Contributing to the DDAL guidelines

Jurrian Doornbos

May 2025

This document outlines steps for contributing to the ICAERUS Drone Data Analytics Library.

We divided Drone Data Analytics in two parts: Models and Data. Models are on GitHub and Data is on Zenodo (or any other file hosting service. Zenodo just gives a nice DOI).

If any help is required. Me and Esther are on the case and are always welcome to help out on the subject matter! Just send us an e-mail, and we will be in touch! Either for resolving through e-mail or organizing a meeting!

Jurrian: jurrian.doornbos@wur.nl
Esther: esther@noumena.io

Funded by
the European Union
Grant agreement N° 101060643

Table of contents

Funded by
the European Union
Grant agreement N° 101060643

Models on GitHub

Understanding GitHub and Repositories: Basics

GitHub is a web-based platform for version control and collaboration, allowing individuals and teams to manage and share code effectively. Here's a breakdown of the essentials:

---

1. What Is GitHub?

GitHub is built around **Git**, a distributed version control system that tracks changes in files, especially code. It is popular among developers for:

- **Version Control**: Keeping track of changes to your files over time.
- **Collaboration**: Allowing multiple people to work on a project simultaneously.
- **Hosting Projects**: Sharing code publicly or privately.

GitHub adds features like:

- A user-friendly interface.
- Tools for collaboration (issues, pull requests, discussions).
- Integration with CI/CD pipelines and project management tools.

---

2. What Is a Repository?

A **repository (repo)** is a project's storage unit. Think of it as a folder that contains:

- **Code**: The primary focus of the repository.
- **Documentation**: Instructions on how to use or contribute to the project (e.g., `README.md`).
- **Other Files**: Configuration files, datasets, or any resources required by the project.

Repositories are hosted on GitHub and can be either:

- **Public**: Anyone can view and contribute (depending on permissions).
- **Private**: Only accessible to authorized users.

---

3. Key Components of a Repository

- **Files and Folders**: Organize your project.
- **Commits**: Snapshots of changes made to the repository over time.
- **Branches**: Separate versions of the codebase to work on new features or fix bugs without affecting the main version.
- **Pull Requests**: Proposals to merge changes from one branch into another.
- **Issues**: Used for tracking bugs, tasks, or feature requests.

Funded by
the European Union
Grant agreement N° 101060643

- **Wiki/Docs**: Optional sections for detailed documentation.

You could also familiarize yourself with the following tutorial on GitHub:
https://docs.github.com/en/get-started/quickstart/create-a-repo
Or the informative video from Fireship: Git It? How to use Git and Github

The ICAERUS Community on GitHub

The ICAERUS-EU GitHub organization serves as a collaborative platform for managing and sharing code, documentation, models related to the ICAERUS-EU project.

1. Joining the ICAERUS-EU GitHub Organization

To become a member:

- **Provide Your GitHub Details**: Send an email to *jurrian.doornbos@wur.nl* with your GitHub username or associated email address.

2. Structuring Your Repository

- A repository can essentially be seen as a folder. The crucial part of this folder is the README.md file. This file will be opened on the webpage of the GitHub repository.
- Within ICAERUS we have standardized the README. The idea is that you follow this standard and fill it in accordingly when making the repository.
- The standard template can be found here: https://github.com/ICAERUS-EU/icaerus-repo-template
- The template ought to cover everything needed to setup your models.*The contents and folder structure etc. are up to yourself.*
- This means that you could different repositories for different models: such as https://github.com/ICAERUS-EU/uavgeo. Or you could do different models inside the same repositories, such as the UCs: https://github.com/ICAERUS-EU/UC1_Crop_Monitoring
- As long as there is a clear structure, and a good readme. We will go along!
- Content wise, a lot is possible, we tend to go for a lot of python code and notebooks because data analytics and machine/deep learning is almost synonymous with python. But also other programming languages, or explanations or tutorials, sensitivity analyses or anything surrounding drone analytics is welcome!

