FLAGSHIP 2.6.3: AI ALGORITHM FOR (SATELLITE) IMAGING RECONSTRUCTION

REPORT FOR WP6 MEETING, 23/01/2024

A. Tricomi[1,2,3], G. Piparo[1], G. A. Anastasi[2], E. Tramontana[2], V. Strati[4]

1. INFN Sezione di Catania
2. Università degli studi di Catania
3. Centro Siciliano di Fisica Nucleare e Struttura della Materia (CSFNSM)
4. Università degli studi di Ferrara

ICSC
Centro Nazionale di Ricerca in HPC,
Big Data and Quantum Computing

General Status of the Flagship

- ➢ The Spoke 2 general meeting was a good opportunity to get **to know each other and discuss this year's work** and also to show the current advances of the flagship! First discussion points also opened with related IG responsible persons.

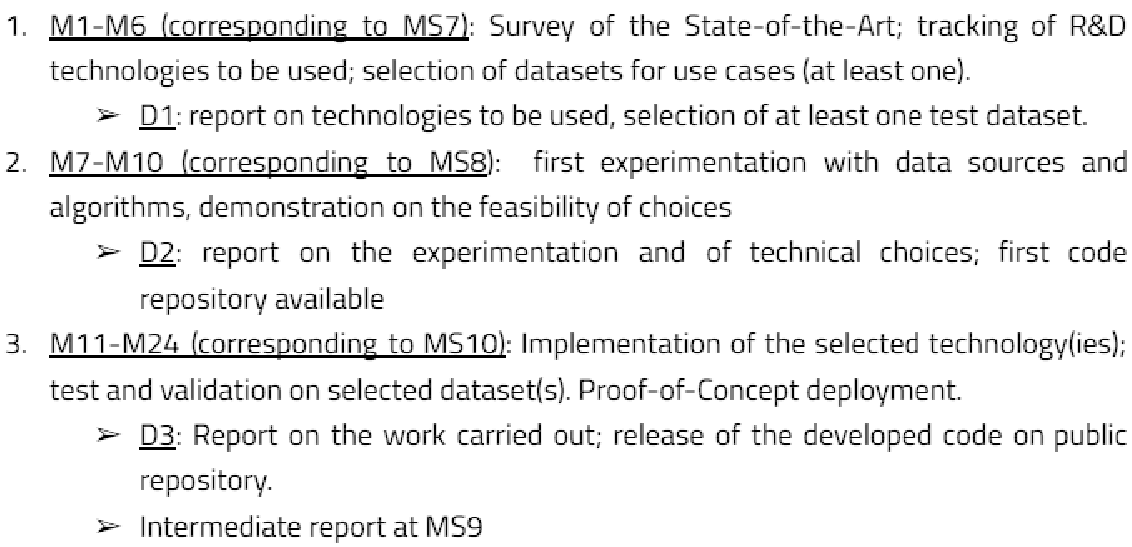
- ➢ The general state is **quite good** despite the somewhat slower start compared to other projects.

- ➢ The flagship work can be divided into three interconnected branches:
  - ➢ 1) Identification of objects using satellite images (**Emiliano and his group**).
  - ➢ 2) Study of vineyard diseases using aerial images (**Virginia and her group**).
  - ➢ 3) General study of techniques for analysing satellite images (**Alessia, myself and Gialex**). Currently, attempts to replicate/integrate the results of group 2) using satellite images.

- ➢ **We are confident that the coordinated work of these three groups can complete the objectives of the flagship.**

