


DRONE2REPORT: A Configuration-Driven Multi-Sensor Pipeline for UAV-Based Crop Phenotyping and 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, Nicolo 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, Lodi, 26900, Italy; nicolo.franguelli@crea.gov.it
- ² Institute of Biology and Biotechnology in Agriculture (IBBA), National Research Council (CNR), Via A. Corti, 12, Milan, 20133, Italy; giulia.moscatelli@ibba.cnr.it; filippo.biscarini@cnr.it
- ³ Research Centre for Genomics and Bioinformatics, Council for Agricultural Research and Analysis of Agricultural Economics, Via San Protaso 69, Fiorenzuola d'Arda, 29017, Italy; giulia.moscatelli@ibba.cnr.it; giorgia.carletti@crea.gov.it
- ⁴ Department of Agricultural and Food Sciences, University of Bologna, Via G. Fanin, 44, Bologna, 40127, Italy; elisabetta.frascaroli@unibo.it; roshan.paudel@unibo.it
- ⁵ Research & Development, SIS - Società Italiana Sementi, Via Mirandola, 5, San Lazzaro di Savena (BO), 40068, Italy; e.groli@sisonweb.com; p.defranceschi@sisonweb.com
- * 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 library, DRONE2REPORT, is released as a standardized and customizable open-source framework for processing 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

Background: Unmanned aerial vehicles (UAVs) have become indispensable tools in precision agriculture and plant phenotyping, enabling 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 orthophotos 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. **Results:** We present DRONE2REPORT, an open-source python-based software that processes orthophotos/orthomosaics from UAV flights to generate vegetation indices, summary statistics and derived subimages, supporting both research and applied breeding contexts. Alongside the basic structure and functioning of DRONE2REPORT, we present also five case studies that illustrate practical applications common in UAV/drone-phenotyping of plants: i) thresholding to remove background noise and highlight regions of interest; ii) monitoring



Received:

Revised:

Accepted:

Published:

Copyright: © 2025 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

plant phenotypes over time; iii) extracting information on plant height to detect events like lodging; iv) integrating multiple sensors (cameras) to construct and optimise 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. **Conclusions:** DRONE2REPORT streamlines UAV image processing for precision agriculture by linking orthophoto data 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; 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 them, 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 [2].

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 (e.g. [3]), have sometimes very specific targets (e.g. seeds [4], berries [5]), 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. DRONE2REPORT was developed to address this gap. It is designed specifically for agriculture (e.g. a field with multiple plots each accommodating one genotype) and implements a configuration-driven pipeline that streamlines the transformation of UAV imagery 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. ~~The software is released under open-source license and readily available at.~~ In this paper, we

present the technical details of this new tool, followed by four 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. We adopt here the following definitions: orthophotos (or orthorectified images) are single drone-captured images that have been geometrically corrected (perspective, camera tilt etc.), while orthomosaics are collection of orthophotos stitched together (taking care of overlaps etc.) [1]. 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 what is called an orthophoto (or orthomosaic, or orthorectified image) 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 QGIS [3] Agisoft Metashape version 1.6.4 (Agisoft LLC, St. Petersburg, Russia), for orthorectification. Together with the orthorectified 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.