3. Adding Your Repository

If you need to create a new repository:

1. Go to the ICAERUS-EU GitHub page.
2. Click on the `Repositories` tab, then select `New`.
3. Enter a descriptive repository name (e.g., `UC1_Crop_Monitoring`).
4. Choose the `Public` visibility option and initialize with a `README` file.

If you already have an existing repository:

Funded by the European Union
Grant agreement N° 101060643

- **Transfer Ownership**:
    1. Navigate to your repository on GitHub.
    2. Click on the `Settings` tab.
    3. Scroll down to the `Danger Zone` section.
    4. Select `Transfer` and specify `ICAERUS-EU` as the new owner.
- **Update Local Clones**: After the transfer, clone the repository from its new location under the ICAERUS-EU organization. Ensure all internal links and references reflect the new URL structure.

4. Model platform.json

JSON (JavaScript Object Notation) is a lightweight, text-based format used for data interchange. It is easy for humans to read and write, and simple for machines to parse and generate, making it ideal for storing and transmitting structured data.

Key Features of JSON

1. **Structure**: Data is organized as key-value pairs in objects `{}` and lists of values in arrays `[]`.
2. **Data Types**: Supports strings, numbers, booleans, arrays, objects, and `null`.
3. **Interoperability**: Widely supported across programming languages and tools, ensuring compatibility.
4. **Extensibility**: Can be nested to represent complex hierarchical data.

The platform.json is our metadata standard to say something *about* the model. Which integrates with the ICAERUS Platform and ensures FAIR principles to be a bit more supported. Please find the standard platform.json over here: https://github.com/ICAERUS-EU/icaerus-repo-template/blob/main/platform.json

1. Basic Metadata

- **`title`**: The name of the model.
    - Example: `"Plant disease detection with YOLOv8"` describes the purpose of the model succinctly.
- **`creator`**: The model developer or contributor.
    - Example: `"Esther Vera"` specifies the individual responsible for the model.
- **`date`**: The date when the model was added or last updated.
    - Example: `"22/12/2023"` ensures version tracking.

2. Links

- **`icon`**: A URL pointing to an image or icon file representing the model visually.
    - Adds a specific, accessible visual marker.

Funded by
the European Union
Grant agreement N° 101060643

- **`url`**: A direct link to the repository or folder where the model resides.
  - Example: `"https://github.com/ICAERUS-EU/UC1_Crop_Monitoring/tree/main/models/01_plant_disease_detection_yolov8_v1"` directs users to the model's location.

3. Classification and Applications

- **`icaerus_applications`**: A list of broad application areas the model belongs to.
  - Example: `["crop-monitoring"]` ties the model to ICAERUS's crop monitoring domain.
- **`uses_cases`**: Specific use case identifiers linked to ICAERUS's structured documentation.
  - Example: `["1"]` refers to use case #1 in the project documentation.
  - For OCTs. This is filled in as “OCT” (see line 44 over here: https://github.com/ICAERUS-EU/ICAERUS_DDAL/blob/main/oct_platform.json)

4. Extra Fields

- **`extra`**: A placeholder for additional, project-specific key-value pairs.
  - Example: Empty keys `"key1": ""` and `"key2": ""` allow customization without breaking the schema.

5. Filters

Used to tag the model with specific attributes, making it easier to search for on the platform:

These filters are a work in progress. Fill them in how you want them. And can be anything! We will use these later for additional filtering and might change them around to align them better. But for now, fill in as many keywords as you think of.

- **`content`**: Tags about the technology and methodology used.
  - Example: `["Deep learning", "Computer vision", "Detection"]`.
- **`data-type`**: Input data types the model supports.
  - Example: `["RGB images"]`.
- **`subject`**: The subject or domain the model focuses on.
  - Example: `["Vineyards"]`.
- **`location`**: Geographical scope or relevance of the model.
  - Example: `["Spain"]`.
- **`model input`**: Expected data type/keyword for model input.
  - Example: `["Orthomosaic", “RAW Image”, “Tabular”]`.
- **`model output`**: Expected data type/keyword for model output.
  - Example: `["Bounding box", “Segmentation”]`.