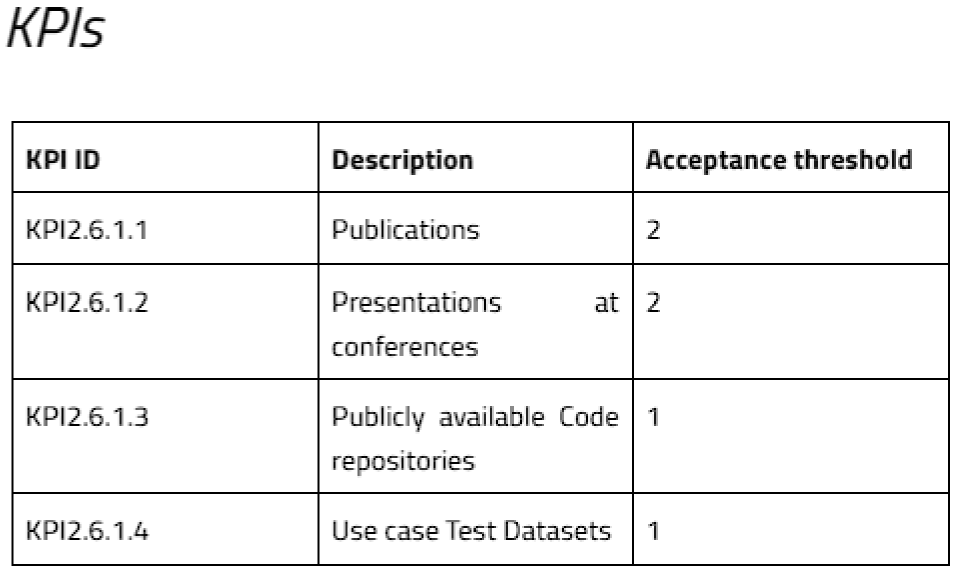
MILESTONES AND KPI OF FLAGSHIP 2.6.3

Milestones



1. M1-M6 (corresponding to MS7): Survey of the State-of-the-Art; tracking of R&D technologies to be used; selection of datasets for use cases (at least one).
   - ➢ D1: report on technologies to be used, selection of at least one test dataset.
2. M7-M10 (corresponding to MS8): first experimentation with data sources and algorithms, demonstration on the feasibility of choices
   - ➢ D2: report on the experimentation and of technical choices; first code repository available
3. M11-M24 (corresponding to MS10): Implementation of the selected technology(ies); test and validation on selected dataset(s). Proof-of-Concept deployment.
   - ➢ D3: Report on the work carried out; release of the developed code on public repository.
   - ➢ Intermediate report at MS9

KPIs



KPI ID	Description	Acceptance threshold
KPI2.6.1.1	Publications	2
KPI2.6.1.2	Presentations at conferences	2
KPI2.6.1.3	Publicly available Code repositories	1
KPI2.6.1.4	Use case Test Datasets	1

Current status

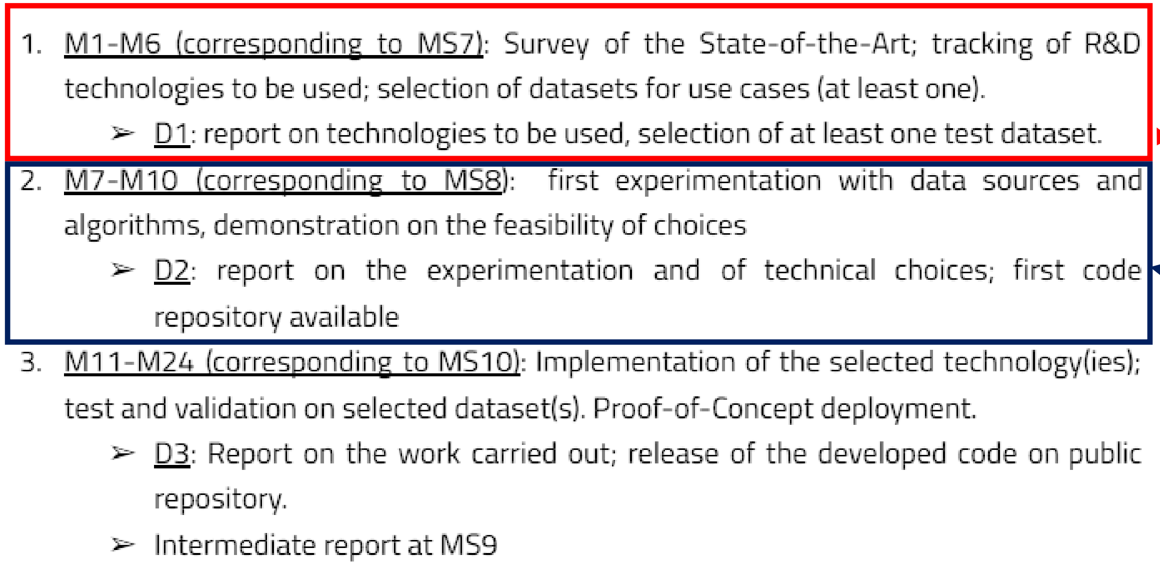
1*

*D. Marletta, A. Midolo, E. Tramontana. Detecting Photovoltaic Panels in Aerial Images by Means of Characterising Colours. Technologies 2023, 11(6), 174; https://doi.org/10.3390/technologies11060174

ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

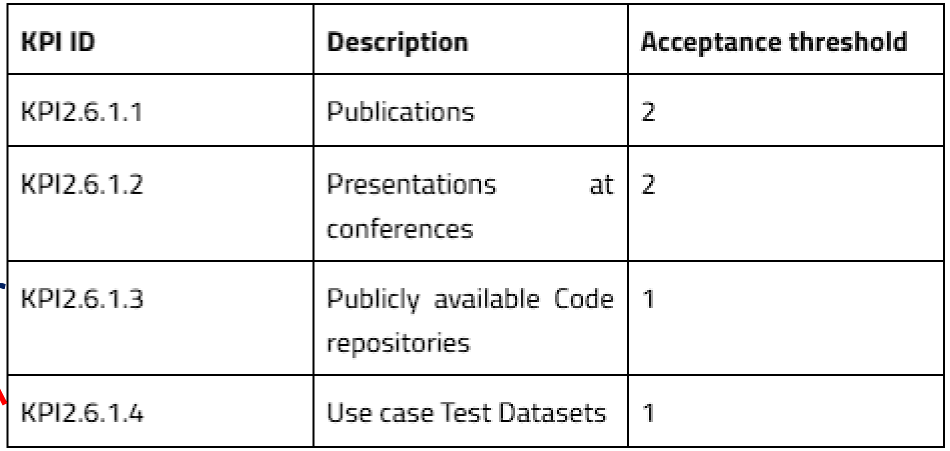
Milestones and KPI of Flagship 2.6.3

Milestones



1. M1-M6 (corresponding to MS7): Survey of the State-of-the-Art; tracking of R&D technologies to be used; selection of datasets for use cases (at least one).
   - ➢ D1: report on technologies to be used, selection of at least one test dataset.
2. M7-M10 (corresponding to MS8): first experimentation with data sources and algorithms, demonstration on the feasibility of choices
   - ➢ D2: report on the experimentation and of technical choices; first code repository available
3. M11-M24 (corresponding to MS10): Implementation of the selected technology(ies); test and validation on selected dataset(s). Proof-of-Concept deployment.
   - ➢ D3: Report on the work carried out; release of the developed code on public repository.
   - ➢ Intermediate report at MS9

KPIs



KPI ID	Description	Acceptance threshold
KPI2.6.1.1	Publications	2
KPI2.6.1.2	Presentations at conferences	2
KPI2.6.1.3	Publicly available Code repositories	1
KPI2.6.1.4	Use case Test Datasets	1

Current status

1*

*D. Marletta, A. Midolo, E. Tramontana. Detecting Photovoltaic Panels in Aerial Images by Means of Characterising Colours. Technologies 2023, 11(6), 174; https://doi.org/10.3390/technologies11060174

ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

AgriSentinel: A Comprehensive Library for Downloading, Manipulating and Analysing Satellite Data

- In some of the discussions at the Spoke 2 meeting, **the difficulty of managing the downloading and manipulation of satellite data** for the realization of datasets using the available frameworks emerged.
- **We decided to develop a library that would serve as a unified framework for the preparation of datasets used for training AI techniques, and possibly integrate these techniques within it.**
- Development is already underway and the first version of the code is already available on github (currently in private form): https://github.com/gpiparo2/AgriSentinel.
- **In addition to being useful internally, the development of this library could be a valuable support for the completion of certain KPIs.**

A (BRIEF) EXPLANATION OF HOW AGRISENTINEL WORKS

- Agrisentinel is a modular library in python, currently consisting of 4 modules:

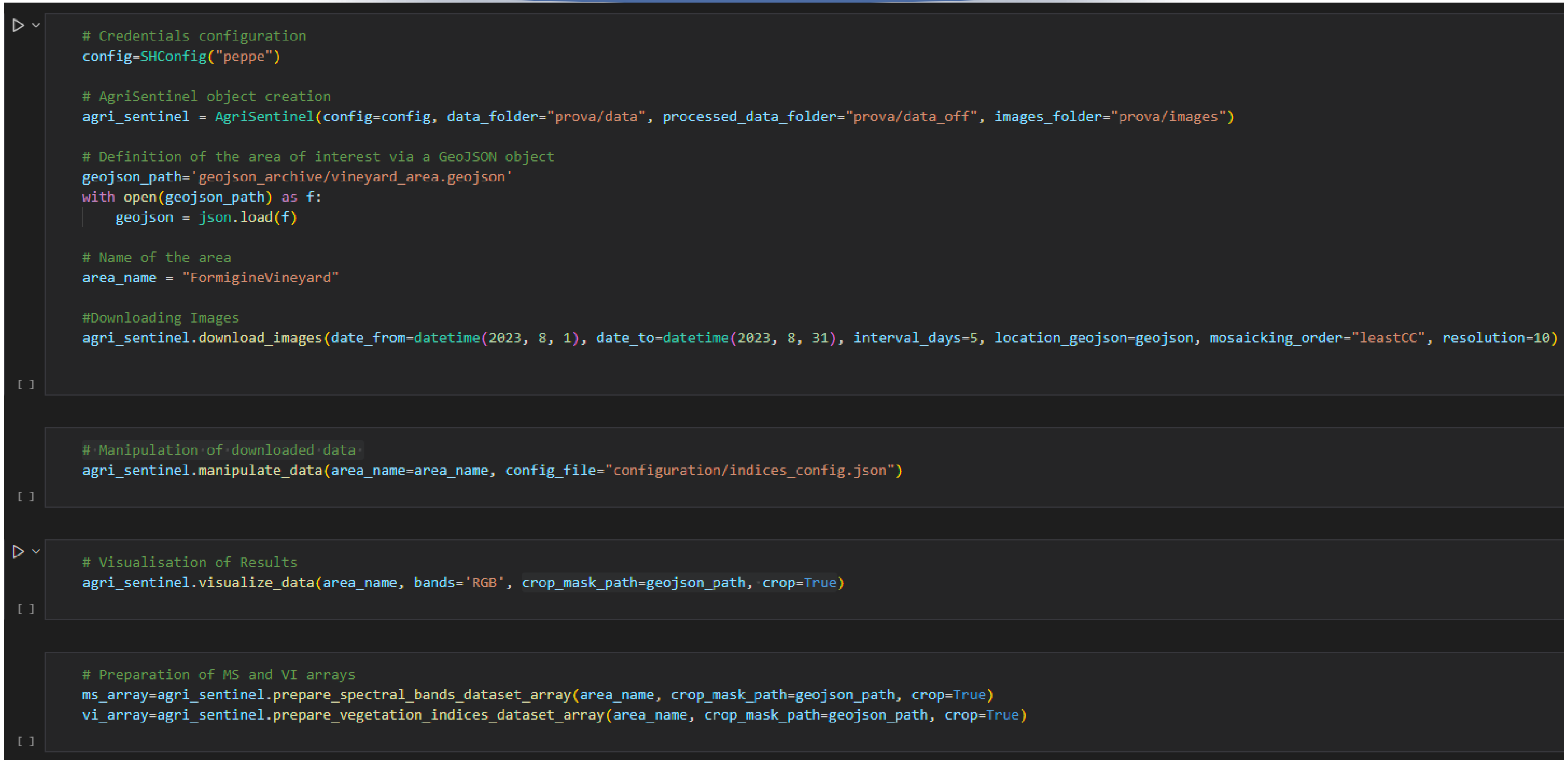
1) **AgriSentinel_Download:** Allows the download of satellite data using the sentinelhub API. Currently available are Sentinel2-L2A, but other satellites (Copernicus Constellation, LandSat, other commercial satellites) can also be integrated.

2) **AgriSentinel_DataManipulator:** Allows data manipulation for the creation of datasets (with images in tiff format) of multispectral images and vegetation indices (currently 19 have been implemented). It also allows images to be loaded into lists and numpy arrays to be used for analysis, e.g. for training and testing AI models.

3) **AgriSentinel_Visualiser:** Allows multispectral images and vegetation indices to be printed out for visualization.

4) **Agrisentinel:** serves as an interface for the previous modules.