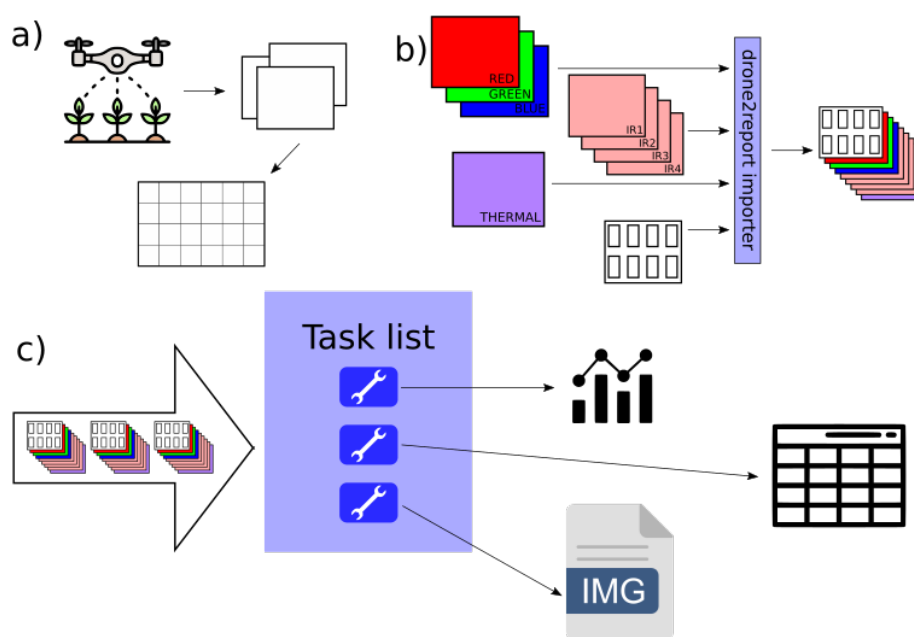


Figure 1. General application workflow. Panel a: drones capture raw, partial images of the field (depending on the mounted sensors). All images from a single flight are collated in a unique orthophotoorthomosaic. Panel b: orthophotosorthomosaics coming from one or more sensors, together with a shape file that specifies the ROIs, enter DRONE2REPORT and are converted to an internal *dataset* object. Panel c: all datasets are inputted, 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 orthophoto 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 a 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.

DRONE2REPORT can receive as input one or more orthophotos. If more than one image 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-channels RGBT *dataset*. Differences in image sizes and resolutions are solved via interpolation and taking into account possible differences in georeferencing data. 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 images channels. This unified representation enables to perform algebraic operations on all combined channels simultaneously, and even combine data coming from different flights and sensors. Even when multiple orthophotos are combined into a *dataset* the ROIs are specified only once.

One *dataset* can thus be composed of ~~several~~ several orthophotos, and several *datasets* 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 is implemented for computing various vegetation indices, rescaling and extracting subimages, and preparing data for successive elaboration (e. g. training neural networks).

2.2. Software Implementation

DRONE2REPORT is a Python-based, open source, free software. It is written to be integrated in standard elaboration pipelines aimed to 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 ~~Box~~ Listing 2, 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/>). Once the configuration file is ready, DRONE2REPORT will be run from the shell command line.

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 3 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.

Finally, new types of DATA could be supported, which would require some care to curate the exposed functionalities.

New indices and functionality 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 [8] and geopandas [9] 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 on-line in the following Github repository: <https://github.com/ne1s0n/drone2report>.

Listing 1: Example of function computing the GLI vegetation index on a passed image

```
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)
    )
```

Listing 2: Example of configuration .ini file to run drone2report

```
[DEFAULT]
#these parameters are shared by all sections
infolder=</base/folder>
outfolder=</output/folder>
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
```

