

Article

A Rapid Review on the Use of Free and Open Source Technologies and Software Applied to Precision Agriculture Practices

Rogério P. dos Santos ^{1,2}, Nuno Fachada ², Marko Beko ^{2,3} and Valderi R. Q. Leithardt ^{2,4,*}¹ Campus Capanema, Instituto Federal do Paraná—(IFPR), Rua São João, Realeza 2343, Paraná, Brazil² COPELABS, Lusófona University, 1749-024 Lisbon, Portugal; nuno.fachada@ulusofona.pt (N.F.);

beko.marko@ulusofona.pt (M.B.)

³ Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal⁴ VALORIZA—Research Centre for Endogenous Resource Valorization, Polytechnic Institute of Portalegre, 7300-555 Portalegre, Portugal

* Correspondence: valderi@ippportalegre.pt

Abstract: Technology plays a crucial role in the management of natural resources in agricultural production. Free and open-source software and sensor technology solutions have the potential to promote more sustainable agricultural production. The goal of this rapid review is to find exclusively free and open-source software for precision agriculture, available in different electronic databases, with emphasis on their characteristics and application formats, aiming at promoting sustainable agricultural production. A thorough search of the Google Scholar, GitHub, and GitLab electronic databases was performed for this purpose. Studies reporting and/or repositories containing up-to-date software were considered for this review. The various software packages were evaluated based on their characteristics and application formats. The search identified a total of 21 free and open-source software packages designed specifically for precision agriculture. Most of the identified software was shown to be extensible and customizable, while taking into account factors such as transparency, speed, and security, although some limitations were observed in terms of repository management and source control. This rapid review suggests that free and open-source software and sensor technology solutions play an important role in the management of natural resources in sustainable agricultural production, and highlights the main technological approaches towards this goal. Finally, while this review performs a preliminary assessment of existing free and open source solutions, additional research is needed to evaluate their effectiveness and usability in different scenarios, as well as their relevance in terms of environmental and economic impact on agricultural production.

Keywords: software evaluation; low-cost solutions; automation; iot; monitoring



Citation: dos Santos, R.P.; Fachada, N.; Beko, M.; Leithardt, V.R.Q. A Rapid Review on the Use of Free and Open Source Technologies and Software Applied to Precision Agriculture Practices. *J. Sens. Actuator Netw.* **2023**, *12*, 28. <https://doi.org/10.3390/jsan12020028>

Academic Editors: Hossein Anisi and Pelin Angin

Received: 15 February 2023

Revised: 16 March 2023

Accepted: 21 March 2023

Published: 24 March 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Free and open-source applications for precision agriculture are becoming increasingly popular as they offer an affordable and accessible alternative to proprietary systems. These applications—which are freely licensed to use, copy, study, modify and/or improve [1]—provide farmers with access to data and tools that can help them make informed decisions about their crops and land management. Free and open-source applications for precision agriculture can also help farmers to better monitor and manage their resources, as well as to increase their yields and reduce their environmental impact. Additionally, these applications can independently be built upon to develop innovative solutions to address specific challenges in the agricultural sector [2], something which would be difficult or even impossible with proprietary solutions [3]. These innovative technologies have been improved by incorporating software, hardware, and data [4].

Precision agriculture is a technological or agricultural management approach that uses various sources of data and information from sensors, satellite images, global positioning systems, and specific software to make more accurate and personalized decisions at all stages of the agricultural production process [5]. Through the application of these technologies, it is possible to identify issues and better explore these cultures, while being able to monitor weather conditions, analyze soil and plants, and apply inputs more efficiently [6]. Precision agriculture can be seen as a subsection of precision practices facilitated by technological advances [7]. Precision practices occur in four main stages: data collection, planning, implementation, and culture management [6].

Increasingly, agricultural production has been benefiting from open source technological applications for precision practices, leveraging their cost-reduction potential to promote agricultural sustainability [8,9]. Information and communication technology (ICT) solutions, such as artificial intelligence (AI), machine learning, and the Internet of Things (IoT), are being used to improve the efficiency and accuracy of agricultural processes. AI can be used to monitor crop health, detect pests, and predict crop yields. Machine learning can be used to analyze large datasets and identify patterns in agricultural data. IoT can be used to connect different devices and sensors in the field to provide real-time data and insights into crop health, soil conditions, and weather patterns [10].

There are several options in terms of free and open-source technologies and software for precision agriculture, and selecting a context-appropriate technological tool from among those available can be crucial in obtaining good results [11]. Therefore, technology transfers of open-source software can be an essential and transformative activity for precision agriculture, bridging the gap between theory and reality [12].

