drones

MDPI

Technical Note

DRONE2REPORT: A Configuration-Driven Multi-Sensor Batch-Processing Engine for UAV-Based Plot Analysis in Precision Agriculture

Nelson Nazzicari [1,*], Giulia Moscatelli [2], Agostino Fricano [3], Elisabetta Frascaroli [4], Roshan Paudel [4], Eder Groli [5], Paolo De Franceschi [5], Giorgia Carletti [3], Nicolò Franguelli [1] and Filippo Biscarini [2]

[1] Research Centre for Animal Production and Aquaculture, Council for Agricultural Research and Analysis of Agricultural Economics, Viale Piacenza, 29, 26900 Lodi, Italy; nicolo.franguelli@crea.gov.it
[2] Institute of Biology and Biotechnology in Agriculture (IBBA), National Research Council (CNR), Via A. Corti, 12, 20133 Milan, Italy; giulia.moscatelli@ibba.cnr.it (G.M.); filippo.biscarini@cnr.it (F.B.)
[3] Research Centre for Genomics and Bioinformatics, Council for Agricultural Research and Analysis of Agricultural Economics, Via San Protaso 69, 29017 Fiorenzuola d'Arda, Italy; agostino.fricano@crea.gov.it (A.F.); giorgia.carletti@crea.gov.it (G.C.)
[4] Department of Agricultural and Food Sciences, University of Bologna, Via G. Fanin, 44, 40127 Bologna, Italy; elisabetta.frascaroli@unibo.it (E.F.); roshan.paudel@unibo.it (R.P.)
[5] Research & Development, SIS-Società Italiana Sementi, Via Mirandola, 5, 40068 San Lazzaro di Savena, Italy
* Correspondence: nelson.nazzicari@crea.gov.it

Highlights

What are the main findings?

- UAV imagery plays an ever-growing role in agriculture, particularly in breeding and crop monitoring.
- Data-processing pipelines are typically a mixture of standard and custom software, leading to inconsistencies and difficulties in reproducibility.

What are the implications of the main findings?

- A new dedicated Python (3.12+) library, DRONE2REPORT, is released as a standardized and customizable open-source framework for processing orthorectified UAV images in agriculture.
- We demonstrate a wide range of applicability and integration, covering both traditional approaches (e.g., vegetation indices) and innovative techniques (e.g., deep learning and computer vision).

Abstract

Unmanned aerial vehicles (UAVs) have become indispensable tools in precision agriculture and plant phenotyping, enabling the rapid, non-destructive assessment of crop traits across space and time. Equipped with RGB, multispectral, thermal, and other sensors, UAVs provide detailed information on canopy structure, physiology, and stress responses that can guide management decisions and accelerate breeding programs. Despite these advances, the downstream processing of UAV imagery remains technically demanding. Converting orthomosaics into standardized, biologically meaningful data often requires a combination of photogrammetry, geospatial analysis, and custom scripting, which can limit reproducibility and accessibility across research groups. We present DRONE2REPORT, an open-source python-based software that processes orthomosaics from UAV flights to generate vegetation indices, summary statistics, derived subimages, and text (html) reports, supporting both research and applied crop breeding needs. Alongside the basic structure and functioning of DRONE2REPORT, we also present five case studies that illustrate practical applications common in UAV-/drone-phenotyping of plants: (i) thresholding to

Check for updates

Academic Editors: Hongquan Wang, Taifeng Dong and Liming He

Received: 19 January 2026
Revised: 14 April 2026
Accepted: 14 April 2026
Published: 18 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

remove background noise and highlight regions of interest; (ii) monitoring plant phenotypes over time; (iii) extracting information on plant height to detect events like lodging or the falling over of spikes; (iv) integrating multiple sensors (cameras) to construct and optimize new synthetic indices; (v) integrate a trained deep learning network to implement a classification task. These examples demonstrate the tool's ability to automate analysis, integrate heterogeneous data and models, and support reproducible computation of agronomically relevant traits. DRONE2REPORT streamlines orthorectified UAV-image processing for precision agriculture by linking orthomosaics to standardized, plot-level outputs. Its modular, configuration-driven design allows transparent workflows, easy customization, and integration of multiple sensors within a unified analytical framework. By facilitating reproducible, multi-modal image analysis, DRONE2REPORT lowers technical barriers to UAV-based phenotyping and opens the way to robust, data-driven crop monitoring and breeding applications.

Keywords: drone phenotyping; orthorectified image processing; software package; precision agriculture; UAV; vegetation index; NDVI; deep learning; neural networks

1. Introduction

Advances in precision agriculture and plant phenotyping increasingly rely on high-throughput, non-destructive methods for measuring crop performance under field conditions. Remote sensing platforms have become central to this effort, providing quantitative information on canopy structure, physiology, and stress responses. Among these, unmanned aerial vehicles (UAVs, or drones) are especially attractive due to their flexibility, affordability, and ability to generate repeated observations at plot or field scale. UAV-mounted sensors—ranging from RGB cameras to multispectral, thermal, and LIDAR (Laser Imaging, Detection And Ranging) units—enable detailed measurement of vegetation indices, canopy temperature, and topography that can inform breeding programs and management decisions [1].

Although the potential of UAV imagery for plant science is well recognized, the software ecosystem supporting its analysis remains fragmented. Many researchers rely on commercial photogrammetry suites for orthomosaic production, followed by manual processing in GIS software or custom scripts for index calculation and region-of-interest (ROI) extraction. This workflow can be labor-intensive, error-prone, and difficult to reproduce across experiments or research groups. Existing open-source solutions are usually not designed for agriculture applications ([2]), have sometimes very specific targets (e.g., seeds [3], berries [4]), focus on single data types (e.g., RGB or multispectral), provide limited support for multi-modal datasets, or lack transparent configuration and automation mechanisms.

As UAV datasets grow in size and complexity, there is a growing need for tools that combine reproducibility, flexibility, and ease of use, while remaining accessible to plant researchers without advanced programming expertise. In a typical elaboration pipeline, after photogrammetry comes a wide variety of analyses that are highly application-dependent, and often implemented using ad hoc solutions. DRONE2REPORT was developed to address this step, providing a tool able to generate standardized and reproducible post-photogrammetry results. It is designed specifically for agriculture (e.g., a field with multiple plots each accommodating one genotype) and implements a configuration-driven batch processing engine that streamlines the transformation of orthomosaics into standardized outputs—such as vegetation indices, ROI-based statistics, and summary reports—across multiple sensor types. By emphasizing reproducibility, modularity, and compatibility with common research workflows, DRONE2REPORT aims to lower the barrier to adoption of UAV-based phenotyping and facilitate robust comparisons across experiments, genotypes, and environments. In this paper, we present the technical details of this new tool, followed by five case studies describing examples of its application, a comparison with other available software packages, and a discussion of the merits, limitations and future expansions of DRONE2REPORT.

2. Methods

2.1. Data Workflow

The general workflow of DRONE2REPORT is reported in Figure 1. Here, we adopt the following definitions: orthophotos (or orthorectified images) are single drone-captured images that have been geometrically corrected (perspective, camera tilt, etc.), while orthomosaics are a collection of orthophotos stitched together (taking care of overlaps, etc.) [5]. The first step is data acquisition, with the drone(s) flying over the field of interest and collecting a series of overlapping images, which are then collated together into an orthomosaic. This computational step produces a georeferenced composite image and corrects for several technical issues such as perspective distortion, map projection, and lens aberrations. The generation of orthophotos and orthomosaics is now considered a well-established process, thanks to the availability of mature open-source tools such as OpenDroneMap [6], MicMac [7], and COLMAP [8]. Together with the orthomosaic image (typically in the .tif format), a digital elevation model (DEM) file is also produced, which describes the three-dimensional structure of the field. Depending on the experiment, radiometric calibration using reflectance panels can also be required.