181

Listing 3: Example of function computing the GLI vegetation index on a passed image

```

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')]
    %DIF >
    except ValueError:
        return np.nan
    %DIF >
    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) ~~detection of lodging~~ analysis of height; iv) combining multiple sensors and index optimization; v) leveraging a deep neural network for image classification. 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), whose positive values tend to identify green vegetation (e.g. leaves and stems), while negative values tend to reflect soil/non-vegetation [10]. 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 [11].

To illustrate the removal of background noise, we use one RGB image of tobacco leaves from the Github repository (<https://github.com/tanghaibao/jcvi/wiki/GRABSEEDS>) of the GRABSEED software [4] (see Figure 2)

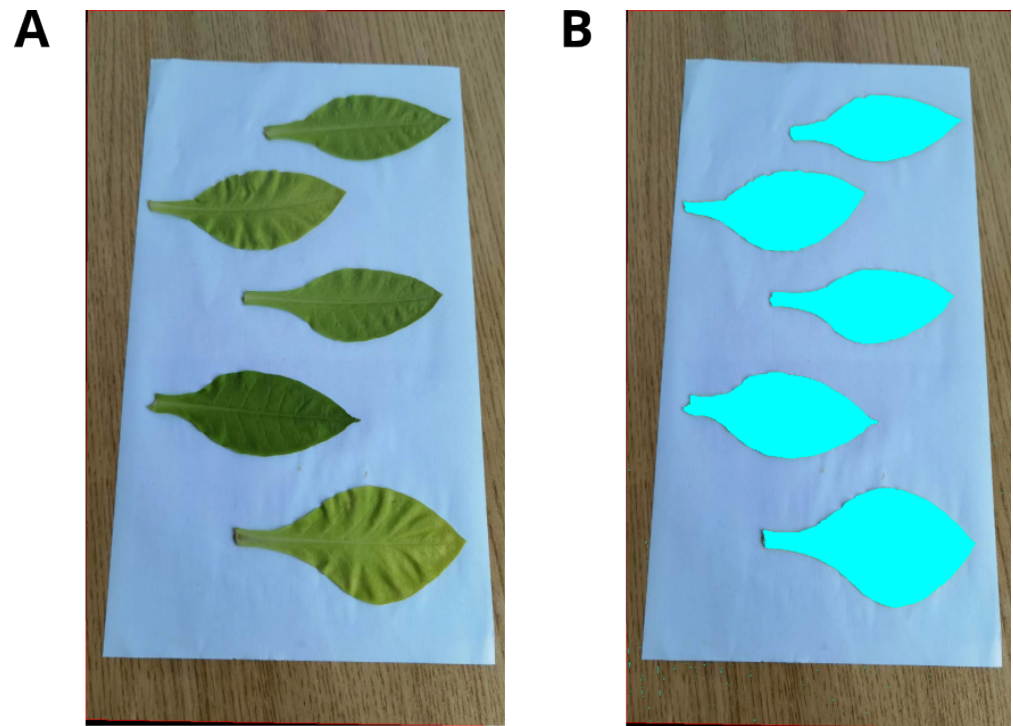


Figure 2. Example RGB image of tobacco leaves (*Nicotiana tabacum*). A) The original image of the tobacco leaves as downloaded from <https://www.dropbox.com/s/is4jrmlmmcfdpdp/test-data.zip?dl=1>; B) The same image with threshold 0.1: the pixels highlighted in cyan are those that will be used for the 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:

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

where $\mathbf{p} := [p_{\text{red}}, p_{\text{green}}, p_{\text{blue}}]$ 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 3. 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.

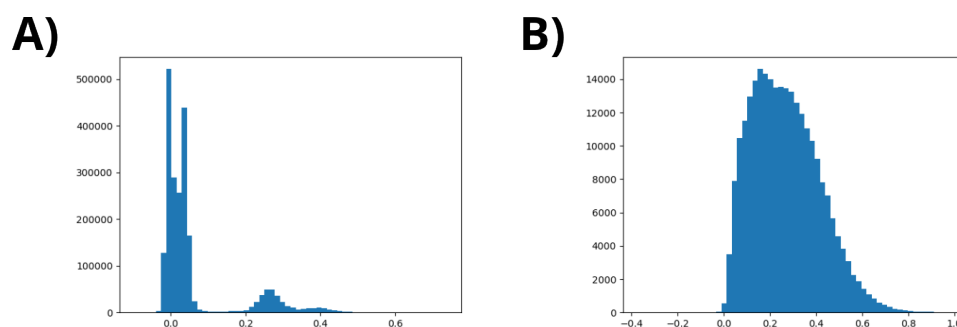


Figure 3. Distribution of values for the vegetation index GLI (Green Leaf Index) for: A) the tobacco leaves image; and B) for the barley field plot image

In a more realistic case the GLI values 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 of a single plot from a barley field in Fiorenzuola d'Arda (PC), Italy, $44^{\circ}55'27''N$, $9^{\circ}54'47''E$, taken on March 25th 2024 ("Polyploidbreeding" research project: <https://polyploidbreeding.ibba.cnr.it/>). For this image, the distribution of GLI index values is unimodal (no clusters: Figure 3, panel B).

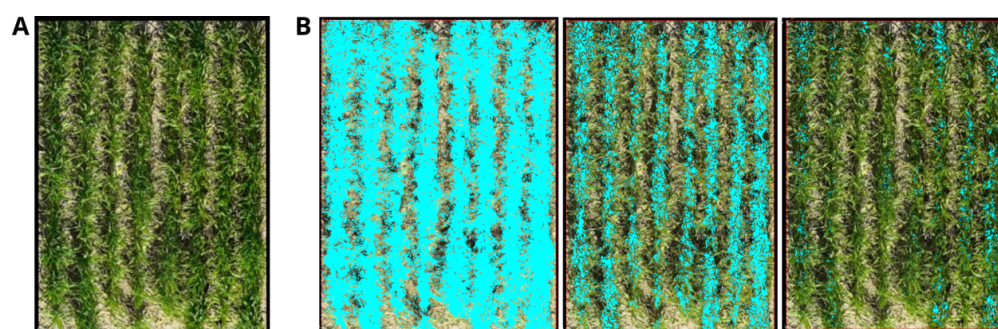


Figure 4. Example RGB image of one single barley field plot. A) The original image of the barley field: all pixels would be used for index calculations; B) The same plot with increasing thresholds for the index values: 0.2, 0.4, 0.6 (rescaled in $[0, 1]$). As the threshold is increased, fewer and fewer 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 4, panel B, 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 used. 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 4 (A) 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 that was 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 $(GLI > 0) \wedge (GLI < 0.8)$. In all cases, the evaluation of different index values and the selection of the best threshold is not automated in DRONE2REPORT.

Table 1. Green-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 this 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	254493	254493	0.26837	0.25397	0.14712	0.97753	-0.36842
GLI > 0.0)	254493	254251	0.26866	0.25424	0.14690	0.97753	0.00000
(GLI > 0.1)	254493	222159	0.29813	0.28090	0.13327	0.97753	0.10000
(GLI > 0.2)	254493	160082	0.35525	0.33333	0.11271	0.97753	0.20000
(GLI > 0.3)	254493	99914	0.41901	0.39706	0.09542	0.97753	0.30000
(GLI > 0.4)	254493	48731	0.49456	0.47200	0.08247	0.97753	0.40000
(GLI > 0.5)	254493	18278	0.58052	0.55932	0.07340	0.97753	0.50000
(GLI > 0.6)	254493	5799	0.67194	0.65289	0.06573	0.97753	0.60000
(GLI > 0.7)	254493	1860	0.76379	0.74713	0.05775	0.97753	0.70000