The selection of free and open-source software for agricultural purposes is based on a number of factors, such as its technical features, maintenance burden (and its associated cost), ease of use, documentation, security, and compatibility with existing systems [13,14]. In essence, it is crucial to ensure that the software is able to meet the requirements of the agricultural sector and is able to handle the data and tasks efficiently.

According to De Oca and Flores [15], the development of open-source software aimed at precision agriculture has grown thanks to the involvement of three main technologies: computer vision, drones, and AI. Recently, a series of studies also propose using sensors to monitor production, improving the quality and quantity of crop data, and allowing, together with the appropriate software, to provide information, reminders and alerts to farmers [16].

In order to ensure the success of these initiatives, it is important to understand the needs of the farmers, the existing infrastructure, and the available technologies. Additionally, it is essential to identify the most suitable and cost-effective solutions to meet the requirements of the farmers and to ensure that the data collected is accurate, secure, and reliable. Finally, it is necessary to develop strategies to ensure that the data collected is used to improve the efficiency of the agricultural production process [17].

Faced with this scenario, we carried out a rapid review [18] with the goal of identifying free and open-source software capable of solving real-world problems in precision agriculture. For this purpose, we simplified the PRISMA extension for scoping reviews [19], adapting it to the topic under investigation [20] in the context of a rapid review [18]. Searches were carried out in electronic databases, drawing attention to current field-wise demands—as well as their interrelationships—and highlighting the programming languages used in their development and integration. The purpose of this investigation is to understand the tools and technological solutions available to meet the specific requirements of precision agriculture, thus providing a decision support document to stakeholders [21] of this increasingly popular approach.

This paper is organized as follows. Section 2 presents the methods used in this review, namely, information sources, search strategy, eligibility criteria, and selection procedures, and data items extracted from each selected work. Section 3 presents the results, summarizing the selected works and extracted data, and offering a narrative synthesis of

the applications found. Section 4 provides a discussion and critical analysis of this research, as well as of its limitations. Finally, Section 5 puts forward our final considerations.

2. Methods

2.1. Information Sources

For the purpose of this rapid review, we used Google Scholar, GitHub, and GitLab as electronic data sources, as shown in Table 1, together with the respective URLs. The most recent search was carried out in October 2022.

Table 1. Information sources used in this review and their respective URLs.

Source	URL
Google Scholar	https://scholar.google.com
GitHub	https://github.com
GitLab	https://gitlab.com

Google Scholar is a free and accessible search engine that organizes a variety of scientific scholarly publication formats, and is currently the most comprehensive database of scientific articles [22]. In turn, GitHub and GitLab are publicly available and searchable open-source software repositories. In particular, GitHub is by far the most used code repository in general [23] (with GitLab comfortably in second position [23]), and in scholarly publications in particular [24].

2.2. Search Strategy

The works discussed in this review were identified by the following search string:

“open-source software” and “precision agriculture”

After defining the search string and the data sources to be searched, the key terms corresponding to each of the information sources were also created. Key terms are defined junctions within the search string and separated by quotes. Table 2 presents the information source and the respective search strings. Some tweaks within the search strings column are also presented. In the Google Scholar database row, exclusion attributes are added, and, in the GitHub and GitLab database rows, search strings were defined and carried out by topic.

Table 2. Search strings used for each information source.

Source	Search Strings
Google Scholar	“open-source software” and “precision agriculture” -simulators-games-“farming simulator”
GitHub	“precision agriculture”
GitLab	“precision agriculture”

Searches were conducted by the first author, with results revised by the second author over multiple iterations according to the selection procedure discussed in the following subsection.

2.3. Eligibility Criteria and Selection Procedure

After defining the search string for each information source, it was necessary to define the inclusion and exclusion criteria. If a specific work is eligible according to these criteria, it is automatically selected. Care is taken so that possible duplicate entries are merged.

The following inclusion criteria were considered for this review:

- Works focused on precision agriculture;

- Works consisting of free and open-source software with full source code publicly available, preferably in a software repository such as GitHub, GitLab, or equivalent;
- Works published between 2012 and 2022;
- Source code must have been updated at least once in the last 3 years (counting down from October 2022);
- Works only in the English language.

The rationale for selecting works published since 2012 is based on the availability and cost of unmanned aerial systems (UAS) and 4G technologies, essential for precision agriculture, and which started to become mainstream around that year [25,26]. With respect to works with code updated in the last 3 years, this was an empirically selected time interval to exclude deprecated or abandoned software, a common occurrence in open source projects [27].