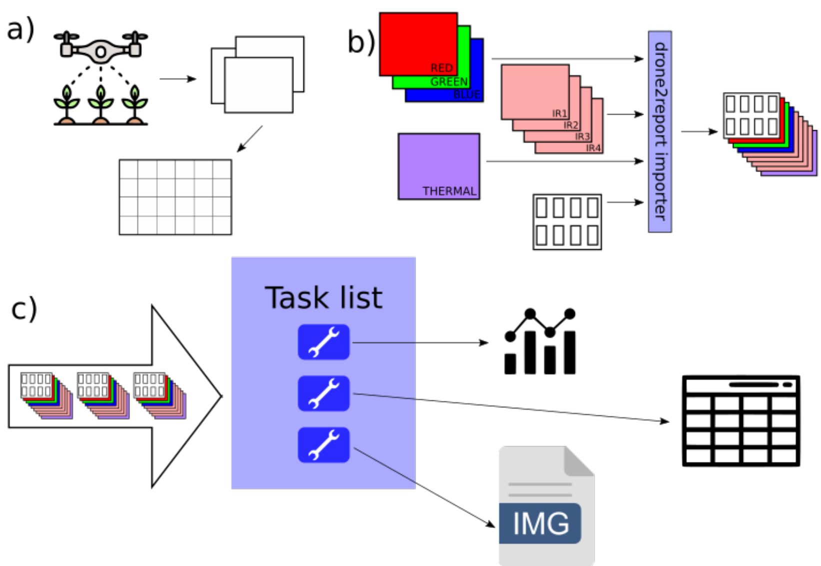

Figure 1. General application workflow. Panel (**a**): drones capture raw, partial images of the field (depending on the mounted sensors). All images from a single flight are collated in a unique orthomosaic. Panel (**b**): orthomosaics coming from one or more sensors, together with a shape file that specifies the ROIs, enter DRONE2REPORT and are converted to an internal *dataset* object. Panel (**c**): all datasets are input, in turn, to all queued tasks, each task producing its specific output (e.g., tables, other images, etc.).

In a typical agricultural application, several flights are performed over the same field during a growing season. In fact, the flights may start on the empty field just after sowing, so that the first orthomosaic and DEM file represent the baseline for all successive flights. On this empty field, a "shape file" is traced, containing the shapes and placements of all regions of interest (ROIs). Typically, each plot becomes an ROI, with the edge case of the entire field comprised of a single ROI. Defining ROIs is essential, because the area scanned by the drone is always larger than the actual target field surface and typically covers borders, utility roads and other external elements.

Orthophotos are typically multichannel images, with common cases being three channels (visible RGB spectrum), while multispectral images may include additional bands, such as near-infrared (NIR). Thermal data is encoded as a single channel and possibly rendered as grayscale images, with each pixel corresponding to a single temperature read. Even elevation models (DEM files) can, at this stage, be considered as a valid data input: a single channel image where each pixel corresponds to a measured height. The number of channels is typically maintained when orthophotos are then combined in orthomosaics.

DRONE2REPORT is designed to receive one or more orthomosaics as input. In the elaboration chain, it is thus placed after acquisition, photogrammetry, and, if needed, radiometric calibration. If more than one orthomosaic is provided, they will be internally stacked into a new multichannel image called *dataset*. For example, providing an RGB image plus a thermal image will create a four-channel RGBT *dataset*. Differences in image sizes and resolutions are solved via cubic interpolation. Differences in the spatial reference system (SRS) are handled via reprojection of all data into the reference taken from the first input orthomosaic. This results in the creation of a single tensor-like data structure with a single shape (width and length) and as many channels as the sum of all input image channels. This unified representation enables performing algebraic operations on all combined channels simultaneously, and even combine data coming from different flights and sensors. Even when multiple orthomosaics are combined into a *dataset*, the ROIs are specified only once.

One *dataset* can thus be composed of several orthomosaics, and several *dataset*s can be loaded. DRONE2REPORT then proceeds to feed all the data to a queue of tasks. As of the current implementation, a number of tasks are implemented for computing various vegetation indices, rescaling and extracting subimages, and preparing data for successive elaboration (e.g., training neural networks).

After all tasks have run, DRONE2REPORT will run the rendering queue. "Render" items are similar to the tasks but are designed to collect results of other analyses and produce summary and synthetic information. One of the render tasks is called "report" and is configured to collect and display all the produced log files. Moreover, the task adds a 2D rendering of the sampling point, color coded using the computed indexes.

2.2. *Software Implementation*

DRONE2REPORT is a Python-based, open source, free software. It is written to be integrated in standard elaboration pipelines aimed at agricultural drone images, both for research and industrial applications. As such, it has no GUI and runs, ideally server-side, on data files and configuration (.ini) files. It has an extensive logging functionality and can be used out of the box without any Python coding. However, it is designed with modularity in mind, allowing for straightforward extension through the addition of new tasks.

An example configuration (.ini) file is provided in Listing 1, demonstrating how to compute two vegetation indices from a single RGB image. The basic structure of the configuration file includes three main sections: (i) DEFAULT specifies base parameters like the input and output folders, the number of cores (processing units) to be used, and whether a verbose output log is desired; (ii) DATA specifies the image file name (with some metadata, i.e., date, description, etc.), its type (only tif_multichannel is currently supported), the value that indicates missing pixel data (e.g., -1), the channels (total and visible), the maximum pixel intensity value, the path to the shape file; (iii) TASK specifies which vegetation indices will be computed. More details on how to use DRONE2REPORT can be found in the on-line documentation (https://drone2report.readthedocs.io/en/latest/) (accessed on 13 April 2026). Once the configuration file is ready, DRONE2REPORT will be run from the shell command line.

Listing 1. Example of configuration file to run DRONE2REPORT. This takes the form of a standard .ini file, divided in sections (between square brackets) and supporting comments (using the hash symbol).

```
[DEFAULT]
#these parameters are shared by all sections
infolder=</base/folder>
outfolder=</folder/for_analysis_results>
logfolder=</folder/for_log_files>
cores=4
skip_if_already_done=False
verbose=True

[DATA image]
#this section specifies an input orthomosaic
active=True
meta_date=2024/03/25
meta_time=1.45 pm
meta_desc=Barley field barley, March 2024, RGB flight
type=tif_multichannel
orthomosaic=${DEFAULT:infolder}/<img-name.ext>
channels=red,green,blue
visible_channels=red,green,blue
max_value=255
nodata=-1
shapes_file=${DEFAULT:infolder}/<img.shp>
shapes_index=id

[TASK indices]
#this section tells DRONE2REPORT to compute
#two indices: GLI and HUE
active=True
outfolder=${DEFAULT:outfolder}/<results>
indices=GLI,HUE

[RENDER report]
#this section details the output html report
active=True
report_file=${DEFAULT:outfolder}/d2r.html
title=Drone2Report analysis
author=Your name
#all .csv files in this folder, if generated by
#index task, will be loaded and plotted
index_folder=${DEFAULT:outfolder}/<results>
```

The modularity of DRONE2REPORT can be leveraged at several levels. The easiest customization would be to add new specialized vegetation indices to the proper Python file (e.g., `matrix_returning_indices.py` file in the folder `/drone2report/d2r/tasks/` from the GitHub repository). Listing 2 reports an example of such a function.

A more powerful intervention consists of designing a new TASK, which will then require its own section in the .ini file. For that purpose, a TASK template is provided in the documentation of the tool. Finally, new types of DATA could be supported, for which some care would be required so as to curate the exposed functionalities. New indices and functionalities can be proposed and eventually integrated in the main software codebase via GitHub pull requests.

Note that DRONE2REPORT anticipates the most common needs of agricultural experiments. For example, when a vegetation index is computed, the software already summarizes it through a series of descriptive statistics, which are collated and saved in tabular format.

Internally, DRONE2REPORT uses the GDAL [9] and GeoPandas [10] libraries for manipulation of georeferenced images. DRONE2REPORT is released with a custom Conda environment for multi-platform installation. The DRONE2REPORT open-source code is available online in the following GitHub repository: https://github.com/ne1s0n/drone2report (accessed on 13 April 2026).

Listing 2. Example of function computing the GLI vegetation index on a passed image.

```python
def GLI(img, channels):
  """Green leaf index, uses red, green, blue channels"""
  try:
    red   = img[:,:,channels.index('red')]
    green = img[:,:,channels.index('green')]
    blue  = img[:,:,channels.index('blue')]

  except ValueError:
    return np.nan

  return(
    (2.0*green - red - blue) /
    (2.0*green + red + blue)
  )
```

3. Results

To illustrate how to use DRONE2REPORT for image processing from drone phenotyping, and to show some potential and useful applications of vegetation indices, we present five case studies: (i) thresholding; (ii) evolution of vegetation indices over time; (iii) analysis of height; (iv) combining multiple sensors and index optimization; (v) leveraging a deep neural network for image classification. The code and data to reproduce these case studies can be found at https://github.com/ne1s0n/paper-drone2report (accessed on 13 April 2026). We emphasize that the presented case studies are not original research work, but mere examples of possible uses of DRONE2REPORT.

3.1. Case Study N. 1: Thresholding

In the first case study, we use DRONE2REPORT to calculate a vegetation index after applying different value thresholds to highlight the regions of interest. Thresholding can be useful to remove alien elements from the computation. In a typical case, it is desirable to compute a vegetation index only on the areas actually covered by plants, excluding the soil. In this example, we compute the Green Leaf Index (GLI, see Formula (1)), whose positive values tend to identify green vegetation (e.g., leaves and stems), while negative values tend to reflect soil/non-vegetation [11]. Our objective might be to calculate the proportion of vegetation to soil, or to compare the green intensity of different fields or over time. In both cases, we may want to remove the image background (i.e., "noise", like a road, a pond or just bare soil) which can be achieved via thresholding [12].