(GLI > 0.8)	254493	798	0.85531	0.84240	0.04428	0.97753	0.80000
(GLI > 0.9)	254493	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-single-plot image~~ (one single barley variety) at one single timepoint, the images selected for this case study cover 20 plots (20 barley varieties) ~~for over~~ 10 flights. For-Flights were performed in Fiorenzuola d'Arda (Italy, 44°55'40.2466" 009°53'42.4730") between Feb 19, 2024 (first flight on empty field) and June 12, 2024 (tenth flight). As vegetation indices, we use GLI (described before, see Formula 1) and HUE [12], 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 [13].

To calculate vegetation indices over multiple images (i.e. successive flights) ~~for field/area~~-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 ~~the-their~~ trends over time to monitor the evolution of the field. In the barley ~~field~~ example each plot represents a different variety (with repetitions due to experimental design). It is thus possible to ~~also-implement-make~~ inter-varietal comparisons. This is shown in Figure 5 (B), where we highlighted two arbitrary varieties (in red and blue). The

barley plots ~~are shown to~~ 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 5, A). 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. ~~The indexes also show variability between the different~~ Variability exists between barley varieties, probably linked to their different genetic backgroundmake-up.

A



B

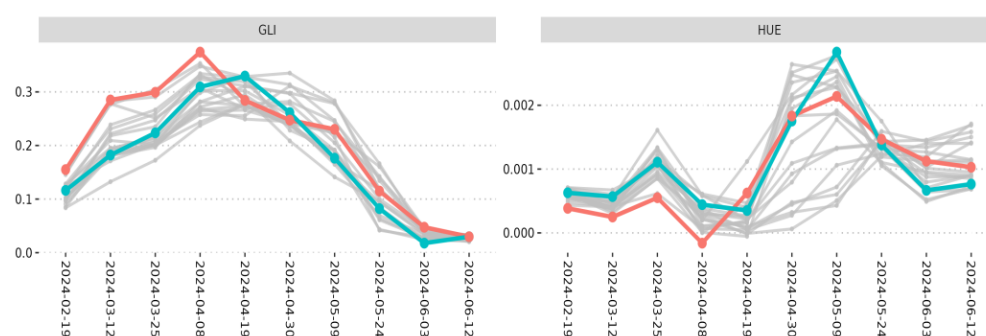


Figure 5. Vegetation indices over time. A) The same barley plot drone-photographed in February, April, and June. B) Evolution of GLI index (left) and HUE index (right) values for twenty barley varieties over the entire growing season (10 flights in February–June 2024). Two arbitrary selected varieties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

3.3. Case Study N. 3: ~~Detecting Lodging~~ Analysis of height from DEM Filesfiles

As mentioned before, drone phenotyping produces also 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 reference. By subtracting the heights of the first flight from all successive flights, it is possible to estimate plant growth.

In Figure 6, A, 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, being soil, an erect plant, or a lodged one. The evolution of per-plot average heights over time is reported in Figure 6 (B) for the successive flights. Note that the very first flight is used as reference and thus all pixels are zero by definition.

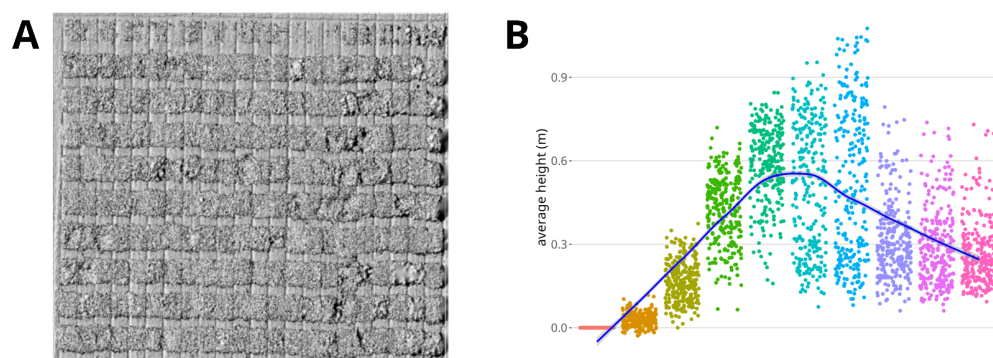


Figure 6. Example .dem file of a barley field (left: flight on 19/04/2024); plant height (metres) calculated from the .dem file by drone2report, over 10 flights covering the growing season (right: [the dates for the 10 flights are the same as in Figure 5](#)).

After the first flight the heights quickly increase up to the 5th–6th flight (mid-April: the plants were growing). In our barley field trials, DRONE2REPORT shows a decrease of 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 examining the right-most part of Figure 5 (A), where a large portion of barley plants appears to have been flattened. ~~Therefore, we can see that the analysis of DEM files allowed us to detect if and when lodging occurred in our field of interest~~ [In general, similar height-decrease patterns may be linked to a variety of cases, 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.](#)