In turn, three exclusion criteria were defined, namely:

- Applications not directly related to precision agriculture;
- Farming simulator applications, farm simulator games;
- Other games.

2.4. Data Items

For each selected work, we extracted six data items, namely:

- Source code repository;
- Year of publication;
- Programming language(s);
- User interface(s);
- Category;
- Keywords.

The source code repositories for the selected works are either found directly from the searches performed on GitHub and GitLab, or from the associated research paper found on Google Scholar (if there is such a paper, and if mentions the repository). The year of publication is taken from the publication date of the research article, or in case the work does not have an associated paper, the year of the first commitment on the respective source code repository.

The programming language in which an application is developed—although mainly of interest to developers—is especially important in free and open-source software, which can be freely modified, improved and extended. Developers or technically-oriented users should preferably consider software developed in a language they are familiar with, specifically if they intend to modify the source code and/or implement new features. Software can be developed in more than one language, and this should also be factored in the selection process, as it can increase development complexity. For the purpose of this review, programming languages were determined from the GitHub’s “Languages” feature, which details the languages used in a project by percentage, or, if this information was insufficient or nonexistent, by manually browsing the source code. We only consider languages that are used for developing the projects, ignoring languages used exclusively for documentation (e.g., HTML or Markdown) or for the build process (for example, Bash scripts or Makefiles).

The user interface defines the way that the end user interacts with the software in question. It can be a frontend interface, such as a *graphical* or *text* interface, or a backend interface, namely working as a *library* or as a *plugin* for another tool. The user interface may narrow the type of end users of a given package. For example, text interfaces—which allow a tool to be controlled via the command line—as well as libraries, are typically more suited to developers and/or technically-minded users. Several tools offer more than one type of interface, therefore broadening their scope of potential users.

Technology and software applications were grouped according to the following categories:

- A Crop and climate protection and diagnosis;

- B Nutrition and fertilization of crops;
- C Crop irrigation;
- D Soil management, planting, growing and harvesting crops;
- E Production management, machinery and equipment.

Some applications, due to their varied number of features, fall into more than one category. In this case, they were placed in the category corresponding to their main functionality and purpose. However, since predefined categories may limit a broader understanding of the underlying topics, we additionally derived five keywords for each of the selected works. Keywords were agreed upon by the first and second authors, and were obtained from one or more of the following sources, ordered by relevance: explicit keywords in referenced article, if available; extracted from the abstract; extracted from the full article; and/or, obtained from the README file in the respective code repository. Keywords with similar meanings or different capitalizations were merged together as lowercase concepts. For example, “UAV”, “UAS”, “unmanned aerial vehicle”, or “drone” were merged into “uas”. Keywords directly pertaining to the research topic were explicitly ignored, namely “precision agriculture”, “open source”, “free and open source”, “software”, as well as possible combinations, as we expect these to be representative of all the selected works, and therefore would not provide a deeper understanding of the fundamental themes under discussion. Programming language names, such as Python or Java, were also ignored, as these are analyzed separately.

3. Results

3.1. Selection of Works

Table 3 presents the quantification of the works found in their respective electronic databases. A total of 249 articles were identified on Google Scholar and 59 projects were found in the GitHub and GitLab repositories, with a grand total of 308 works recognized using the search strings described in Section 2.2. After applying the filtering criteria, 234 papers from the Google Scholar database and 53 packages from GitHub and GitLab were excluded, in a total of 287 excluded works. The works included are then presented, of which 15 are from Google Scholar and 6 are from GitHub. Therefore, a new scenario is presented with 21 works, corresponding to the proposal defined in the research objectives.

Table 3. Number of works identified with the search strings for each information source, as well as number of works excluded and included after the filtering criteria has been applied. Value in bold corresponds to the number of works selected for this review.

Info. Source	Identified	Excluded	Included
Google Scholar	249	234	15
GitHub	54	48	6
GitLab	5	5	0
Totals	308	287	21

3.2. Characteristics of Selected Works

The characteristics of selected works are detailed in Table 4, namely the bibliographic reference, information source, short name, user interface(s), category, programming language(s), and link to source code repository. The keywords derived for each work are provided in Table A1 in the Appendix A.