To illustrate the removal of background noise, we first use one RGB image of tobacco leaves from the GitHub repository (https://github.com/tanghaibao/jcvi/wiki/GRABSEEDS (accessed on 13 April 2026) of the GRABSEED software [3] (Figure 2).

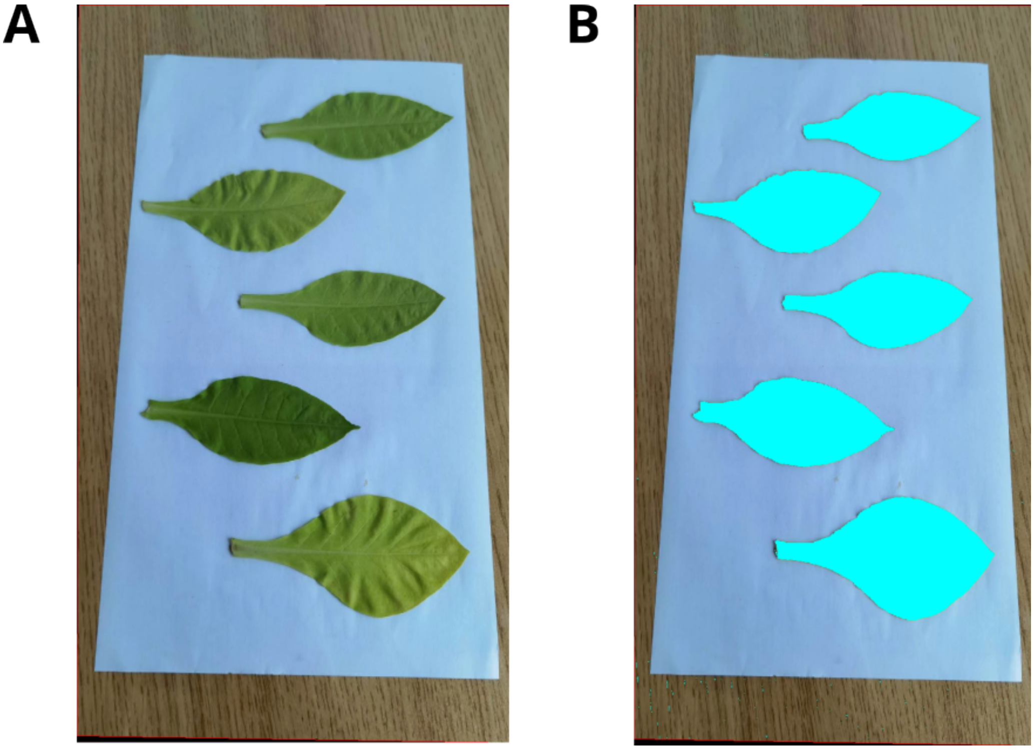

Figure 2. Example RGB image of tobacco leaves (*Nicotiana tabacum*). (**A**) The original image of the tobacco leaves as downloaded from https://www.dropbox.com/s/is4jrmlmmcfdpdp/test-data.zip?dl=1 (accessed on 13 April 2026); (**B**) the same image with threshold 0.1: the pixels highlighted in turquoise are those that will be used by DRONE2REPORT for the calculations of the vegetation indices.

In this example, our objective is to calculate the GLI vegetation index for the tobacco leaves. GLI is computed per-pixel using the following formula:

$$\mathrm{GLI}(\mathbf{p}) = \frac{(2 \cdot p_{green} - p_{red} - p_{blue})}{(2 \cdot p_{green} + p_{red} + p_{blue})} \tag{1}$$

where $\mathbf{p} := [p_{red}, p_{green}, p_{blue}]$ is a single pixel of an RGB image and contains the normalized intensities of the three channels.

With reference to Figure 2, the aim is to remove the background table (brownish) and the light-blue paper-cloth from the calculation. As a first step, we computed GLI on the image and obtained the distribution reported in Figure 3A. The image highlights two clear clusters, one around zero, the other larger than 0.2. These two groups correspond to the non-leaf background and to the tobacco leaves, with different intensities of green. Thus, a threshold of 0.1 would separate the two clusters. This value is incorporated when invoking DRONE2REPORT [TASK index]. We obtain a resulting GLI average value of 0.289 ± 0.07, calculated only from the leaves. If we apply the same procedure to other images of tobacco leaves, we can then compare the corresponding GLI values.

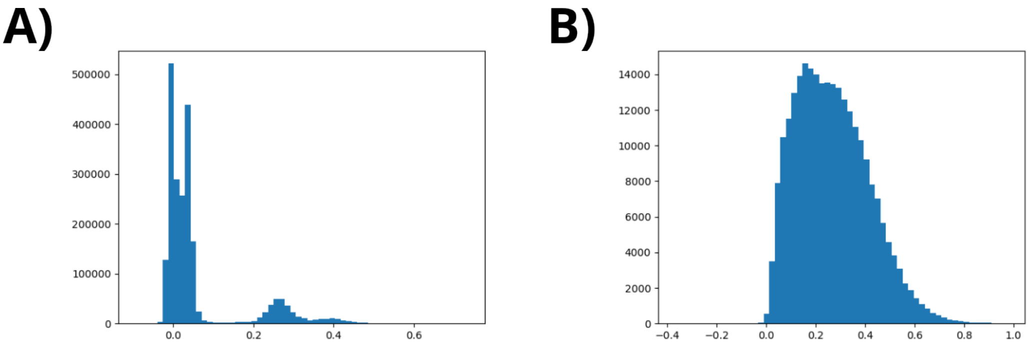

Figure 3. Distribution of values for the vegetation index GLI for: (**A**) the tobacco leaves image; and (**B**) for the barley field plot image.

In a more realistic case, the GLI value clusters from different segments of the image may not be as clear as in the tobacco-leaves example. An example is reported in Figure 4, panel A: a drone-captured RGB image (drone: DJI Mavic 3 Enterprise with integrated CMOS camera; resolution: 0.33 cm/pixel; overlap: 80%; sidelap: 80%) of a single plot from a barley field in Fiorenzuola d'Arda (PC), Italy, 44°55′27″ N, 9°54′47″ E, taken on March 25th 2024. Flight ground references were obtained via ground control points and GNSS ("Polyploidbreeding" research project: https://polyploidbreeding.ibba.cnr.it/ (accessed on 13 April 2026). For this image, the distribution of GLI index values is unimodal (no clusters: Figure 3B).

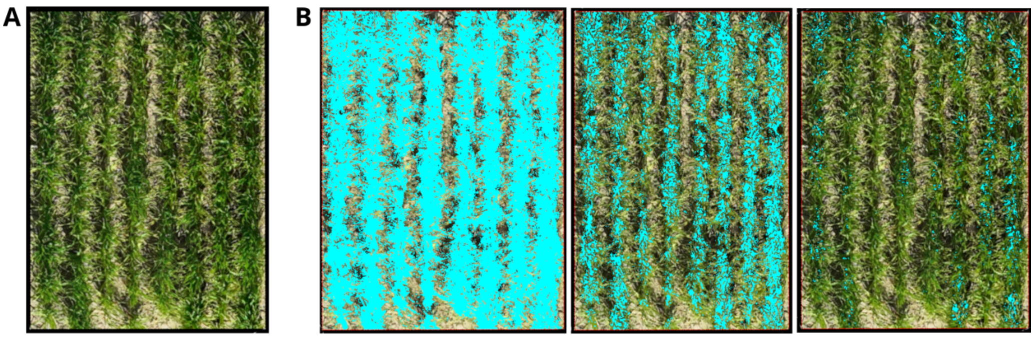

Figure 4. Example RGB image of one single barley field plot. (**A**) The original image of the barley field: all pixels would be used for index calculations; (**B**) the same plot with increasing thresholds for the index values: 0.2, 0.4, 0.6 (rescaled in $[0, 1]$). As the threshold is increased, fewer and fewer pixels are used for the calculations of the index (indicated in turquoise).

In such a case, no clear-cut threshold could possibly separate all background pixels. We used DRONE2REPORT [TASK thumbnail] to generate copies of the original image with increasing GLI thresholds: from 0.1 to 1 (Figure 4B, shows the image with thresholds 0.2, 0.4 and 0.6). We then calculated the GLI index using only the pixels above the threshold, from the original image (no threshold, all pixels used), to the maximum 0.9 threshold. The DRONE2REPORT [TASK index] was used for index calculations, each time specifying the corresponding threshold (if any). Results are shown in Table 1.