~~It is also possible to estimate~~ [From the DEM files also](#) the per-plot volumes [can be estimated](#) using DRONE2REPORT. Consider that the integral of an area is proportional to the sum of the heights of each pixel. When computing indexes like GLI or HUE the actual result of the computation is a matrix with exactly the same dimension 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 of the DEM file for a plot. This serves as a numeric approximation of the volume, which in turn could 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 wavelength (e.g. RGB images), from ~~multispectral sensors~~ [sensors in the non-visible wavelength](#) (e.g. infra-red and near-infra-red images) and from ~~a thermal camera~~ [thermal cameras](#). Differences in image resolutions are managed automatically via ~~interpolation, and differences~~ [cubic interpolation](#). [Differences](#) in 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), one multispectral (ten bands) and one thermal (one band). 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:

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

where $\mathbf{p} \in \mathbb{R}^{14}$ is a pixel on the merged 14-bands image. The set of 30 coefficients $[n_0, \dots, n_{14}, d_0, \dots, d_{14}]$ are to be chosen to maximize the Pearson's correlation between the array of heading dates and the array of per-plot average index values and the target variable (heading dates, collected in a 264-cells array). Formula 3 is designed as a generalization of the most commonly used indices such as NDVI, GLI, VARI, etc.[14].

This thus becomes a 30-parameters optimization problem, which can be solved via a new custom DRONE2REPORT task. Internally, the task uses Scipy's [15] *optimize.minimize()* function based on the *Nelder-Mead* numerical method [16]. 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. The index was optimized on the merged 14-bands image and, for comparison, on both the visible RGB image and the 10-bands multispectral image, separately. Additionally, the plot also reports the Pearson's correlation with 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).

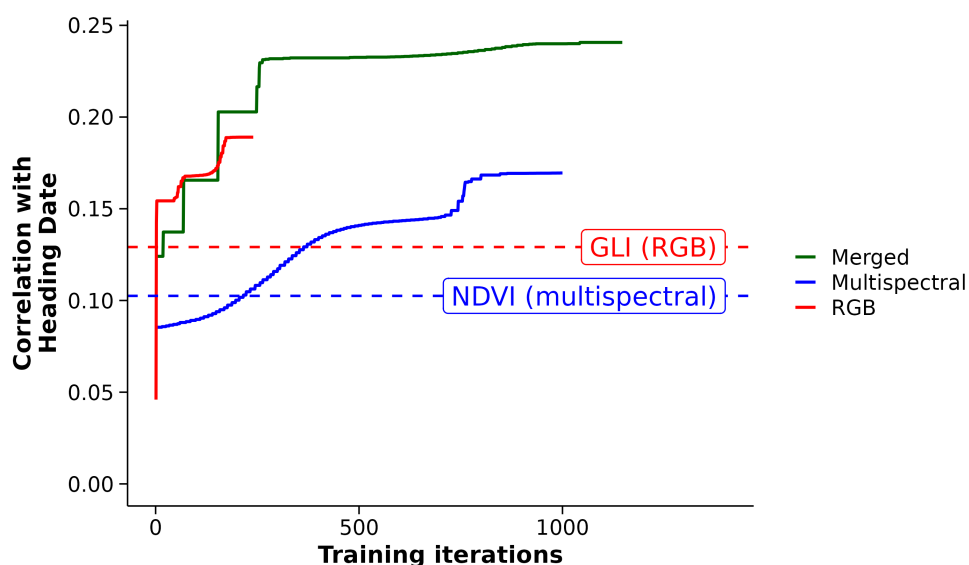


Figure 7. Optimization trajectories through training cycles. The custom index described in Formula 3 was optimized for maximal Pearson's correlation between the vector of its per-plot averages and the phenotype heading date (here acting as target variable). Solid lines show three optimization trajectories on different input images, namely RGB, 10-bands multispectral, and a 14-bands merged image containing RGB, multispectral and temperature data. For reference, the two best performing indices (NDVI and GLI) are also reported.

The results show that it is possible to optimize a custom vegetation index on a specific drone dataset and beat the correlations of standard indices. The optimization process is iterative and stochastic, so different images required different amounts of iteration before converging. Optimizing on the RGB image required little more than two hundred iterations, while on the 14-bands merged image it required 1148 iterations. These numbers would

change if the process were repeated and are influenced by the initialization process, which assigns random values to the parameters. However, in general a longer optimization process is expected the more parameters are to be tuned.

The indices obtained with this case study are likely overfitting, at least partially. A focused, future study, could estimate the fraction of overfitting and apply regularization techniques to ensure good generalization of the results.

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 [17]. As a preliminary step, we trained a deep neural network based on the well-known ResNet50 architecture [18], as implemented in the TensorFlow/Keras framework [19].

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 the 15,926 images available in the dataset, 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%) and validation (30%) subsets. The corresponding training trajectories are reported in Figure 8.