Table 4. Selected works, together with their bibliographic reference, information source, short name, user interface(s), category, programming language(s), and link to source code repository (all links last accessed on 12 March 2023). Categories include: (A) crop and climate protection and diagnosis; (B) nutrition and fertilization of crops; (C) crop irrigation; (D) soil management, planting, growing and harvesting crops; and (E) production management, machinery, and equipment.

No.	Ref.	Source(s)	Short Name	Interfaces	Cat.	Language(s)	Repository
1	[28]	Scholar	Agri-robot	Text, Graphic	D	Python	github.com/Harry-Rogers/PiCar
2	[29]	Scholar	MeteoMex	Text	A	C/C++	github.com/robert-winkler/MeteoMex
3	[30]	Scholar	OWL	Graphic	D	Python	github.com/geezacoleman/OpenWeedLocator
4	[31]	Scholar	AgriQ	Text, Graphic	D	Python	github.com/LAPyR/NDVI-with-modified-Mobius-cameras
5	[32]	Scholar	EasyIDP	Text, Graphic, Library	B	Python	github.com/UTokyo-FieldPhenomics-Lab/EasyIDP
6	[33]	Scholar	AGRO IoT	Text	A	C/C++, Python	github.com/ajayarunachalam/RPA
7	[34]	Scholar	HydroTi	Graphic	C	Java, JavaScript	github.com/AnthonyFTL
8	[35]	Scholar	GeoFIS	Graphic, Plugin	E	Java, C/C++, R	www.geofis.org
9	[36]	Scholar	SoftwarePilot	Text	A	Java, Python	github.com/boubinng/softwarepilot
10	[37]	Scholar	DAGAPy	Text, Graphic Library	D	Python	github.com/LabDig/DAGAPy
11	[38]	Scholar	EVAPO	Graphic	C	Java	github.com/waltermaldonado/EVAPO
12	[15]	Scholar	UAS Thermal Imaging	Text, Graphic, Library	C	MATLAB, Python	github.com/LAPyR/Thermal-imaging-workflow-code
13	[39]	Scholar	ITALLIC	Library	A	Python	github.com/getiria-onsongo/itallic
14	[40]	Scholar	Ag-YOLO	Text	A	C/C++	github.com/rossqin/RQNet
15	[41]	Scholar	VICAL	Graphic	D	JavaScript	github.com/CenidRaspaRiego/VICAL
16	[42]	GitHub	MapFilter 2.0	Graphic	A	Java	github.com/LeonardoAgricola/MapFilter2.0
17	[43]	GitHub	AI Agro	Library	E	Python	github.com/RentadroneCL/AI-Agro
18	[44]	GitHub	BovHEAT	Text	E	Python	github.com/bovheat/bovheat
19	[45]	GitHub	Adaptive Irrigation	Text	C	Python, C/C++	github.com/SoothingMist/Embeddable-Software-for-Irrigation-Control
20	[46]	GitHub	Deepway	Library	E	Python	github.com/fsalv/DeepWay
21	[47]	GitHub	RQGIS	Plugin, Library	E	R, Python	github.com/r-spatial/RQGIS3

3.3. Description of Selected Works

Individual works are described in the following Sections 3.3.1–3.3.5, where applications are grouped according to the categories defined in Section 2.4. For each software, a brief description is presented, highlighting its main characteristics and functionalities, while its potential of use is framed and contextualized within the various categories of precision agriculture.

3.3.1. A—Crop and Climate Protection and Diagnosis

Winkler [29] presents the MeteoMex project (Table 4, item 2), which aims to integrate current IoT platforms for community-oriented environmental monitoring. This monitoring helps to protect natural resources such as water and energy, as well as detect timely health risks and improve the quality of food production. The use of low-cost mini-boards connected to environmental sensors from the Things Board open-source platform is demonstrated. Programmed in Arduino C with WiFi capability, MeteoMex can be deployed for monitoring air and soil quality. The authors also discuss a prototype to monitor effluent treatment. It uses a PostgreSQL database with the goal of simplifying the process of data mining. It is considered highly scalable and cost-effective, and may be suitable for deployment in precision agriculture, industry contexts, and public areas. As examples of concrete cases, the following are described: monitoring ambient air at the national institute of respiratory diseases in Mexico, monitoring greenhouses (air and soil), monitoring air conditioning in a high-performance computer room, and monitoring domestic wastewater stations.