A (BRIEF) EXPLANATION OF HOW AGRISENTINEL WORKS



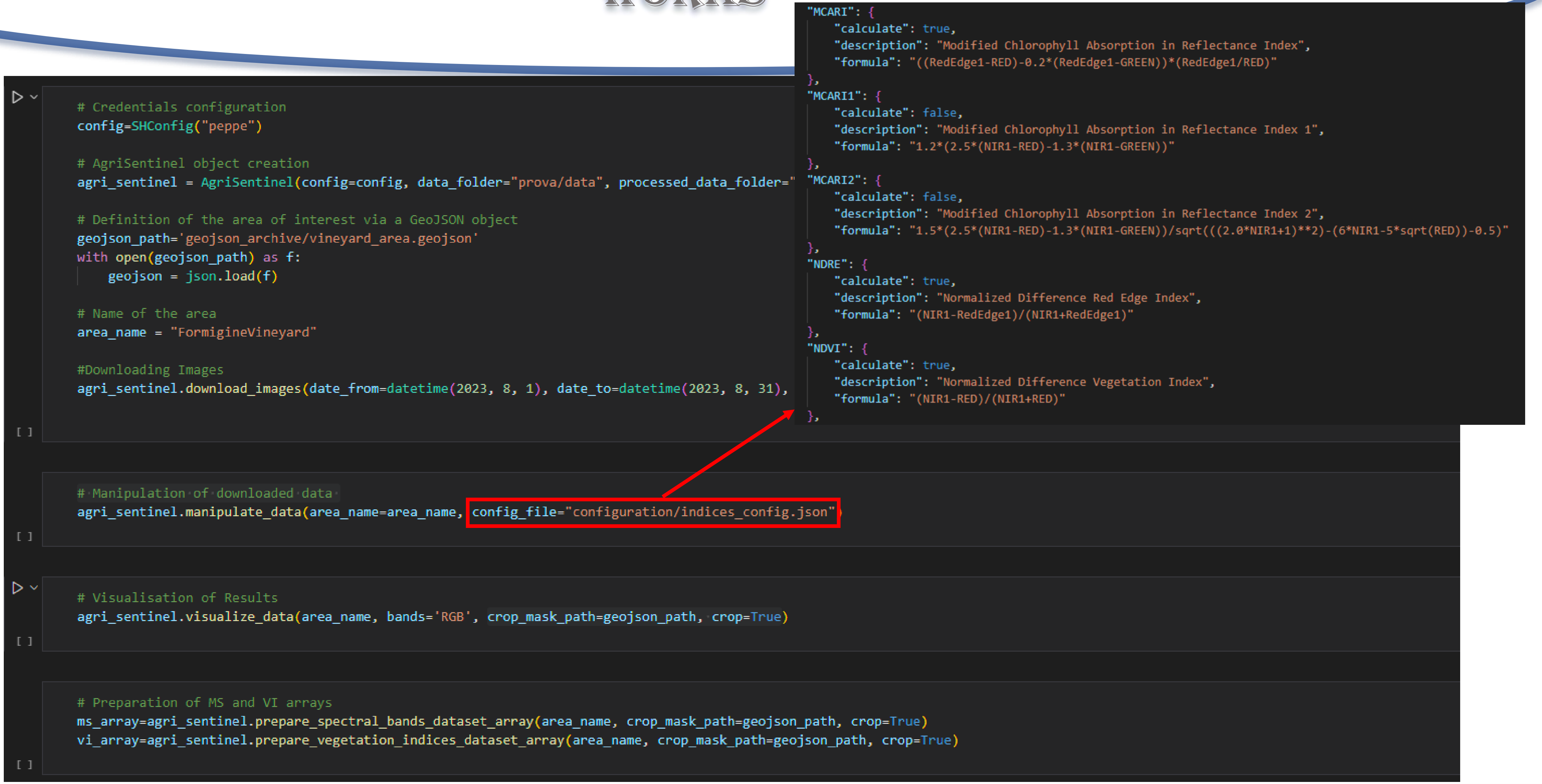
```
# Credentials configuration
config=SHConfig("peppe")

# AgriSentinel object creation
agri_sentinel = AgriSentinel(config=config, data_folder="prova/data", processed_data_folder="prova/data_off", images_folder="prova/images")

# Definition of the area of interest via a GeoJSON object
geojson_path='geojson_archive/vineyard_area.geojson'
with open(geojson_path) as f:
    geojson = json.load(f)

# Name of the area
area_name = "FormigineVineyard"

#Downloading Images
agri_sentinel.download_images(date_from=datetime(2023, 8, 1), date_to=datetime(2023, 8, 31), interval_days=5, location_geojson=geojson, mosaicking_order="leastCC", resolution=10)
```

```
# Manipulation of downloaded data
agri_sentinel.manipulate_data(area_name=area_name, config_file="configuration/indices_config.json")
```

```
# Visualisation of Results
agri_sentinel.visualize_data(area_name, bands='RGB', crop_mask_path=geojson_path, crop=True)
```

```
# Preparation of MS and VI arrays
ms_array=agri_sentinel.prepare_spectral_bands_dataset_array(area_name, crop_mask_path=geojson_path, crop=True)
vi_array=agri_sentinel.prepare_vegetation_indices_dataset_array(area_name, crop_mask_path=geojson_path, crop=True)
```