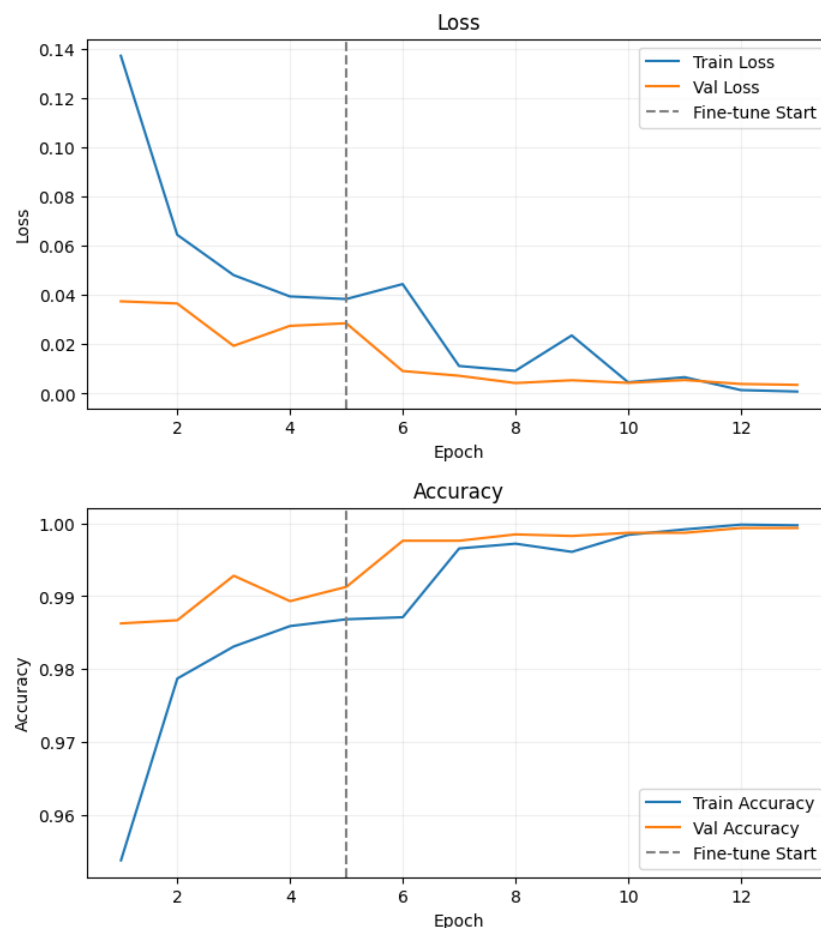


Figure 8. Optimization trajectories through training cycles. The top panel reports the value of the loss function (computed as categorical crossentropy) for the train and validation sets through the training epochs. The bottom panel reports the accuracy, computed as the fraction of correct predictions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

Once training was complete, the trained network object (comprising both the model architecture and the optimized weights) was retained and made available as supplementary material. 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 comprised approximately eighty lines of code, including inline comments and informational messages to guide the user through the execution steps.

Given the demonstrative nature of the task, the model achieved a perfect classification score on the test set.

4. Discussion

In this section, we first compare DRONE2REPORT to the state-of-the-art software tools for drone-images 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 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 [20,21]. These studies consistently highlight persistent bottlenecks not in image acquisition, but in the standardization and analysis of downstream workflows [22].

Several tools and software packages are available for processing and analysing UAV-acquired image data, including commercial products [23]. 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. 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>, [24]) 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. However, its focus is reconstruction; downstream plot-level analytics, multi-sensor alignment, and standardized per-ROI statistics/reporting fall largely outside its scope.

Orfeo ToolBox (OTB: <https://gitlab.orfeo-toolbox.org/>, [25,26]) 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/>, [3]) 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 [27] 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 compute environments.

PlantCV (<https://plantcv.org/>, [28]) 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>, [4]) 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 in input RGB image files, and is not designed to work with UAV-captured field image data.

BerryPortraits (<https://github.com/Breeding-Insight/BerryPortraits/>, [5]) is an open-source Python image-analysis software based on the YOLOv8 model for computer vision [29]. 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 specialised tools, *Berryportraits* only works with RGB images and not with UAV aerial photogrammetry.

FIELDimageR (<https://github.com/OpenDroneMap/FIELDimageR>, [30]) 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>, [31]) 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 Analyser for breeding field (<http://cse.naro.affrc.go.jp/aitoh/PREPs/?en>), [32]) 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=1aZalN0yqli9l2pqyQAPK0liACs7UhrHF>, [33]) is a software platform for the analysis of aerial images which has been 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.

In contrast to most of the above solutions, DRONE2REPORT has been developed specifically to streamline the processing of UAV-captured images for the calculation of vegetation indices and the reporting of relevant summary statistics. It is designed to sit after the orthomosaic generation (e.g. from ODM/WebODM or commercial tools) and provides a configuration-driven, reproducible pipeline that takes in input and multiband image (being it 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) summarising statistics; v) generating thumbnails. Moreover, the software is open source and freely (and easily) customizable.