We see that the number of pixels used decreases with the threshold applied. Not surprisingly, given that GLI measures the green intensity of pixels in the image, GLI values increase with the threshold, from 0.26 to 0.93: essentially, since Figure 4A is a barley field, we are progressively getting rid of the soil and of the less green parts of the plants. Depending on the application, it may be desirable to focus only on vegetation, or even on the plants' greenest parts. In contrast, it may be acceptable to remove only pixels with negative GLI values (GLI > 0.0), which are definitely not green. This is reported in Table 1, second row.

Note that in these examples, for simplicity, we applied thresholding on the same index to be computed. However, it is possible to threshold the image based on one index and then compute one, or more, other indices on the retained pixels in a single pass. Moreover, it is also possible to specify more complex thresholding, e.g., banding, such as $(\text{GLI} > 0) \wedge (\text{GLI} < 0.8)$. In all cases, the evaluation of different index values and the selection of the best threshold is not automated in DRONE2REPORT.

Table 1. An example of reporting table produced by DRONE2REPORT: green leaf index (GLI) values with different thresholding, as computed by [TASK index]. For each Region of Interest (as defined in the shapefile) a set of summary statistics is computed for each index (mean, standard deviation, min and max). after_th: n. of pixels left for the calculation after thresholding. Note that with these settings, the last column (GLI_min) strictly reflects the progression of thresholding.

Threshold	Pixels	After_th	GLI_Mean	GLI_Med	GLI_Std	GLI_Max	GLI_Min
none	254,493	254,493	0.26837	0.25397	0.14712	0.97753	−0.36842
(GLI > 0.0)	254,493	254,251	0.26,866	0.25424	0.14690	0.97753	0.00000
(GLI > 0.1)	254,493	222,159	0.29813	0.28090	0.13327	0.97753	0.10000
(GLI > 0.2)	254,493	160,082	0.35525	0.33333	0.11271	0.97753	0.20000
(GLI > 0.3)	254,493	99,914	0.41901	0.39706	0.09542	0.97753	0.30000
(GLI > 0.4)	254,493	48,731	0.49456	0.47200	0.08247	0.97753	0.40000
(GLI > 0.5)	254,493	18,278	0.58052	0.55932	0.07340	0.97753	0.50000
(GLI > 0.6)	254,493	5799	0.67194	0.65289	0.06573	0.97753	0.60000
(GLI > 0.7)	254,493	1860	0.76379	0.74713	0.05775	0.97753	0.70000
(GLI > 0.8)	254,493	798	0.85531	0.84240	0.04428	0.97753	0.80000
(GLI > 0.9)	254,493	558	0.93064	0.92203	0.02319	0.97753	0.90164

3.2. Case Study N. 2: Monitoring Vegetation Indices over Time

In this case study, we use DRONE2REPORT to monitor the evolution of vegetation indices over time. This is the case when several images of the same field/area are collected at successive time intervals, e.g., for a phenotyping experiment in plant breeding. Vegetation indices over time may be useful to monitor plant growth, soil coverage, maturation, etc. over a period of time of interest, like a growing season.

In this illustration, drone-captured barley-field images from the same experiment as in case study n. 1 were used; within this field, a different barley variety is sown in each plot. However, instead of using one single-plot image (one single barley variety) at one single timepoint, the images selected for this case study cover 20 plots (20 barley varieties) over 10 flights. Flights were performed in Fiorenzuola d'Arda (Italy, $44°55'40.2466''$, $009°53'42.4730''$) between 19 February 2024 (first flight on an empty field) and 12 June 2024 (tenth flight). As vegetation indices, we use GLI (described before, see Equation (1)) and HUE [13], defined as:

$$\text{HUE}(\mathbf{p}) = \arctan\left(\frac{2 \cdot p_{red} - p_{green} - p_{blue}}{30.5} \cdot (p_{green} - p_{blue})\right) \quad (2)$$

where $\mathbf{p} := [p_{red}, p_{green}, p_{blue}]$ is a single pixel of an RGB image and contains the normalized intensities of the three channels. HUE too can be used to discriminate between vegetation and non-vegetation, e.g., soil vs plant, although it is more dependent on lighting, camera and background [14].

To calculate vegetation indices over multiple images (i.e., successive flights) using DRONE2REPORT, we can follow two approaches: (i) we can add several data sections in the configuration (.ini) file, one for each image, then run DRONE2REPORT just once; or (ii) iterate over the number of images, automatically generating each .ini file, thus running DRONE2REPORT as many times as there are images.

Once the chosen vegetation indices are computed for all multiple images, it is possible to plot their trends over time to monitor the evolution of the field. In the barley example, each plot represents a different variety (with three repetitions due to experimental design). It is thus possible to make inter-varietal comparisons. This is shown in Figure 5B, where we highlighted two arbitrary varieties (in red and blue). The barley plots start with moderate green intensity (sparse vegetation), reach a maximum around April to early May (thick vegetation), then decrease toward the end of the growing season (matured yellowish barley in June). This trend matches what we observed in the RGB images (Figure 5A). The two indexes follow a similar trend, with a peak and a decrease. Interestingly, GLI peaks earlier than HUE, which in turn shows a sharper peak around May. Variability exists between barley varieties, probably linked to their different genetic make-up.

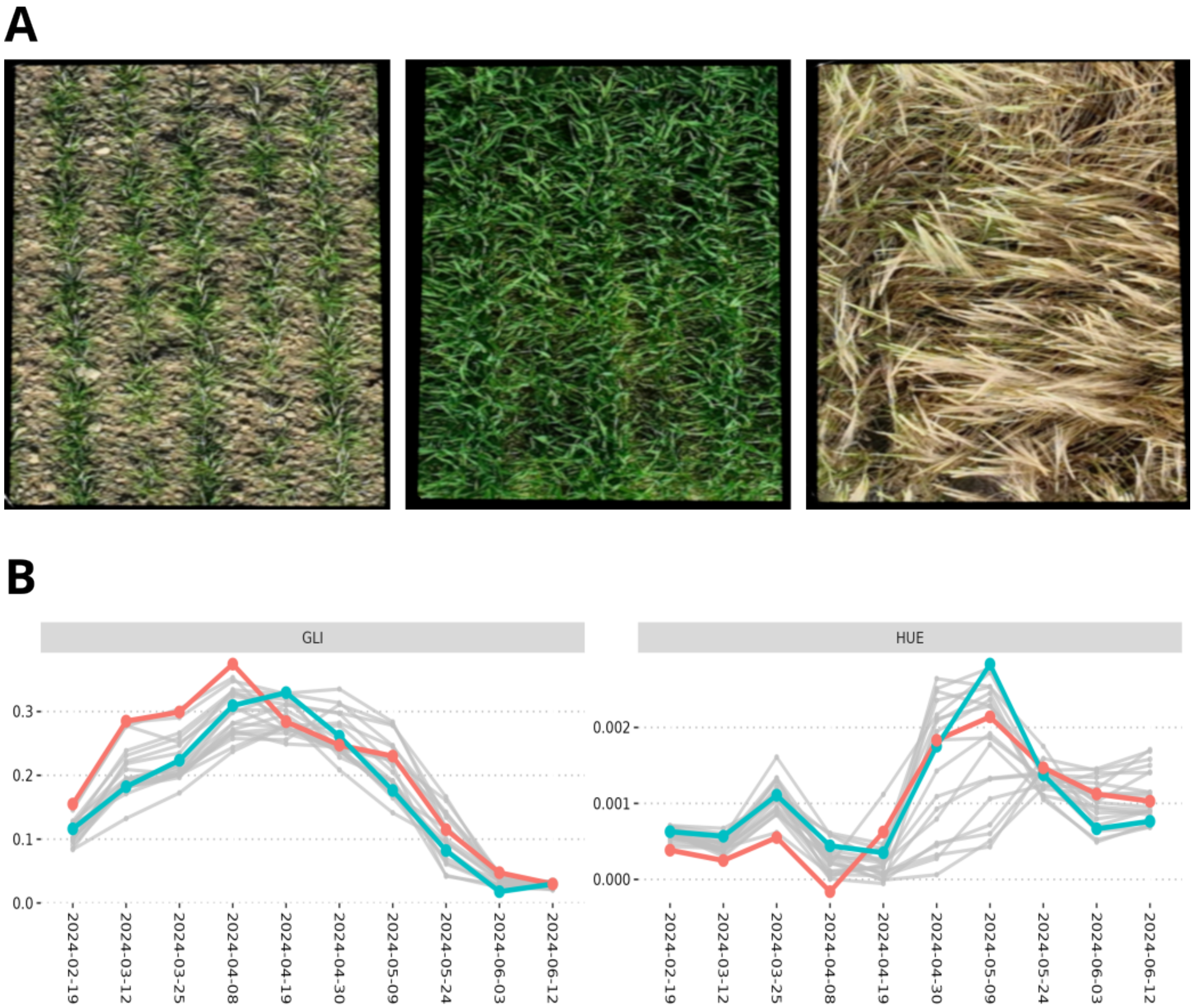

Figure 5. Vegetation indices over time. (**A**) The same barley plot drone-photographed in February, April, and June. (**B**) Evolution of GLI index (**left**) and HUE index (**right**) values for twenty barley varieties over the entire growing season (10 flights in February–June 2024). Two arbitrarily selected varieties are highlighted (red and blue).