A (BRIEF) EXPLANATION OF HOW AGRISENTINEL WORKS



```
# Credentials configuration
config=SHConfig("peppe")

# AgriSentinel object creation
agri_sentinel = AgriSentinel(config=config, data_folder="prova/data", processed_data_folder="

# Definition of the area of interest via a GeoJSON object
geojson_path='geojson_archive/vineyard_area.geojson'
with open(geojson_path) as f:
    geojson = json.load(f)

# Name of the area
area_name = "FormigineVineyard"

#Downloading Images
agri_sentinel.download_images(date_from=datetime(2023, 8, 1), date_to=datetime(2023, 8, 31),
```

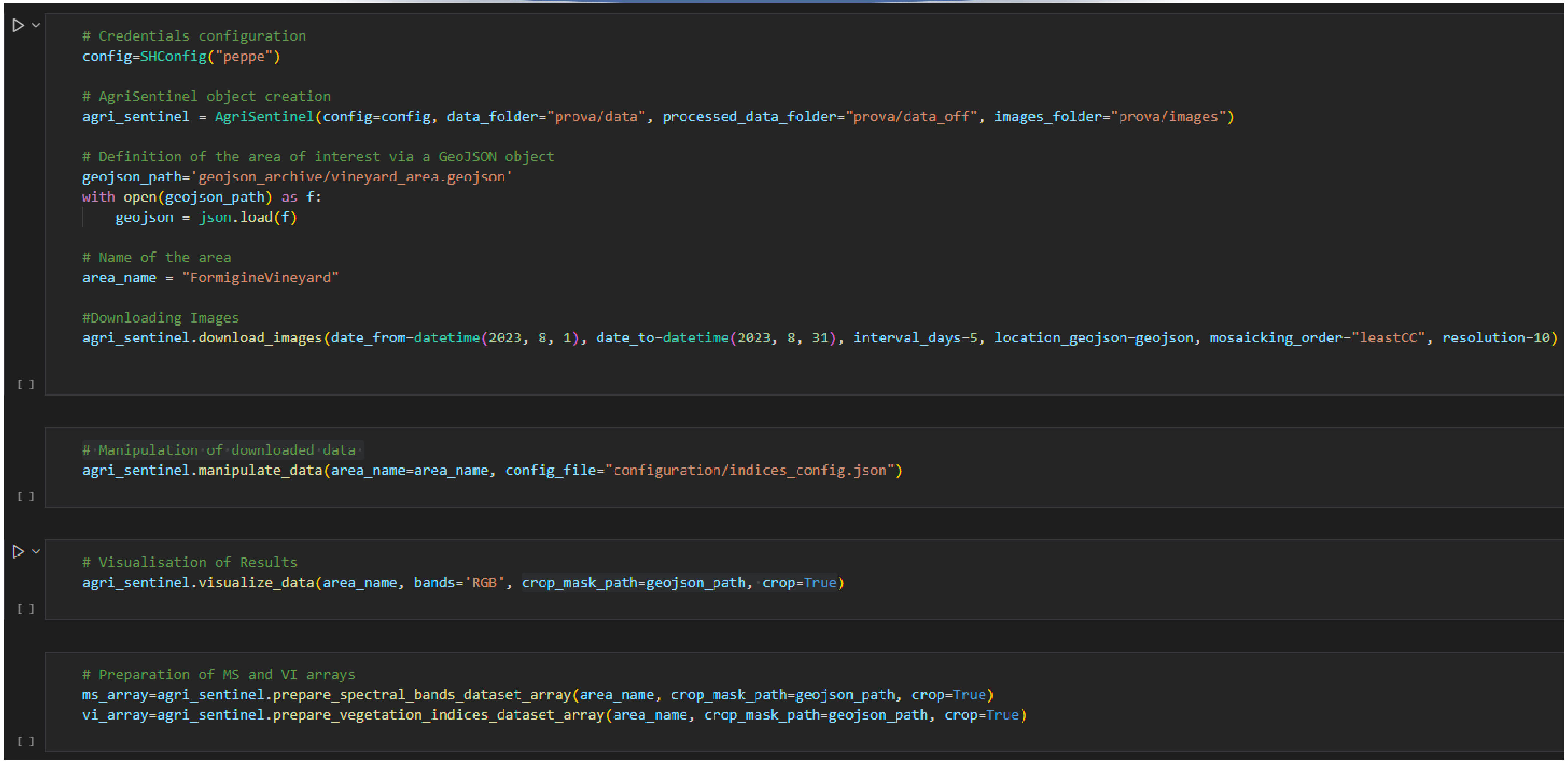
```
# Manipulation of downloaded data
agri_sentinel.manipulate_data(area_name=area_name, config_file="configuration/indices_config.json")
```

```
# Visualisation of Results
agri_sentinel.visualize_data(area_name, bands='RGB', crop_mask_path=geojson_path, crop=True)
```

```
# Preparation of MS and VI arrays
ms_array=agri_sentinel.prepare_spectral_bands_dataset_array(area_name, crop_mask_path=geojson_path, crop=True)
vi_array=agri_sentinel.prepare_vegetation_indices_dataset_array(area_name, crop_mask_path=geojson_path, crop=True)
```

```
"MCARI": {
    "calculate": true,
    "description": "Modified Chlorophyll Absorption in Reflectance Index",
    "formula": "((RedEdge1-RED)-0.2*(RedEdge1-GREEN))*(RedEdge1/RED)"
},
"MCARI1": {
    "calculate": false,
    "description": "Modified Chlorophyll Absorption in Reflectance Index 1",
    "formula": "1.2*(2.5*(NIR1-RED)-1.3*(NIR1-GREEN))"
},
"MCARI2": {
    "calculate": false,
    "description": "Modified Chlorophyll Absorption in Reflectance Index 2",
    "formula": "1.5*(2.5*(NIR1-RED)-1.3*(NIR1-GREEN))/sqrt(((2.0*NIR1+1)**2)-(6*NIR1-5*sqrt(RED))-0.5)"
},
"NDRE": {
    "calculate": true,
    "description": "Normalized Difference Red Edge Index",
    "formula": "(NIR1-RedEdge1)/(NIR1+RedEdge1)"
},
"NDVI": {
    "calculate": true,
    "description": "Normalized Difference Vegetation Index",
    "formula": "(NIR1-RED)/(NIR1+RED)"
},
```