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 map making, *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. [34–36]). We showed several case studies on how vegetation indices can be obtained and used for a range of applications including: removal of background noise via thresholding, comparison

of different fields and/or plots; time-series analysis, lodging detection, 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, that 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.

In conclusion, DRONE2REPORT is a new software tool designed to support repeatable, scalable, and multi-sensor UAV data analysis in agricultural field trials, and is going to be of great use and help for the scientific and technical community in precision agriculture and beyond. By focusing on vegetation index calculation, ROI-level analytics, and reproducible reporting, it fills a critical methodological gap between raw imagery and downstream statistical analysis. 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:

Author Contributions: NN implemented the core functionalities of the software. NN, GM, FB implemented the case studies. AF, EG, PDF, NF, and GC curated the data for the case studies. GM, EF, FB, NF, and RP tested the software. NN and FB wrote the first draft of the article. All authors reviewed and approved the submitted 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.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: DRONE2REPORT is freely available at <https://github.com/ne1s0n/drone2report>. The code and data to reproduce the case studies for this paper are available at <https://github.com/ne1s0n/paper-drone2report>.

Acknowledgments: We sincerely thank the team Module 7 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. Park, J.; Cho, Y.; Kim, S.: Deep learning-based UAV image segmentation and inpainting for generating vehicle-free orthomosaic. *International Journal of Applied Earth Observation and Geoinformation* **115**(103111) (2022)

2. 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. *Frontiers in plant science* **12**, 749374 (2021)
3. Moyroud, N., Portet, F.: Introduction to qgis. *QGIS and generic tools* **1**, 1–17 (2018)
4. 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* **20**(1), 140 (2024)
5. 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* **20**, 172 (2024)
6. Welcome to OpenDroneMap’s documentation — OpenDroneMap 3.5.6 documentation. <https://docs.opendronemap.org/> Accessed 2025-08-27
7. Rupnik, E., Daakir, M., Pierrot Deseilligny, M.: Micmac—a free, open-source solution for photogrammetry. *Open geospatial data, software and standards* **2**(1), 14 (2017)
8. GDAL/OGR contributors: GDAL/OGR Geospatial Data Abstraction Software Library. Open Source Geospatial Foundation, (2025). <https://doi.org/10.5281/zenodo.5884351> . Open Source Geospatial Foundation. <https://gdal.org>
9. Jordahl, K., Bossche, J.V., Fleischmann, M., Wasserman, J., McBride, J., Gerard, J., Tratner, J., Perry, M., Badaracco, A.G., Farmer, C., Hjelle, G.A., Snow, A.D., Cochran, M., Gillies, S., Culbertson, L., Bartos, M., Eubank, N., maxalbert, Bilogur, A., Rey, S., Ren, C., Arribas-Bel, D., Wasser, L., Wolf, L.J., Journois, M., Wilson, J., Greenhall, A., Holdgraf, C., Filipe, Leblanc, F.: Geopandas/geopandas: V0.8.1. <https://doi.org/10.5281/zenodo.3946761> . <https://doi.org/10.5281/zenodo.3946761>
10. Louhaichi, M., Borman, M.M., Johnson, D.E.: Spatially located platform and aerial photography for documentation of grazing impacts on wheat. *Geocarto International* **16**(1), 65–70 (2001)
