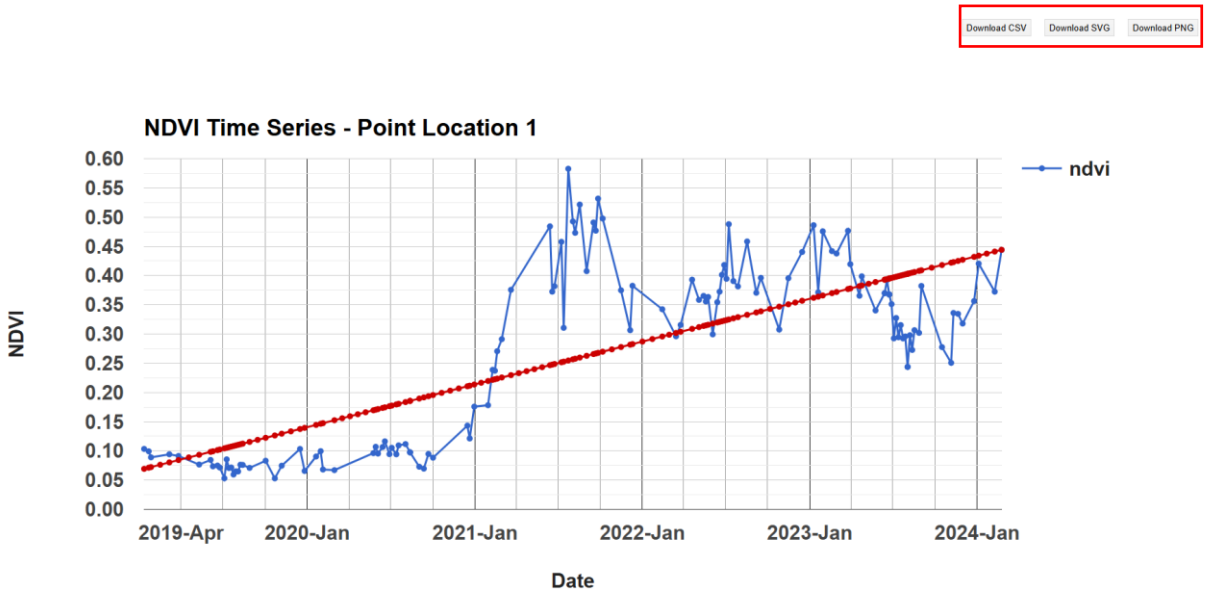


Figure 17. NDVI time series of site EAMENA-0189408 showing vegetation growth.

In the example above, we can see that sometime after Jan-2021, the classification at **Point Location 1,** 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 any location to inspect the layers' values in the chosen location.

8 Exports

From the **Tasks** tab in the **GEE Code Editor**, you can export all the features and raster datasets generated during processing including the study area boundary, training samples, validation samples, classification maps, change classification maps, change areas, and sites under threat.

- Once the export tasks appear in the **Task Manager**, click the **Run** button next to each feature or raster you wish to export.