Arunachalam and Andreasson [33] propose a continuous crop monitoring system called smart AGRO IoT (Table 4, item 6). The system automates the deployment of sensors

and storage, providing an elegant dashboard for visualizing the continuous stream of real-time data. The system has a sensor data acquisition mechanism, collecting data in a continuous fashion, and is easy to implement in growth chambers and greenhouses. The system can be extended to farms with specific configurations, while its instances are heterogeneous in terms of configuration and installation. The proposed solution, based on Raspberry Pi or Arduino hardware, is scalable and provides stability in monitoring any environment. The system is functional in x , y and z axes along the rails with transverse slides. Soil loosening, digging, and seeding can be performed with a robot arm by providing x and y coordinates as location points via the program's interface. Once the coordinates are specified, the robot arm is moved to the location point and sowing starts with the digging operation, after which the seeds are sown.

Boubin and Stewart [36] present the SoftwarePilot middleware, (Table 4, item 9), used in fully autonomous precision agriculture applications. The system provides data detection and flight control utilities, allowing users to customize machine learning models for pathfinding, object detection, and data analysis. The work discusses the subcomponents of the SoftwarePilot microservices design, namely the routines and drivers, and how they can be used to control UAV flights autonomously and in data sensing. An interactive coding session is also featured, teaching how to develop a simple application. Finally, the authors report that the system has been used to process thousands of images captured autonomously in agricultural missions.

Onsongo et al. [39] developed the Integrated Tool for AgData Lat Long Imputation and Cleaning (ITALLIC) system (Table 4, item 13), which detects and corrects errors in location data by imputing missing values to location-dependent data. An open source API is used to geocode the location. The system, developed in Python, includes a visualization tool that facilitates the validation of results by the user by plotting the original and corrected locations on a map.

Qin et al. [40] propose Ag-YOLO (Table 4, item 14), a low-cost real-time detector for pulverization applied to case studies with palm trees. The software developed and presented in this study is adapted for use in embedded RGB camera systems. With the increasing availability of drones that can spray pesticides, the included algorithm can contribute to selective pulverization. The system receives as input the images acquired by an on-board camera and sends the instructions to the flight management unit in real time, achieving better levels of accuracy and representation power, both in terms of computation cost and memory usage. This seeks to save and reduce the environmental impact and economic costs to the farmer. According to the author, the technology is low-cost and necessary for the widespread use of target-oriented selective spraying.

Maldaner et al. [42] present the MapFilter 2.0 software (Table 4, item 16), developed to analyze and remove inconsistent data in high-density agricultural datasets. The authors claim that the software is easy to install and has a friendly interface, helping users to generate reliable maps of the areas of cultivation based on the optimization of data quality. The project, developed in the Java language, performs a visual analysis of the data and reapplies the global and local filtering with other input values. All statistical analyses were performed in the R software [48] using the gstat library [49]. The developed application demonstrated the potential for practical use of spatial data filtering and received Brazilian patent No. BR512019002014-6.

3.3.2. B—Crop Nutrition and Fertilization

Wang et al. [32] present EasyIDP (Table 4, item 5), an open-source software package, developed with the aim of reducing the workload in processing raw image data obtained from UAVs. The focus of the package is on agricultural applications, since drone photometry techniques are widely used for phenotyping high-yielding plants. Six field datasets with different crop characteristics were selected to develop and test the performance of the proposed package. The tool has considerable potential in reducing the computational load in data annotations for machine learning applications. According to the authors, results

showed that the tool works as expected, both in cropping and reversing tasks tested on a variety of crops.

3.3.3. C—Crop Irrigation

Irrigation management can achieve better ways of using water resources in the context of precision agriculture, and is one of the most important and complex steps in plantation environments [50]. In this sense, with precision irrigation, farmers seek to address drought situations, minimizing the use of water without devaluing the crop. It is possible to estimate the water needs for a crop through a set of equations extracted from the physics of evapotranspiration [45]. Other irrigation systems use intelligent approaches for different crops in greenhouses, allowing, for example, to autonomously activate or deactivate the water supply system depending on the environmental conditions and soil moisture [51].

An application for Android smartphones called EVAPO is presented by Júnior et al. [38] (Table 4, item 11), the objective of which is to estimate evapotranspiration in real time, anywhere in the world, increasing water conservation and seeking more efficient irrigation. It uses climate data from conventional surface weather stations. The app is available for free on the Play Store. The results presented in this study conclude that the data can be used reliably, estimating evapotranspiration and enabling its applicability to smartphones. The performance of the proposed method was evaluated, comparing the evapotranspiration power, estimated by EVAPO using climate data from conventional surface stations in various locations around the world. The agrometeorological data used from these stations were average, maximum and minimum air temperature, relative humidity, insolation, and wind speed.

