Connecting Google Colab and Earth Engine for Satellite Imagery Analysis and Vegetation Monitoring

Dr Ana María Pacheco Pascagaza
University of Manchester
Towards the UK fire danger rating system (UKFDRS)

The project aims to establish and validate the scientific basis and essential components necessary for the development of a bespoke UK Fire Danger Rating System (FDRS).

This system will facilitate the evaluation of current and future fire regimes, enabling the assessment, management, and mitigation of wildfire impacts and the associated risk of uncontrollable wildfires throughout the UK.
Agenda

- Introduction
- Google Earth Engine (GEE) – What it is? Benefits?
- Google Colab – What it is? Benefits?
- Connecting GEE API in Google Colab
- Importing and visualising datasets
- Data analysis applications
- Practical overview of the Colab notebook
- Limitations
- Questions
The Earth is undergoing rapid changes, and environmental degradation is intensifying, significantly impacting vegetation patterns.

Environmental monitoring, forest monitoring, land cover assessment, climate studies, fires and disaster management rely heavily on access to up-to-date satellite data.

The ability to extract meaningful insights from these vast datasets is crucial for informed decision-making that help addressing environmental challenges.

There is an urgent need for timely and efficient analysis of satellite imagery to monitor and understand these transformations.
Google Earth Engine

- A cloud-based geospatial platform developed by Google

- Allows their users to analyse and process large-scale Earth observation and geospatial data

Represents a valuable tool for researchers, scientists, environmentalists, and developers interested in connecting the power of geospatial data and analysis to address various Earth-related challenges and questions.
GEE Key Features and benefits

- **Data Archive**
  - Provides access to a massive archive of Earth observation and derived products.
  - This data spans multiple decades and covers the entire globe in most of the cases.

- **Code Editor**
  - You can write and run JavaScript or Python code within the Earth Engine Code Editor.

- **Geospatial Analysis**
  - Offers a wide range of geospatial analysis tools and functions on large datasets.
  - Examples: image processing, spatial analysis, time-series analysis, and machine learning.

- **Visualization**
  - You can visualise geospatial data and analyse results interactively on the map within the Code Editor, making it easier to explore and understand complex spatial patterns.

- **Collaboration**
  - Supports collaboration by enabling users to share their scripts, visualisations, and analysis results.
  - Collaborators can work on projects simultaneously and access shared data and code.
Google Colab

- It is a free cloud-based platform provided by Google
- It allows you to write and execute Python code in a web-based interactive environment
- It is presented in a Jupyter notebook format
Google Colab Key Features

**Free access**
- Provides access to access to Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) at no cost.

**Google drive**
- It is integrated with Google Drive, making easy to store and share notebooks

**Libraries**
- It comes with a wide range of pre-installed Python libraries
- Example: pandas, scikit-learn, TensorFlow, PyTorch etc.

**Notebooks**
- Notebooks are interactive documents that combine code execution cells with formatted text and visualisations.
- You can write and run Python code, add explanations, charts, and images, creating a narrative around their data analysis or machine learning projects.

**Collaboration**
- Supports collaboration by enabling users to share their scripts, visualisations, and analysis results with others.
- Collaborators can work on projects simultaneously and access shared data and code.
- You can integrate Colab with version control (e.g. GitHub) to track changes, and edit.
Let’s explore a Colab Jupyter notebook and how you can connect it with Google Earth Engine
Setting up

**STEP 1:** Install your libraries

```python
# library to link Google Earth Engine
!pip install earthengine-api

# Other libraries that you might need for further analysis
!pip install geemap
!pip install matplotlib-venn
```

**STEP 2:** Import your libraries and link to your google drive

```python
import ee
import geemap
from google.colab import drive

print("Setup completed")
drive.mount("/content/drive")
```
1. You need to access your google account

   ee.Authenticate()
e.e.Initialize()
To authorize access needed by Earth Engine, open the following URL in a web browser and follow the instructions. If the web browser does not start automatically, please manually browse the URL below.


The authorization workflow will generate a code, which you should paste in the box below.

Enter verification code: 

---

Earth Engine Notebook Client - ukfdr@gmail.com wants access to your Google Account

---

Select what Earth Engine Notebook Client - ukfdr@gmail.com can access

- View and manage your Google Earth Engine data. Learn more.
- Manage your data and permissions in Cloud Storage and see the email address for your Google Account. Learn more.

Make sure that you trust Earth Engine Notebook Client - ukfdr@gmail.com

---

Authorization code

Please copy this code, switch to your application and paste it here:

4jL4o0FL PuZBxx2d 4jL4o0FL PuZBxx2d 4jL4o0FL PuZBxx2d CECclXs...