11. Hamuda, E., Glavin, M., Jones, E.: A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and electronics in agriculture* **125**, 184–199 (2016)
12. 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 sensing of environment* **66**(1), 17–28 (1998)
13. Meyer, G.E., Neto, J.C.: Verification of color vegetation indices for automated crop imaging applications. *Computers and electronics in agriculture* **63**(2), 282–293 (2008)
14. Henrich, V., Krauss, G., Götze, C., Sandow, C.: Idb - www.indexdatabase.de, entwicklung einer datenbank für fernerkundungsindizes. Bochum: AK Fernerkundung **15** (2012)
15. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy 1.0 Contributors: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **17**, 261–272 (2020) <https://doi.org/10.1038/s41592-019-0686-2>
16. Nelder, J.A., Mead, R.: A simplex method for function minimization. *The computer journal* **7**(4), 308–313 (1965)
17. Peccia, F.: Weed detection in soybean crops (2018). <https://doi.org/10.17632/3fmjnm7ncc6.2> . <https://www.kaggle.com/datasets/fpeccia/weed-detection-in-soybean-crops> Accessed 2026-01-15
18. 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*, pp. 770–778 (2016)
19. Chollet, F., *et al.*: Keras. <https://keras.io> (2015)
20. 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* **8**(6), 212 (2024) <https://doi.org/10.3390/drones8060212> . Accessed 2025-08-27
21. 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. *The Plant Phenome Journal* **7**(1), 20100 (2024)
22. 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. *The Plant Phenome Journal* **7**(1), 20100 (2024) <https://doi.org/10.1002/ppj2.20100> . Accessed 2025-08-27
23. Sharma, M., Raghavendra, S., Agrawal, S.: Development of an open-source tool for uav photogrammetric data processing. *Journal of the Indian Society of Remote Sensing* **49**(3), 659–664 (2021)
24. Vacca, G., *et al.*: Web open drone map (webodm) a software open source to photogrammetry process. In: *Fig Working Week 2020. Smart Surveyors for Land and Water Management*, (2020)
25. Orfeo ToolBox – Orfeo ToolBox is not a black box. <https://www.orfeo-toolbox.org/> Accessed 2025-08-27

26. Grizonnet, M., Michel, J., Poughon, V., Inglada, J., Savinaud, M., Cresson, R.: Orfeo ToolBox: open source processing of remote sensing images. *Open Geospatial Data, Software and Standards* **2**(1), 15 (2017) <https://doi.org/10.1186/s40965-017-0031-6> . Accessed 2025-08-27
27. Dobesova, Z.: Evaluation of effective cognition for the qgis processing modeler. *Applied Sciences* **10**(4), 1446 (2020)
28. PlantCV (2021). <https://plantcv.org> Accessed 2025-08-27
29. 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), pp. 1–6 (2024). IEEE
30. Matias, F.I., Caraza-Harter, M.V., Endelman, J.B.: FIELDImageR: An R package to analyze orthomosaic images from agricultural field trials. *The Plant Phenome Journal* **3**(1), 20005 (2020) <https://doi.org/10.1002/ppj2.20005> . Accessed 2025-08-27
31. Chen, C.J., Zhang, Z.: Grid: A python package for field plot phenotyping using aerial images. *Remote Sensing* **12**, 1697 (2020) <https://doi.org/10.3390/rs12111697>
32. Itoh, A., Njane, S.N., Hirafuji, M., Guo, W.: Preps: An open-source software for high-throughput field plant phenotyping. *Plant Phenomics* **6**, 0221 (2024)
33. 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* **6**, 0164 (2024)
34. 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 sensing* **7**(4), 4026–4047 (2015)
35. 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* **13**(3), 707 (2023)
36. 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* **6**(3), 421–436 (2023)

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.