Carrillo-Pasiche et al. [34] present an irrigation system for urban areas based on IoT (Table 4, item 7). The system analyzes important factors for an effective irrigation process, considering the use of water in an economical and viable way. In this sense, a prototype was developed with Arduino Uno, connected to sensors and driven by a web application called HydroTi, capable of determining when to irrigate and how much water to use. The solution was evaluated by comparing the different types of irrigation in order to determine the water consumption and validating its effectiveness.

Raeth [45] presents a system (Table 4, item 19) that calculates irrigation needs in crops based on information specific to the crop, soil, terrain and climatic conditions. The system allows the development of the irrigation schedule for different management conditions and the calculation of the water supply scheme for different cultivation patterns. Furthermore, the system can also be used to evaluate farmers' irrigation practices and estimate crop performance under irrigation conditions. The software uses meteorological data and evapotranspiration equations to demonstrate an approach to limiting the amount of water withdrawn for irrigation. The software is written in Python and C++, and can be embedded in human-supervised but fully automated irrigation controllers.

De Oca and Flores [15] present the development of an unmanned aerial system to capture and process vegetation index thermal images (Table 4, item 12). This index is used to determine water stress and evaluate the irrigation process. An interface based on the Raspberry Pi was also designed, collecting the necessary images by means of a thermal camera. The entire image correction process has a significant impact in increasing the quality of the orthomosaic. To generate the orthomosaic from the aerial images, the WebODM software was used. Temperature calibration works by applying a linear model for temperature estimation and the vegetation index is calculated from the corrected and calibrated images. The authors, based on the experimental results, state that the proposed system works well for its intended purposes.

3.3.4. D—Soil Management, Planting, Growing, and Harvesting the Crop

De Freitas Coelho et al. [37] present an embedded system to perform an analysis of spatial variability data shortly after its acquisition in the field (Table 4, item 10). Sampling strategies typically happen manually and make the analysis process difficult. To avoid

these problems, the proposed software solution performs the analysis immediately after collection. Hardware with sensor communication ports and good processing power is suggested. The system was developed in Python, and, according to the authors, proved to be efficient with tests in soybean planting, while the comparison of the generated maps demonstrated the importance of filtering the data before analyzing it.

De Oca and Flores [31] propose the AgriQ system (Table 4, item 4), an unmanned aerial system (UAS) for precision agriculture. The design, which includes both hardware and software, allows for a flexible, low-cost approach. The experiments carried out took place in two different vegetation areas with eight vegetation indices. Qualitative and quantitative comparisons were made to assess the quality of the system versus a commercial application. The main advantage found was the smaller amount of images required to reconstruct a map, resulting in a reduction of computational load and processing time.

Rogers and Fox [28] present a terrestrial robot designed to move autonomously, being able to store the geospatial location of each seed with precision (Table 4, item 1). The robot platform was designed to plant seeds with greater precision, within the georeferenced line. The system can plan and control an efficient point-to-point route, including accelerating on long straight lines and slowing down on corners. Through a geospatial location module, the robot stores the coordinates of each planted seed so that it can return to it. According to the performed experiments, the authors claim that the differential location accuracy is such that driving the robot to revisit specific plants is possible for seed spacings up to 139 mm.

Coleman [30] developed the OpenWeedLocator (OWL) project with the aim of maximizing crop yield potential in agricultural environments with limited humidity and a focus on weed removal (Table 4, item 3). The system integrates weed detection on a Raspberry Pi with a relay control board that mounts on robots and vehicles for timely spraying. Four color-based algorithms that exploited the green color levels of weeds for detecting them were used to validate the hardware. All image processing software was written in Python, making use of OpenCV [52] and NumPy [53], as well as several other libraries.

Jiménez-Jiménez et al. [41] present the VICAL system (Table 4, item 15), an open source tool to calculate maps used in agricultural applications, as well as time series of any agricultural area in the world, leveraging data from the Landsat and Sentinel-2 satellites. The system was implemented in JavaScript and developed on the free Google Earth Engine platform. The tool can be beneficial for users with poor access to satellite data or lack of computing infrastructure to handle large volumes of data. To evaluate the performance of the VICAL system, individual images were downloaded from agricultural stations in two irrigation districts in Mexico, located in the north of Sinaloa state and in the Lagunera region, respectively. According to the authors, this system can save time and prevent error-prone repetitive and trivial procedures, associated with manual calculations.