A (BRIEF) EXPLANATION OF HOW AGRISENTINEL WORKS



```python
# Credentials configuration
config=SHConfig("peppe")

# AgriSentinel object creation
agri_sentinel = AgriSentinel(config=config, data_folder="prova/data", processed_data_folder="prova/data_off", images_folder="prova/images")

# Definition of the area of interest via a GeoJSON object
geojson_path='geojson_archive/vineyard_area.geojson'
with open(geojson_path) as f:
    geojson = json.load(f)

# Name of the area
area_name = "FormigineVineyard"

#Downloading Images
agri_sentinel.download_images(date_from=datetime(2023, 8, 1), date_to=datetime(2023, 8, 31), interval_days=5, location_geojson=geojson, mosaicking_order="leastCC", resolution=10)
```

```python
# Manipulation of downloaded data
agri_sentinel.manipulate_data(area_name=area_name, config_file="configuration/indices_config.json")
```

```python
# Visualisation of Results
agri_sentinel.visualize_data(area_name, bands='RGB', crop_mask_path=geojson_path, crop=True)
```

```python
# Preparation of MS and VI arrays
ms_array=agri_sentinel.prepare_spectral_bands_dataset_array(area_name, crop_mask_path=geojson_path, crop=True)
vi_array=agri_sentinel.prepare_vegetation_indices_dataset_array(area_name, crop_mask_path=geojson_path, crop=True)
```

(SOME) OPEN POINTS IN DEVELOPMENT

- Make the code more general to deal with images from **multiple satellites** (commercial and free).

- Develop a module to manage **image analysis** (time series, index averages) and AI methods.

- Choosing or developing algorithms for **cloud management**.

- Adding more **vegetation indices** or other variables that can be calculated from satellite images.

- Improving the library structure and writing documentation (ongoing).