Funded by
the European Union
Grant agreement N° 101060643

File Size Limitations GitHub

While GitHub is a robust platform, it has limitations on file sizes to ensure smooth performance and encourage proper resource management. Keep these in mind when sharing large model-weight files, which are encouraged to be hosted on Zenodo.

1. Single File Size:
    - Maximum file size: 100 MB.
    - Files larger than this will be rejected when you try to push them.
    - A warning is issued for files larger than 50 MB.
2. Repository Size:
    - There's no strict limit, but GitHub recommends keeping repositories below 1 GB.
    - Exceeding 5 GB may result in warnings, and performance issues can arise.
3. Git Large File Storage (LFS):
    - GitHub provides Git LFS for managing large files like datasets, videos, and high-resolution images.
    - Files tracked with Git LFS are stored outside the Git repository.
    - GitHub provides 1 GB of free storage for Git LFS per user or organization.

Funded by
the European Union
Grant agreement N° 101060643

ICAERUS Datasets on Zenodo

Zenodo is a repository for sharing and archiving research outputs, such as datasets, software, and publications. It's an excellent tool for ensuring your work is accessible, citable, and preserved over time. Here's how to upload your content and link it to the ICAERUS-HE community.
What Is Zenodo?

Zenodo is a free and open platform developed by CERN under the European OpenAIRE project. Key benefits include:

- DOI Assignment: Automatically generates a DOI (Digital Object Identifier) for your uploaded work, making it citable.
- Community Tagging: Enables association with specific research groups or initiatives, like ICAERUS-HE.

Step-by-Step Tutorial for Uploading to Zenodo

1. Create an Account

Go to Zenodo. Sign up using your email or log in via ORCID, GitHub, or an existing Zenodo account.

2. Prepare Your Upload

- Content: Ensure you have all necessary files (e.g., datasets, software, publications) prepared for upload.
- Metadata: Gather details such as title, authors, description, keywords, and funding information.
- Format: If uploading datasets, consider compressing large files into .zip or .tar.gz.

3. Start a New Upload

- Click the Upload button on the Zenodo homepage.
- Drag and drop your files or select them manually.

4. Fill in Metadata

- Title: Enter the title of your work.
- Authors: List all contributors.
- Description: Provide a clear and concise summary of the content.
-

Small introduction (1-2 lines) about what will be found in the dataset and what is the dataset about (classification, detection, segmentation, for building orthomosaics, parameters). This dataset encompasses the following data:

- Folder: description (number of images / subfolders, type of images, size of images, tables, parameters).
    - Subfolder: description (number of images, type of images, size of images, tables, parameters).
- Folder: description (number of images, type of images, size of images, tables, parameters).

Funded by the European Union
Grant agreement N° 101060643

A paragraph explaining the flight conditions or specifications about how this data was collected in detail. This can include: which drone, camera, resolution, other sensors, altitude, where, time when the data was collected, processing parameters, ….
Possible applications: a description of what you could use this dataset for.

- Keywords: Add terms that make your work discoverable.
- Funding: Include relevant grant IDs (ICAERUS Grant ID:

5. Select the ICAERUS-HE Community

- Scroll to the Communities section.
- Search for and select ICAERUS-HE.
- Associating with this community ensures that your work is discoverable by other ICAERUS-HE researchers and aligns with the project's goals.
- If you don't see the ICAERUS-HE community listed, contact your project lead or the Zenodo support team.

6. Choose Access Rights

- Open Access: Ideal for most research outputs.
- Embargoed Access: Set a future date for public release if necessary.
- Restricted Access: Limit access and approve requests manually.

7. License Your Work

Select an appropriate license:

- Creative Commons (CC) licenses for datasets or publications.
- GNU GPL, MIT, or Apache License for software.