Don't sign in or provide access to Earth Engine Notebook Client - ukfdr@gmail.com if you don't want to continue, close this window.
## Datasets available in GEE achive

<table>
<thead>
<tr>
<th>Category</th>
<th>Datasets</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical Satellite Data</td>
<td>Landsat 8 Surface Reflectance, Sentinel-2 Surface Reflectance, MODIS Land Surface Temperature</td>
<td>Satellite imagery datasets providing reflectance, temperature, and other Earth observation data.</td>
</tr>
<tr>
<td>Radar Satellite Data</td>
<td>ALOS PALSAR, Sentinel-1</td>
<td>Radar satellite datasets offering various applications including terrain mapping and monitoring.</td>
</tr>
<tr>
<td>Topography</td>
<td>SRTM (Shuttle Radar Topography Mission)</td>
<td>Elevation and topography data derived from radar measurements.</td>
</tr>
<tr>
<td>Land Cover Data</td>
<td>Copernicus Global Land Cover Layers, MODIS Land Cover Type</td>
<td>Global land cover classification datasets at various resolutions and time periods.</td>
</tr>
<tr>
<td>Climate Datasets</td>
<td>ERA5 Climate Reanalysis, CHIRPS Precipitation Data</td>
<td>Climate datasets offering temperature, precipitation, and other meteorological variables.</td>
</tr>
<tr>
<td>Derived Maps</td>
<td>Global Forest Change, NASA Global Fire Maps, GEDI Biomass</td>
<td>Maps derived from satellite data, including forest change, fire occurrence, and biomass.</td>
</tr>
</tbody>
</table>

See more at: [Earth Engine Data Catalog](https://earthengine.google.com/search) | [Google for Developers](https://developers.google.com/earth-engine)
Define parameters you want to consider

- Scale
- Area
- Time period

```python
# Spatial resolution
scale = 30

# Define the coordinates of the polygon
coordinates = [[-8.709412085888312, 60.8875076094291],
               [-8.709412085888312, 49.5],
               [1.8209834219241872, 49.5],
               [1.8209834219241872, 60.8875076094291]]

# Study area
polygon = ee.Geometry.Polygon(coordinates)

# Time period
start_date = "2017-01-01"
end_date = "2022-12-31"
```
Define functions that you want to apply to your imagery

- Cloud masking
- Vegetation indices
Define functions that you want to apply to your imagery.
Visualisation of Satellite data
Visualisation of other available datasets and plotting histograms

SRTM
<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
<th>Google Earth Engine Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal Analysis</td>
<td>Analysing changes over time in satellite data to detect trends and patterns.</td>
<td>ee.Reducer.mean(), ee.Reducer.median(), ee.Reducer.stdDev(), ee.ImageCollection.filterDate()</td>
</tr>
<tr>
<td>Climate Studies</td>
<td>Studying climate change, weather patterns, and climate variables like temperature and precipitation.</td>
<td>ee.ImageCollection.filter(), ee.ImageCollection.mean(), ee.ImageCollection.reduce()</td>
</tr>
<tr>
<td>Disaster Management</td>
<td>Responding to natural disasters by analysing pre- and post-disaster imagery to identify affected areas.</td>
<td>ee.ImageCollection.filterBounds(), ee.ImageCollection.mosaic(), ee.Image.reduceRegion()</td>
</tr>
<tr>
<td>Topography Analysis</td>
<td>Deriving elevation, slope, aspect, and other topographic information from elevation datasets.</td>
<td>ee.Image.gradient(), ee.Image.hillshade(), ee.Terrain.slope(), ee.Terrain.aspect()</td>
</tr>
<tr>
<td>Derived Maps</td>
<td>Generating thematic maps such as biomass, vegetation height, forest change, and fire risk.</td>
<td>Custom algorithms using ee.Image.expression(), ee.Image.updateMask(), and ee.Reducer functions.</td>
</tr>
</tbody>
</table>
## Limitations

<table>
<thead>
<tr>
<th></th>
<th>Google Earth Engine</th>
<th>Google Colab</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ease of Use</strong></td>
<td>Requires proficiency in Earth Engine's JavaScript-based code.</td>
<td>Familiar Python interface for users.</td>
</tr>
<tr>
<td><strong>Learning Curve</strong></td>
<td>Steeper learning curve for newcomers.</td>
<td>More accessible for Python developers.</td>
</tr>
<tr>
<td><strong>Access to Data</strong></td>
<td>Extensive satellite and geospatial datasets available.</td>
<td>Limited to publicly available datasets.</td>
</tr>
<tr>
<td><strong>Customisation</strong></td>
<td>Highly customizable, but requires coding skills.</td>
<td>Customization through Python libraries.</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Excellent for handling large geospatial datasets.</td>
<td>Limited by Colab's CPU/GPU resources.</td>
</tr>
<tr>
<td><strong>Interactive Analysis</strong></td>
<td>Offers an interactive map-based interface.</td>
<td>Limited interactivity, primarily code.</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>Data is hosted on Google's servers, storage is not an issue.</td>
<td>Limited storage, temporary session data.</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Free tier available, but costs can increase with extensive use.</td>
<td>Limited free GPU hours, additional costs for extended use.</td>
</tr>
<tr>
<td><strong>Offline Use</strong></td>
<td>Internet connection required to access datasets.</td>
<td>Internet connection required for usage.</td>
</tr>
</tbody>
</table>
Let’s have a practical showcase of a Colab notebook to explore optical satellite data and derive a time series of the normalised vegetation index NDVI.

[FOSS4G_UK.ipynb - Colaboratory (google.com)]
Do you want to learn more?

- Explore the Geemap Github where you can find, videos, and tutorials that will helpful [Tutorials – geemap](#).

- Explore more about Google Colab at [Welcome to Colaboratory - Colaboratory (google.com)](#).
Any questions?

Thank you

Email: ana.pachecopascagaza@manchester.ac.uk
Website: https://ukfdrs.com/
Twitter: @ukfdrs
GitHub: https://github.com/UKFDRS