3.3. Case Study N. 3: Analysis of Height from DEM Files

As mentioned before, drone phenotyping also produces a DEM file alongside the orthomosaic image. DEM files report the height of each pixel in the orthomosaic image: this height is usually reported in meters, and is measured from a reference ellipsoid. Given that the first flight is usually performed on an empty field, it can be used as a reference. By subtracting the heights of the first flight from all successive flights, it is possible to estimate plant growth.

In Figure 6A, a hill-shade rendering of a DEM file for the example barley field is reported (part of it), to visualize the 3D structure of the field image. Pixel heights will differ depending on the content, whether soil, an erect plant, or a lodged one. The evolution of per-plot average heights over time is reported in Figure 6B for the successive flights. Note that the very first flight is used as reference and thus all pixels are zero by definition.

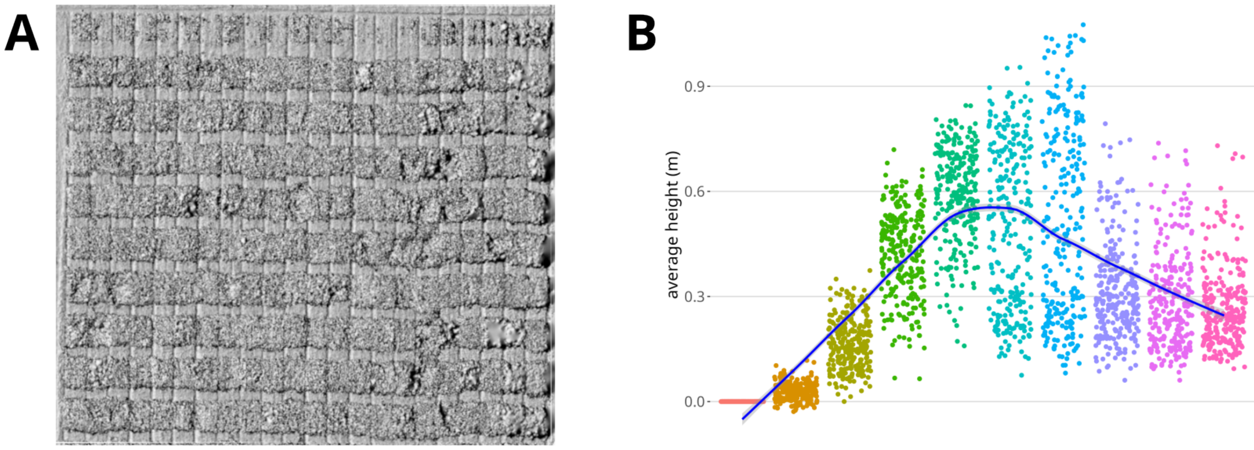

Figure 6. Example .dem file of a barley field (**A**): flight on 19 April 2024); plant height (meters) calculated from the .dem file by DRONE2REPORT, over 10 flights covering the growing season; (**B**) the dates for the 10 flights are the same as in Figure 5.

After the first flight, the heights quickly increased up to the 5th and 6th flights (mid-April: the plants were growing). In our barley field trials, DRONE2REPORT shows a decrease in height after the 7th flight due, in this case, to lodging, consistently with the genetic material examined (old cultivars and landraces) and the adverse weather conditions. This can also be visually confirmed by examining the right-most part of Figure 5A, where a large portion of barley plants appears to have been flattened. In general, similar height-decrease patterns may be linked to a variety of causes, e.g., they may also indicate the moment when the spike begins to fall over (barley plants are known to “hide” the spike in the vegetation). Here, we knew there was lodging; however, to be able to detect lodging (and where it occurred in the field), more complex machine-learning models for image segmentation and classification should be implemented, as shown in Section 3.5.

From the DEM files, the per-plot volumes can also be estimated using DRONE2REPORT. Consider that the integral of an area is proportional to the sum of the heights of each pixel. When computing indices like GLI or HUE, the actual result of the computation is a matrix with exactly the same dimensions as the ROI (i.e., the plot). It is, however, possible to select “indexes” that return a single scalar number, e.g., the summation of all values for a plot, or other custom algebraic operations. For example, adding all height values of a plot (taken from the DEM file) allows us to compute a finite approximation of the Riemann integral of the above-ground volume, which in turn can be used to estimate crop yield or biomass.

3.4. Case Study N. 4: Index Optimization on Images Merged from Heterogeneous Sensors

DRONE2REPORT can transparently import images from different sensors and merge them into a single data structure where all channels of the original images are stacked. For example, it is possible to merge images coming from visible wavelengths (e.g., RGB images), from sensors in the non-visible wavelength (e.g., infrared and near-infrared images) and from thermal cameras. Differences in image resolutions are managed automatically via cubic interpolation. Differences in the spatial reference system (SRS) are handled via reprojection of all data into the reference taken from the first input orthomosaic. ROIs are taken from the single shape file (common to all orthomosaics).

The resulting data structure exposes, for each pixel, a number of channels equal to the sum of all channels of the original images. Thus, it is possible to leverage different data sources together and transparently.

In this case study, three images are merged: one RGB (three bands; camera described in Case Study n. 1), one multispectral (ten bands; camera: DJI Matrice 300 RTK with Micasense Dual Camera System sensor; resolution: 0.83 cm/pixel; overlap: 80%; sidelap: 80%) and one thermal (one band; drone: DJI Mavic 3 Enterprise Thermal with integrated camera; resolution: 1.57 cm/pixel; overlap: 68%; sidelap: 90%). All images are of a barley field close to harvesting time. The field is structured in 264 plots, each phenotyped for heading date, i.e., the number of days from sowing when the spike (head) has emerged from the flag leaf sheath in 50% of the plants in a plot. The aim of of this case study is to optimize a new vegetation index of the form:

$$I(\mathbf{p}) = \frac{n_0 + \sum_{i=1}^{14} n_i p_i}{d_0 + \sum_{i=1}^{14} d_i p_i} \tag{3}$$

where $\mathbf{p} \in \mathbb{R}^{14}$ is a pixel on the merged 14-bands image. The set of 30 coefficients $[n_0, \ldots, n_{14}, d_0, \ldots, d_{14}]$ is to be chosen to maximize Pearson's correlation between the array of per-plot average index values and the target variable. Formula 3 is designed as a generalization of the most commonly used indices. In fact, indices such as NDVI, GLI and VARI [15] can all be derived as special combinations of the d_i, n_i coefficients.

This thus becomes a 30-parameter optimization problem, which can be solved via a new custom DRONE2REPORT task. Internally, the task uses `Scipy`'s [16] *optimize.minimize()* function based on the *Nelder-Mead* numerical method [17].

The dataset, consisting of 264 plots and their corresponding heading dates, was split into the training set (80% of the data) and validation set (20% of the data). The results of the optimization process are reported in Figure 7, which shows the optimization trajectory of the custom index in Equation (3) through all training iterations. Additionally, the plot also reports Pearson's correlation with the heading date for the two best-performing vegetation indices among the twenty-one currently implemented in DRONE2REPORT, namely NDVI (Normalized Difference Vegetation Index, for multispectral images) and GLI (Green Leaf Index, for RGB images).

Notice how the correlation on the validation set trails below the values computed on the training set. At some point, just before iteration 250, the correlation on the validation set drops, while on the training set continues to rise. This is expected, and it is a clear symptom of overfitting. This can easily be contrasted by a number of techniques, e.g., Early Stopping [18], that can intercept the sudden drop on the validation set and stop the training process. Alternatively, the set of parameters maximizing correlation on the validation set can be extracted from the training log. In this second case, the training will continue to run until the optimizer's convergence criteria are met.