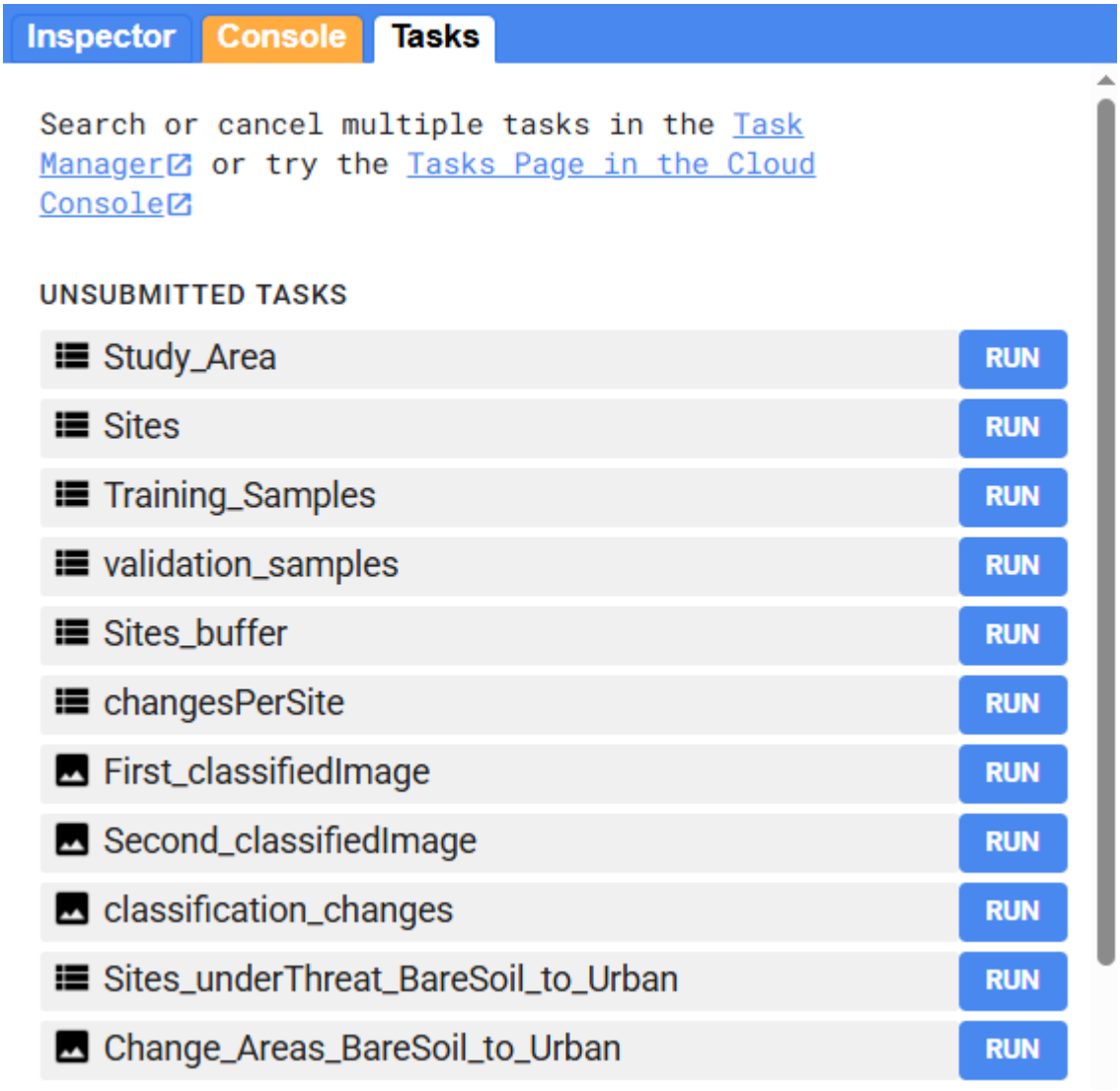
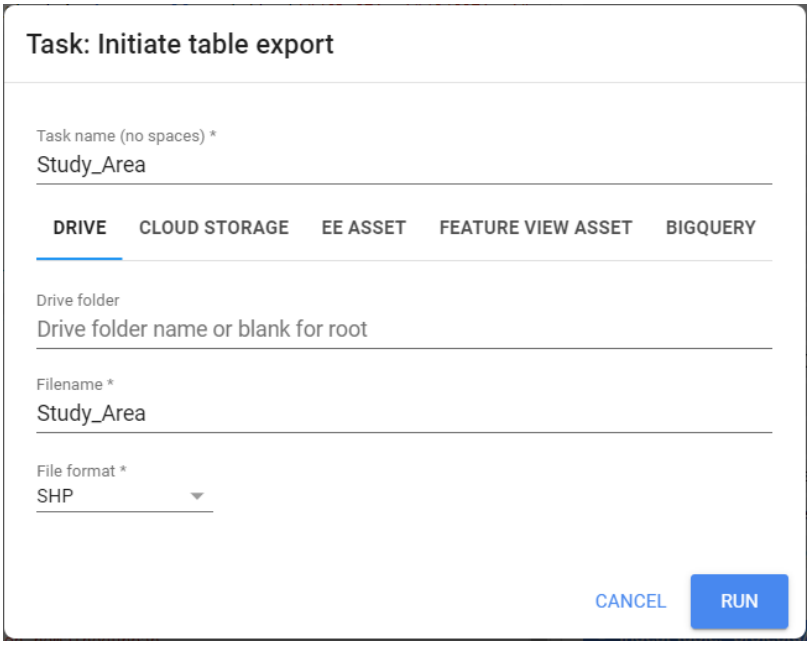


Figure 18. Export datasets from the code Editor Task Manager.