8. Publish

- Review all metadata and files to ensure accuracy.
- Click Publish to upload your work to Zenodo.
- Once published, Zenodo generates a DOI that can be shared or cited in publications.

Additional Tips

- Versioning: Zenodo supports versioning for uploads. Update your work without losing the DOI.
- High File Sizes: For large files exceeding 50 GB, consider contacting Zenodo support for special arrangements.

Funded by the European Union
Grant agreement N° 101060643

The ICAERUS Data json:

Just like the models, we provide a json file for platform integration for datasets as well. The template can be found over here:
https://github.com/ICAERUS-EU/Zenodo_Datasets/blob/main/datasets_template.json

When you upload a dataset, you can do two things. 1. e-mail jurrian.doornbos@wur.nl with your filled in dataset.json. Or 2. you create a new Issue to add your provided platform.json over at Github: Zenodo_datasets (https://github.com/ICAERUS-EU/Zenodo_Datasets/issues). We will do the double checks and integrate them in the main ones.

Explanation of Fields in `dataset.json`

The .json file opens and closes with curly brackets: {}

1. **Dataset Metadata**

- **`name`**: The title of the dataset.
  - Example: `"Dataset name"` clearly identifies the dataset.
- **`author`**: A list of authors or contributors to the dataset.
  - Example: `["Author1", "Author2", "Author3"]` specifies contributors.
- **`date`**: Years when the dataset was collected or published.
  - Example: `[2017, 2018, 2019]` reflects the temporal scope of data collection.
- **`link`**: URL to access or learn more about the dataset.
  - Example: `"https://example.com"` provides direct access.
- **`description`**: A concise summary of the dataset's purpose.
  - Example: `"A brief description of the dataset and its purpose."` offers context.

---

2. **Dataset Parameters**

The `parameters` field provides additional details about the dataset. and are in new curly brackets. These are not fully implemented yet an currently under debate. But are always welcome to be included!

- **`location`**: Describes the location where the dataset was collected.
  - In format `"Location description or coordinates: Location, Country"`.
- **`coordinates (wgs84)`**: Latitude and longitude of the data collection area in WGS84 format.
  - In format: `"Latitude, Longitude"`.
- **`sensor type`**: Types of sensors used to collect data.
  - In a list in a choice from: `["RGB", "Multispectral", "Thermal", "Hyperspectral", "UAV-LiDAR", "TLS"]` indicates diverse data modalities.
- **`sensors`**: Specific sensor models used.

Funded by the European Union
Grant agreement N° 101060643

    - In a list in a choice from: `["DJI FC300X", "Parrot Sequoia"]` provides brand and type details.
- **`flight height (m)`**: The altitude (in meters) at which UAVs captured data.
    - Example: `[10, 30, "Unspecified"]`
- **`contents`**: Summarizes the dataset's contents and measurements.
    - Example: `"A brief summary of the data contents and measurements"` helps users understand the dataset at a glance.

The format of a dataset.json:

```
{
    "name": "Dataset name",
    "author": [
        "Author1",
        "Author2",
        "Author3"
    ],
    "date": [
        2017,
        2018,
        2019
    ],
    "link": "https://example.com",
    "description": "A brief description of the dataset and its purpose.",
    "parameters": {
        "location": "Location description or coordinates: Location, Country",
        "coordinates (wgs84)": "Latitude, Longitude",
        "sensor type": [
            "RGB",
            "Multispectral",
            "Thermal",
            "Hyperspectral",
            "LIDAR",
            "TLS",
            “GPR”
        ],
        "sensors": [
            "Specific sensors used for data collection. in format: Brand type, e.g.
DJI FC300X, Parrot Sequoia"
        ],
        "flight height (m)": [
            10,
            30,
            "Unspecified"
        ],
        "contents": "A brief summary of the data contents and measurements"
    }
}
```


Funded by
the European Union
Grant agreement N° 101060643