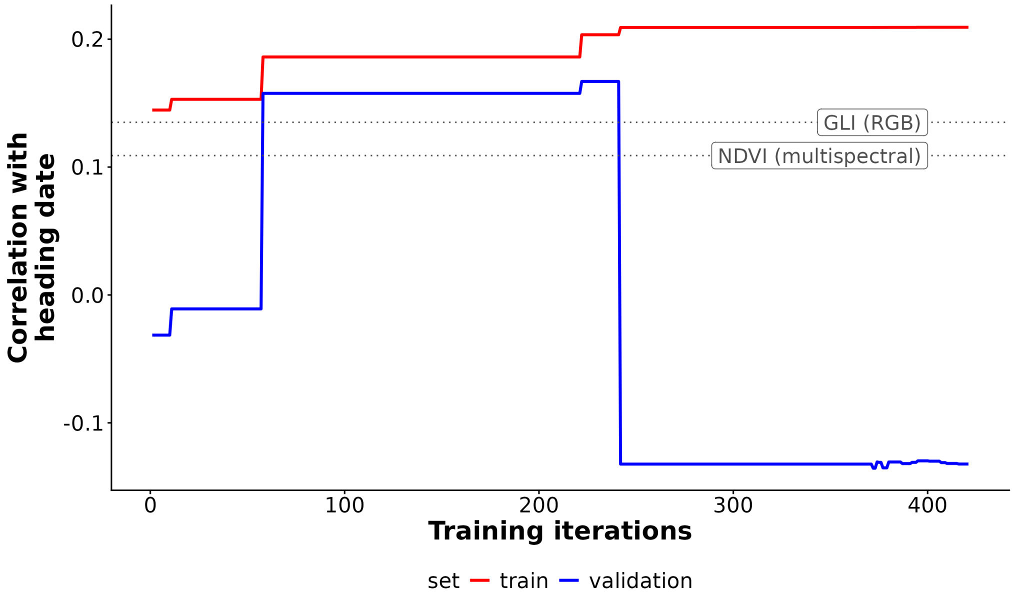

Figure 7. Optimization trajectories through training cycles. The custom index described in Formula (3) was optimized for maximal Pearson's correlation between the vector of its per-plot averages and the phenotype *heading date* (here acting as target variable). The color indicates if the values refer to the correlation on the training or validation set. For reference, the two best performing indices (NDVI and GLI) are also reported.

3.5. Case Study N. 5: Deep Learning for Image Classification

DRONE2REPORT can be used to feed georeferenced data into a trained deep-learning model and to collect inference results with minimal user-provided code. For this case study, we used UAV imagery from a publicly available dataset developed for weed detection in soybean [19]. This is a four-class (soybean, broadleaf weed, grass, bare soil) classification problem, comprising a 15,926-image dataset.

As a preliminary step, we trained a deep neural network based on the well-known ResNet50 architecture [20], as implemented in the TensorFlow/Keras framework [21]. The complete training procedure, including the modifications applied to the base ResNet50 architecture and the strategies adopted for early stopping and learning-rate scheduling, is reported in a supplementary notebook to ensure reproducibility. Out of all available images, forty images (ten per class) were retained and completely isolated from the training process to form an independent test set. The remaining images were split between training (70%, 11,120 images) and validation (30%, 4766 images) subsets. The corresponding training trajectories are reported in Figure 8.

Figure 8 shows that the training saturates after 10–12 training epochs, and shows no sign of overfitting (loss on training set and validation set are very similar).

Once training was complete, the trained network object (comprising both the model architecture and the optimized weights) was retained. To enable DRONE2REPORT to leverage the trained network, a custom [TASK] script was implemented to (1) load the trained model, (2) preprocess the raster region-of-interest (ROI) data to match the network input requirements, (3) execute the network in inference mode, and (4) collect the prediction results into an output file. The custom task comprises approximately eighty lines of code, including inline comments and informational messages to guide the user through the execution steps. The network was run on the test set (forty images, ten per class, not involved during the network training), on which it achieved perfect classification accuracy. This simulates a real-world application where an externally trained model is applied on incoming data to guide management decisions.

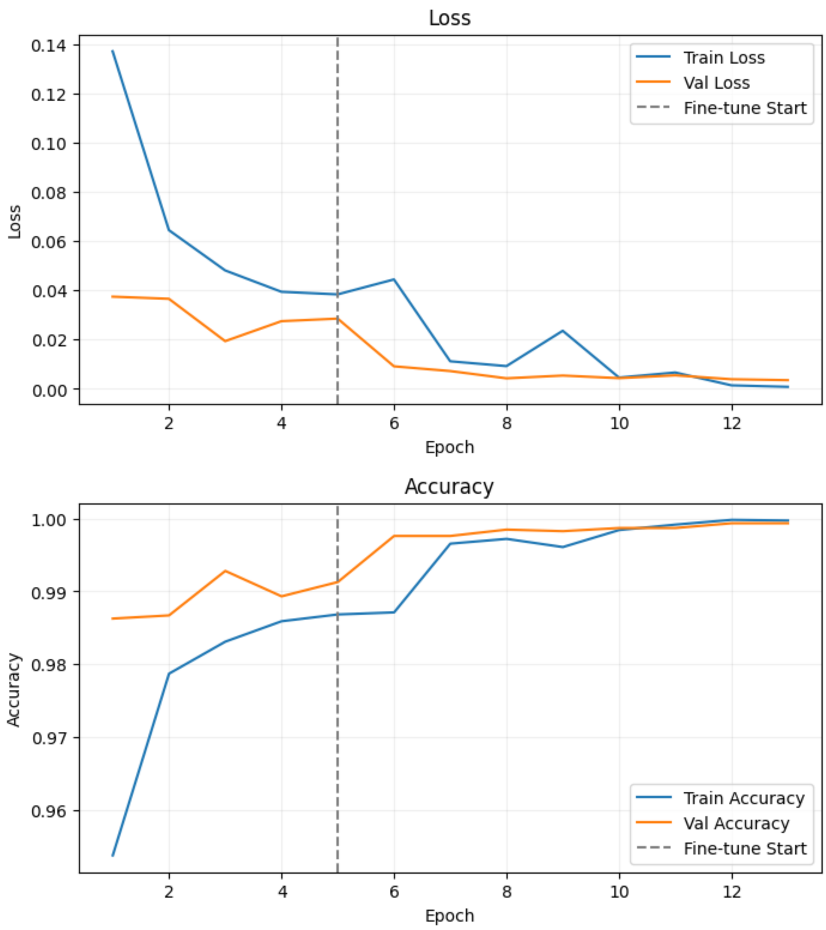

Figure 8. Optimization trajectories through training cycles. The (**top**) panel reports the value of the loss function (computed as categorical cross-entropy) for the train and validation sets through the training epochs. The (**bottom**) panel reports the accuracy, computed as the fraction of correct predictions over the total number of samples.

3.6. Rendering and Reporting

After all [TASK]s are executed, DRONE2REPORT executes all the active [RENDER] sections. At the time of writing, two types of render are implemented. The first one is a small utility function to collect and collate all tables produced by index computation into a single .csv file. This is useful if several runs with different configurations have been executed, so as to have a single collection point. The second render functionality outputs an .html report with all the produced logs and, optionally, the computed indices plotted in a scatter plot using the ROI centroids for coordinates and the computed index for color intensity. An example of such a report is available as Supplementary File S1.

4. Discussion

In this section, we first compare DRONE2REPORT to the state-of-the-art free software tools for drone-image processing in agriculture. Then, we discuss the range of applications of DRONE2REPORT, its strengths and limitations, and the future directions for further developments.

4.1. State-of-the-Art Free Software Solutions, and Positioning of DRONE2REPORT

UAV imagery has become central to high-throughput field phenotyping, with numerous reviews documenting rapid adoption across RGB, multispectral and thermal sensing for crop trait estimation and stress monitoring [22,23]. These studies consistently highlight persistent bottlenecks not in image acquisition, but in the standardization and analysis of downstream workflows [23].

Several tools and software packages are available for processing and analyzing UAV-acquired image data, including commercial products [24]. In this section, we briefly survey the most common open-source and/or freeware tools that are currently available for photogrammetric data processing, and describe where DRONE2REPORT stands in this landscape. We provide a summary table as Supplementary Table S1. The majority of open-source/freeware software packages for UAV-imagery processing is written in `Python` and based on libraries for image manipulation like `scikit-image`, `PIL/Pillow`, `GDAL`, `geopandas`.

OpenDroneMap (ODM/WebODM: https://github.com/OpenDroneMap/WebODM (accessed on 13 April 2026) [25]) is a mature, subscription-based software stack that excels at turning raw images into geospatial products —orthomosaics, point clouds, 3D meshes and DEMs— and even provides "plant health" visualizations (e.g., NDVI/VARI) within the viewer. The focus of ODM/WebODM is reconstruction; however, it can well be used for downstream plot-level analytics, multi-sensor alignment, and standardized per-ROI statistics/reporting.