You can save the exported datasets to various locations (e.g., **Google Drive**, **Cloud Storage**, or **EE Assets**), depending on your workflow and available storage capacity.

- Specify the desired **export format** (if not already defined) and then press **Run** to start the export process.
- For more detailed instructions on exporting files from GEE, refer to the Google Earth Engine Exporting Guide.



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 three inputs that must be defined and imported so that the script can run properly which are the study area, archaeological sites and training sample datasets for all the land cover classes in your new case study area.
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 4.3.2** to define the new training samples.

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

10. Modify and edit the training samples and classification variables in the script based on the land cover classes in your new case study, by following the same instructions from **Section 4.3.1**.

```
// // Define the training samples datasets, colours and labels for each class in your case study
var trainingClasses = [
  {TrainingSample: BareSoil,   color: '#ffeec3', label: 'BareSoil'},   // class 1
  {TrainingSample: Mountain,   color: '#170821', label: 'Mountain'},   // class 2
  {TrainingSample: Quarry,     color: '#c7c6c5', label: 'Quarry'},     // class 3
  {TrainingSample: Urban,      color: '#cdc2a8', label: 'Urban'},      // class 4
  {TrainingSample: Vegetation, color: '#118b29', label: 'Vegetation'}, // class 5
];//Add and define additional training samples (e.g., TS: Water, color: '#2591ff', label: 'Water') or remove any class
```

- 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 several additional edits that advanced remote sensing users can apply to adapt the MLACD script to their specific area of interest. For example:

- **Editing the cloud percentage:** You can adjust the cloud coverage threshold to control the number of images accessed. A lower threshold reduces the number of images but ensures clearer scenes, while increasing the threshold allows more images to be included and may result in selecting images with higher cloud cover but it will potentially cause misclassification.

```
var SenImageCollection = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED")
                  .filterDate(startDateValue, endDateValue)
                  .filterBounds(Study_Area)
                  .filter(ee.Filter.lt('CLOUD_COVERAGE_ASSESSMENT',0.5))
                  .map(function(img){
                    return img.select(homogeneousBands);
                  });
```

- **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.

```
553  // Train the Random Forest classifier using the combined training samples
554  var trainedClassifier = ee.Classifier.smileRandomForest({
555    numberOfTrees: 100,
556    variablesPerSplit: null,
557    minLeafPopulation: 5,
558    bagFraction: 0.6,
559    maxNodes: null,
560    seed: 0
561  }).train({
562    features: monthlyTrainingSamples,
563    classProperty: 'landcover',
564    inputProperties: finalBands
565  });
```

- **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 class.

```
278  //*************************************************************************************************************
279  // Merge the training samples into one feature collection
280  var sampleDatasetFC = ee.FeatureCollection(trainingSets).flatten();
281  // print('sampleDatasetFC', sampleDatasetFC);
282  // Convert the training samples feature collection into an image
283  var sampleDatasetFCImage = ee.Image().byte().paint(sampleDatasetFC, 'landcover').rename('landcover');
284  // Collect 100 training samples for each class using the stratified sample function
285  var StratifiedSampleDataset = sampleDatasetFCImage.stratifiedSample({
286    numPoints: 1500,
287    classBand: 'landcover',
288    region: Study_Area,
289    scale: 10,
290    classValues: [0,1,2,3,4,5,6,7,8,9],
291    classPoints: [200, 200, 200, 200, 200, 200, 200, 200, 200, 200],
292    dropNulls: true,
293    geometries: true
294  });
295  // print (StratifiedSampleDataset, 'StratifiedSampleDataset');
296  Map.addLayer(StratifiedSampleDataset, {}, 'Stratified Sample Dataset',false);
297  // Training/Validation Split
298  var StratifiedSampleDatasetRandom = StratifiedSampleDataset.randomColumn();
299  var training_samples = StratifiedSampleDatasetRandom.filter(ee.Filter.lt('random', 0.7));
300  var validation_samples = StratifiedSampleDatasetRandom.filter(ee.Filter.gte('random', 0.7));
```