3.3.5. E—Management of Production, Animals, Machinery and Equipment

Plenio et al. [44] present an application called the bovine heat detection and analysis tool (BovHEAT), an open-source analysis tool to process data from automated systems and activity monitoring in dairy cattle (Table 4, item 18). Activity data is collected by an accelerometer attached to the animal's neck, which is now widely used in the dairy industry. Developed with the Python programming language, BovHEAT offers fully automatic and scalable processing, and is able to output data in various formats. The advantages of this tool are in obtaining a better understanding of the physiology and behavior of dairy cows and in supporting decision-making in optimizing reproductive management. Two different data sets of eight farms were used: one farm in Canada and seven farms in Germany. Its validation was carried out independently by three researchers involved in the project. Activity data from 60 cows were used, representing a total number of 600 observations per cow, with results indicating that the system requires less user interaction and speeds up data processing.

Leroux et al. [35] present the GeoFIS software (Table 4, item 8). The purpose of this tool is to support the entire process of precision agriculture, from spatial data to spatial

information and decision support. It was designed with a user-friendly interface with the intention of supporting users with no programming skills. Experiments were carried out in three types of crops: banana, wheat, and vineyards. These were chosen to highlight the diversity of applications and data characteristics that can be handled with GeoFIS. The analysis of the three types of crops was an opportunity to also assess the limits of the algorithms and propose new areas for future developments within the tool itself.

Muenchow et al. [47] present RQGIS (Table 4, item 21), an integration of R with the open source QGIS system for statistical geocomputing and geographic information systems (QGIS Development Team, 2022). The proposal extends the statistical resources of R with numerous geoprocessing and data manipulation tools available in a geographic information system (GIS). The combination of R and GIS software creates a powerful environment for geocomputing and advanced statistics. RQGIS supports R named data, seamless exchange of spatial data types, and quick access to a broad suite of geoalgorithms for solving many GIS problems, among which the authors highlight those related with precision agriculture. However, continuous development of QGIS led to a number of unsolvable issues in RQGIS, which has since been deprecated according to its repository. The `qgisprocess` package seems to be the natural successor for the functionality previously offered by RQGIS, but is, at time of writing, in an experimental stage.

Mazzia et al. [46] present a model capable of managing and globally estimating a path with a series of points that an autonomous vehicle, both land and air, can follow (Table 4, item 20). A deep learning model capable of predicting the position of the main paths from the destination was designed and trained. Extensive experimentation, with synthetic datasets and real satellite-derived images of different scenarios, used in vineyard and orchard plantations, demonstrates the effectiveness and feasibility of the methodology as a completely autonomous approach to end-to-end global path planning.

Finally, Rentadrone [43] proposes AI Agro, a set of machine learning algorithms for determining the health of various types of crops (Table 4, item 17). The provided solutions integrate data obtained from temporal, mosaic, multispectral, and thermal imaging, yielding outputs such as thermal maps and normalized vegetation index maps, which allow the identification of stress zones. The solutions can generate local calculations and plant-by-plant mapping of vegetation indices by processing images obtained by drones. According to the authors, the included algorithms can perform batch processing of photo files, JPG and TIFF images, and are capable of mapping the health status of different types of crops visible in multispectral photographs taken with drones. Thus, the software is also capable of calculating the main types of vegetation index.

3.4. Summary of Results

Results are summarized by the years of publication of the selected works, as well as by several of the respective extracted data items—defined in Section 2.4—namely, user interface, category, programming language, and keywords.

With respect to the years of publication, Figure 1 shows the number of works published yearly between January 2017 and October 2022. The upwards publication trend is highlighted with a dashed line, computed using a simple ordinary least-squares linear regression.

In turn, Figure 2 synthesizes the programming languages used in the development of the selected works. In particular, the prevalence of each language is shown in Figure 2a, and the number of different languages used in each work is displayed in Figure 2b. Note that, for the percentages shown in Figure 2a, it was defined that packages that use multiple languages contribute less to the total use of each language. For example, the OWL software [30] exclusively uses Python, and therefore contributes +1 to Python's total usage. SoftwarePilot [36], in turn, is developed with Java and Python; consequently, it contributes +0.5 to Python's total usage, and +0.5 to Java's total usage. We found that this approach represented the prevalence of programming languages more fairly.