Orfeo ToolBox (OTB: https://gitlab.orfeo-toolbox.org/ (accessed on 13 April 2026) [26,27]) is a general-purpose remote sensing library that offers state-of-the-art algorithms via command-line apps, Python bindings and QGIS integration, making it highly flexible for classification, segmentation and raster math. Yet OTB is intentionally low-level: users compose pipelines themselves and must implement domain-specific phenotyping logic (e.g., plot masks, trial layouts, multi-sensor co-registration, batch reporting).

QGIS (https://qgis.org/ (accessed on 13 April 2026) [2]) is a free and open-source desktop geographic information system (GIS) suite. It allows users to create, edit, visualize, analyze, and publish geospatial information. In particular, QGIS provides a Processing Modeler [28] for chaining algorithms and running them in batch, which can partially address reproducibility. In practice, model portability, parameter/version management, and experiment-level reporting remain challenging, especially across groups and computer environments.

PlantCV (https://plantcv.org/ (accessed on 13 April 2026) [29]) is a widely used, open-source platform for modular image analysis in plant science, particularly effective for close-range/greenhouse imagery and single-image pipelines. It is commonly run using Jupyter notebooks. It is not primarily oriented to georeferenced orthomosaics, or plot-level analysis.

GRABSEEDS (https://github.com/tanghaibao/jcvi/wiki/GRABSEEDS (accessed on 13 April 2026) [3]) is a command-line `Python` software tool specifically engineered for the identification and phenotyping of plant seeds, leaves and flowers. It performs edge detection, object (seed) identification, image cropping and text label recognition across a wide range of grains and legumes. However, *GRABSEEDS* only accepts input RGB image files, and is not designed to work with UAV-captured field image data.

BerryPortraits (https://github.com/Breeding-Insight/BerryPortraits/ (accessed on 13 April 2026) [4]) is an open-source `Python` image-analysis software based on the YOLOv8 model for computer vision [30]. This is a command-line tool (no GUI) that detects and segments berries and extracts morphometric data on quality traits like berry color, size, shape, and uniformity. As other highly specialized tools, *Berryportraits* only works with RGB images and not with UAV aerial photogrammetry.

FIELDimageR (https://github.com/OpenDroneMap/FIELDimageR (accessed on 13 April 2026) [31]) is one of the very few software packages for UAV-imagery processing written in `R`. It targets orthomosaics from field trials and provides functions for plot extraction and vegetation indices, but is embedded in the `R` ecosystem and typically used through scripts/notebooks rather than a configuration-driven, sensor-agnostic CLI (command-line interface) designed for end-to-end, repeatable reports.

GRID: *GReenfield Image Decoder* (https://poissonfish.github.io/GRID/index.html (accessed on 13 April 2026) [32]) is a Python package that processes aerial imagery data and is mainly geared towards image segmentation. In this respect, GRID implements pixel-wise K-means cluster analysis to identify pixels of interest (POI) and areas of interest (AOI). GRID provides a GUI (graphical user interface).

PREPs: *Precision Plots Analyzer for breeding field* [33]) is an open-source software for the analysis of UAV images, developed in `C#` within the .NET Microsoft framework. PREPs offers a GUI and allows for image segmentation and for the estimation of crop height, coverage, and volume index. PREPs is only available for the Windows OS.

IHUP: *Integrated High-Throughput Universal Phenotyping* (https://drive.google.com/uc?export=download&id=1aZalN0yqli9l2pqyQAPK0IiACs7UhrHF (accessed on 13 April 2026) [34]) is a software platform for the analysis of aerial images, which was developed in `C#` using the Visual Studio IDE (Integrated Development Environment). IHUP provides a GUI and allows for the calculation of customisable vegetation indices. IHUP is only available for the Windows OS.

DRONE2REPORT was developed specifically to streamline the processing of orthorectified UAV-captured images for the calculation of vegetation indices and the reporting of relevant summary statistics. The tool is designed to run after image acquisition, photogrammetry and, if needed, calibration. It is designed to sit after the orthomosaic generation (e.g., from ODM/WebODM or commercial tools) and provides a configuration-driven, reproducible workflow that takes in input one or more multiband images (being these RGB, multispectral, thermal images and DEM files). Its core functionalities include: (i) identifying regions of interests (ROIs, based on the shape file); (ii) calculating vegetation indices; (iii) thresholding; (iv) summarizing statistics; (v) generating thumbnails.

The software is open-source, easily customizable, and designed for batch processing via a single configuration file (in `.ini` format). It runs entirely via the command line (no GUI dependency), ensuring platform independence and reproducibility across labs and servers. Dependencies are managed through a Conda environment, simplifying installation and portability.

A distinctive feature of DRONE2REPORT is its support for multi-sensor alignment and combined index generation, with built-in resampling and spatial reference system (SRS) management. This enables new analyses that leverage data from multiple sources—an emerging need in precision agriculture. Moreover, the tool emphasizes robust validation and reporting, with support for dry runs and standardized outputs per ROI, ready for integration into statistical workflows.

In short, while *ODM/WebODM* specializes in producing sophisticated digital outputs from raw UAV-imagery data (but can go well beyond that), *OTB* and *QGIS* provide low-level building blocks, *GRABSEEDS* and *BerryPortraits* occupy specific application niches, *GRID* and *PREPs* offer flexible segmentation functionalities, and *PlantCV* and *FIELDimageR* target image-centric or R-centric workflows, DRONE2REPORT contributes a domain-specific, geospatial, configuration-driven bridge from orthomosaics to plot-level, multi-sensor, repeatable reports tailored to field phenotyping. Moreover, DRONE2REPORT can easily integrate heterogeneous predictive models, such as deep neural networks, with minimal code adaptation.

4.2. Applications, Limitations and Future Developments

The main goal of DRONE2REPORT is to automate the calculation of vegetation indices from multi-channel UAV imagery—a key task in modern precision agriculture (e.g., [35–37]). We showed several case studies on how vegetation indices can be obtained and used for a range of applications, including the removal of background noise via thresholding; the comparison of different fields and/or plots; time-series analysis, lodging detection, and the development of novel indices through the optimization of multiple mixed channels for a target metric.

Moreover, DRONE2REPORT can be easily integrated with existing computer vision models, alleviating the burden of data management and enabling the construction of a streamlined processing pipeline.

The tool is easy to use from the command line by simply modifying the configuration file (several different templates are provided in the online documentation). It is also easily portable, on any OS or platforms where `Python` is available; portability is enhanced by the distribution of the software via a conda environment, which seamlessly handles all required dependencies. Additionally, DRONE2REPORT offers ample extendibility, allowing users to define new indices or analysis tasks, as shown in the section on software implementation. We also showed how signals from multiple sensors (RGB, multispectral, thermal, DEM) can be integrated to develop custom indices, a feature largely unexplored in existing tools.

There are, of course, a few limitations that need to be pointed out. First of all, there is no graphical interface (GUI), which may pose a barrier for users unfamiliar with command-line tools. This, however, was a design choice, since the tool was developed as part of a larger, server-based pipeline.

Second, in the current implementation DRONE2REPORT does not expose any radiometric calibration capability, which can be very important to compare vegetation indices across different flights.

Finally, the current reporting capabilities are limited to collecting the execution logs plus some basic rendering of computed indexes. Further developments will be required to implement data exploration, summaries, and synthetic representations of the implemented analyses.

5. Conclusions

DRONE2REPORT is a new software tool designed to support repeatable, scalable, and multi-sensor UAV data analysis in agricultural field trials. It is designed to bridge the gap between orthomosaic creation and actual agriculture applications, a gap typically filled with custom, non-sharable software solutions. Moreover, DRONE2REPORT is designed around reproductibility and modularity, and it is easily expandable to integrate external models. Its design makes it a valuable asset not only for researchers in precision agriculture, but also for broader applications such as forestry, land-use monitoring, and environmental surveys.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/drones10040301/s1, Supplementary Table S1: summary of the main freely available software packages for the processing of image data in agriculture. Supplementary File S1: example of .html report. Note that, by design, the value of the two computed indices (GLI and HUE) are valid on the RGB image and invalid on the thermal one (which does not have the necessary channels). This is done to show what happens when invalid indices are requested.

Author Contributions: N.N. implemented the core functionalities of the software. N.N., G.M. and F.B. implemented the case studies. A.F., E.G., P.D.F., N.F. and G.C. curated the data for the case studies. G.M., E.F., F.B., N.F. and R.P. tested the software. N.N. and F.B. wrote the first draft of the article. All authors have read and agreed to the published version of the manuscript.

Funding: Funded by the European Union-Next Generation EU, Mission 4, Component 1, CUP B53D23007940006: PRIN 2022 project "Polyploidbreeding 4.0" (G.A.: 2022BACN8A). Also funded by CREA, “stage di mobilità breve all’estero”, register number 0107199, 24 November 2023. Also funded by Research and Development of Plant Genetic Resources for Sustainable Agriculture, Center of Excellence for Biodiversity and Molecular Plant Breeding (CoE CroP-Bio-Div), Zagreb, Croatia (PK.1.1.10.0008).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: DRONE2REPORT is freely available at https://github.com/ne1s0n/drone2report (accessed on 13 April 2026). The code and data to reproduce the case studies for this paper are available at https://github.com/ne1s0n/paper-drone2report (accessed on 13 April 2026).

Acknowledgments: We sincerely thank the Module 7 tean at the Netherlands Plant Eco-phenotyping Centre (NPEC) at Wageningen University & Research (NL) for the collaboration that sprouted this software.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Machwitz, M.; Pieruschka, R.; Berger, K.; Schlerf, M.; Aasen, H.; Fahrner, S.; Jiménez-Berni, J.; Baret, F.; Rascher, U. Bridging the gap between remote sensing and plant phenotyping—Challenges and opportunities for the next generation of sustainable agriculture. *Front. Plant Sci.* **2021**, *12*, 749374. [CrossRef]
2. Moyroud, N.; Portet, F. Introduction to qgis. *Qgis Generic Tools* **2018**, *1*, 1–17.
3. Tang, H.; Kong, W.; Nabukalu, P.; Lomas, J.S.; Moser, M.; Zhang, J.; Jiang, M.; Zhang, X.; Paterson, A.H.; Yim, W.C. Grabseeds: Extraction of plant organ traits through image analysis. *Plant Methods* **2024**, *20*, 140. [CrossRef]
4. Loarca, J.; Wiesner-Hanks, T.; Lopez-Moreno, H.; Maule, A.F.; Liou, M.; Torres-Meraz, M.A.; Diaz-Garcia, L.; Johnson-Cicalese, J.; Neyhart, J.; Polashock, J.; et al. Berryportraits: Phenotyping of ripening traits cranberry (vaccinium macrocarpon ait.) with yolov8. *Plant Methods* **2024**, *20*, 172. [CrossRef]
5. Park, J.; Cho, Y.; Kim, S. Deep learning-based UAV image segmentation and inpainting for generating vehicle-free orthomosaic. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *115*, 103111. [CrossRef]
6. Welcome to OpenDroneMap’s Documentation—OpenDroneMap 3.5.6 Documentation. Available online: https://docs.opendronemap.org/ (accessed on 27 August 2025).
7. Rupnik, E.; Daakir, M.; Pierrot Deseilligny, M. Micmac—A free, open-source solution for photogrammetry. *Open Geospat. Data Softw. Stand.* **2017**, *2*, 14. [CrossRef]
8. Schönberger, J.L.; Frahm, J.M. Structure-from-Motion Revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2016.
9. GDAL/OGR Contributors. *GDAL/OGR Geospatial Data Abstraction Software Library*; Open Source Geospatial Foundation: Beaverton, OR, USA, 2025. [CrossRef]
10. Jordahl, K.; Bossche, J.V.; Fleischmann, M.; Wasserman, J.; McBride, J.; Gerard, J.; Tratner, J.; Perry, M.; Badaracco, A.G.; Farmer, C.; et al. *Geopandas/Geopandas*, V0.8.1; 2020. Available online: https://doi.org/10.5281/zenodo.3946761 (accessed on 13 April 2026). [CrossRef]
11. Louhaichi, M.; Borman, M.M.; Johnson, D.E. Spatially located platform and aerial photography for documentation of grazing impacts on wheat. *Geocarto Int.* **2001**, *16*, 65–70. [CrossRef]
12. Hamuda, E.; Glavin, M.; Jones, E. A survey of image processing techniques for plant extraction and segmentation in the field. *Comput. Electron. Agric.* **2016**, *125*, 184–199. [CrossRef]
13. Mathieu, R.; Pouget, M.; Cervelle, B.; Escadafal, R. Relationships between satellite-based radiometric indices simulated using laboratory reflectance data and typic soil color of an arid environment. *Remote Sens. Environ.* **1998**, *66*, 17–28. [CrossRef]
14. Meyer, G.E.; Neto, J.C. Verification of color vegetation indices for automated crop imaging applications. *Comput. Electron. Agric.* **2008**, *63*, 282–293. [CrossRef]
15. Henrich, V.; Krauss, G.; Götze, C.; Sandow, C. *Idb—www.indexdatabase.de, Entwicklung Einer Datenbank für Fernerkundungsindizes*; AK Fernerkundung: Bochum, Germany, 2012; Volume 15.
16. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
17. Nelder, J.A.; Mead, R. A simplex method for function minimization. *Comput. J.* **1965**, *7*, 308–313. [CrossRef]
18. Yao, Y.; Rosasco, L.; Caponnetto, A. On early stopping in gradient descent learning. *Constr. Approx.* **2007**, *26*, 289–315. [CrossRef]

19. Peccia, F. Weed Detection in Soybean Crops. 2018. Available online: https://www.kaggle.com/datasets/fpeccia/weed-detection-in-soybean-crops (accessed on 15 January 2026).
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778.
21. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 13 April 2026).
22. Tanaka, T.S.T.; Wang, S.; Jørgensen, J.R.; Gentili, M.; Vidal, A.Z.; Mortensen, A.K.; Acharya, B.S.; Beck, B.D.; Gislum, R. Review of Crop Phenotyping in Field Plot Experiments Using UAV-Mounted Sensors and Algorithms. *Drones* **2024**, *8*, 212. [CrossRef]
23. Gano, B.; Bhadra, S.; Vilbig, J.M.; Ahmed, N.; Sagan, V.; Shakoor, N. Drone-based imaging sensors, techniques, and applications in plant phenotyping for crop breeding: A comprehensive review. *Plant Phenome J.* **2024**, *7*, 20100. [CrossRef]
24. Sharma, M.; Raghavendra, S.; Agrawal, S. Development of an open-source tool for uav photogrammetric data processing. *J. Indian Soc. Remote Sens.* **2021**, *49*, 659–664. [CrossRef]
25. Vacca, G. Web open drone map (webodm) a software open source to photogrammetry process. In Proceedings of the Fig Working Week 2020. Smart Surveyors for Land and Water Management, Amsterdam, The Netherlands, 10–14 May 2020.
26. Orfeo ToolBox—Orfeo ToolBox is not a Black Box. Available online: https://www.orfeo-toolbox.org/ (accessed on 27 August 2025).
27. Grizonnet, M.; Michel, J.; Poughon, V.; Inglada, J.; Savinaud, M.; Cresson, R. Orfeo ToolBox: Open source processing of remote sensing images. *Open Geospat. Data, Softw. Stand.* **2017**, *2*, 15. [CrossRef]
28. Dobesova, Z. Evaluation of effective cognition for the qgis processing modeler. *Appl. Sci.* **2020**, *10*, 1446. [CrossRef]
29. PlantCV 2021. Available online: https://plantcv.org (accessed on 27 August 2025).
30. Varghese, R.; Sambath, M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
31. Matias, F.I.; Caraza-Harter, M.V.; Endelman, J.B. FIELDimageR: An R package to analyze orthomosaic images from agricultural field trials. *Plant Phenome J.* **2020**, *3*, 20005. [CrossRef]
32. Chen, C.J.; Zhang, Z. Grid: A python package for field plot phenotyping using aerial images. *Remote Sens.* **2020**, *12*, 1697. [CrossRef]
33. Itoh, A.; Njane, S.N.; Hirafuji, M.; Guo, W. Preps: An open-source software for high-throughput field plant phenotyping. *Plant Phenomics* **2024**, *6*, 0221.
34. Wang, B.; Yang, C.; Zhang, J.; You, Y.; Wang, H.; Yang, W. Ihup: An integrated high-throughput universal phenotyping software platform to accelerate unmanned-aerial-vehicle-based field plant phenotypic data extraction and analysis. *Plant Phenomics* **2024**, *6*, 164.
35. Candiago, S.; Remondino, F.; De Giglio, M.; Dubbini, M.; Gattelli, M. Evaluating multispectral images and vegetation indices for precision farming applications from uav images. *Remote Sens.* **2015**, *7*, 4026–4047. [CrossRef]
36. Radočaj, D.; Šiljeg, A.; Marinović, R.; Jurišić, M. State of major vegetation indices in precision agriculture studies indexed in web of science: A review. *Agriculture* **2023**, *13*, 707. [CrossRef]
37. Vélez, S.; Martínez-Peña, R.; Castrillo, D. Beyond vegetation: A review unveiling additional insights into agriculture and forestry through the application of vegetation indices. *J* **2023**, *6*, 421–436. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.