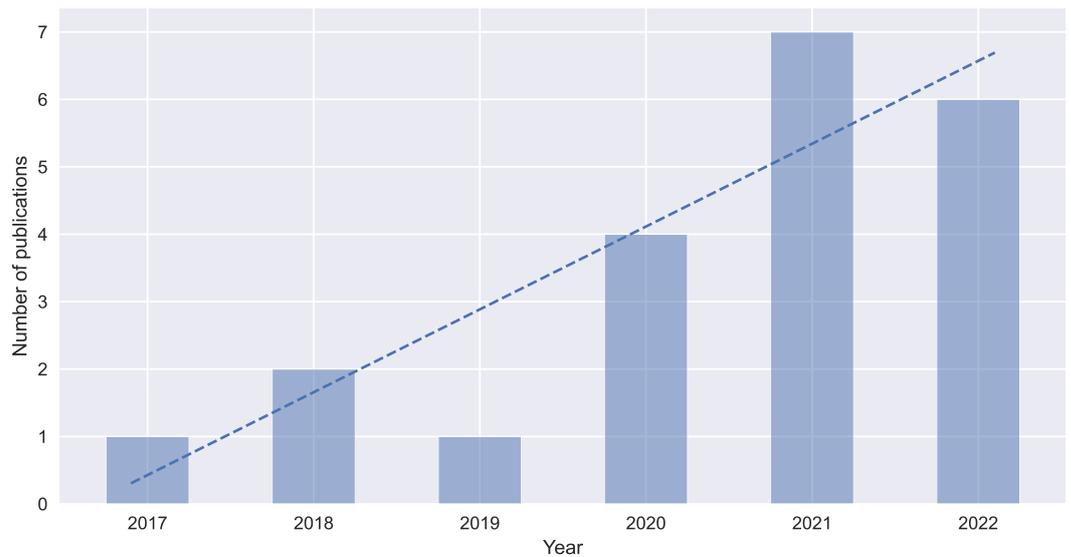


Figure 1. Number of works published each year between January 2017 and October 2022. The dashed line shows the trend, obtained with an ordinary least squares linear regression.

For summarizing the implemented user interfaces, a similar approach to the one used for the programming languages was used, as shown in Figure 3. The percentage of interface implementations, displayed in Figure 3a, was determined proportionally—similarly to the programming languages—i.e., packages having more than one interface contribute less for the totals of each interface. The number of different interfaces used in the surveyed software is highlighted in Figure 3b.

Figure 4 presents the percentage of works by category, while Figure 5 highlights the most frequent keywords found in the selected works using a word cloud—a visual representation of words arranged in a random and scattered way, with the size of each word being proportional to its frequency or importance.

The code and data used for summarizing results and creating Figures 1–4 are available at Zenodo [54].

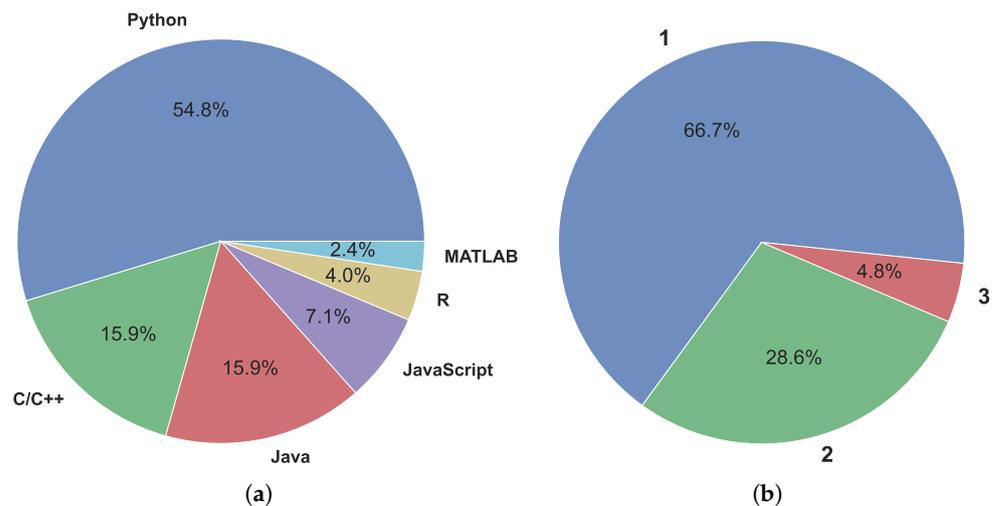


Figure 2. Programming languages used in the development of the selected software packages: (a) by percentage; and (b) by number of languages used for development of a single package. Regarding (a), it was defined that, when a package is developed in $n > 1$ languages, each language contributes $1/n$ to its total usage for all packages. Percentages do not exactly add up to 100% due to round-off errors.

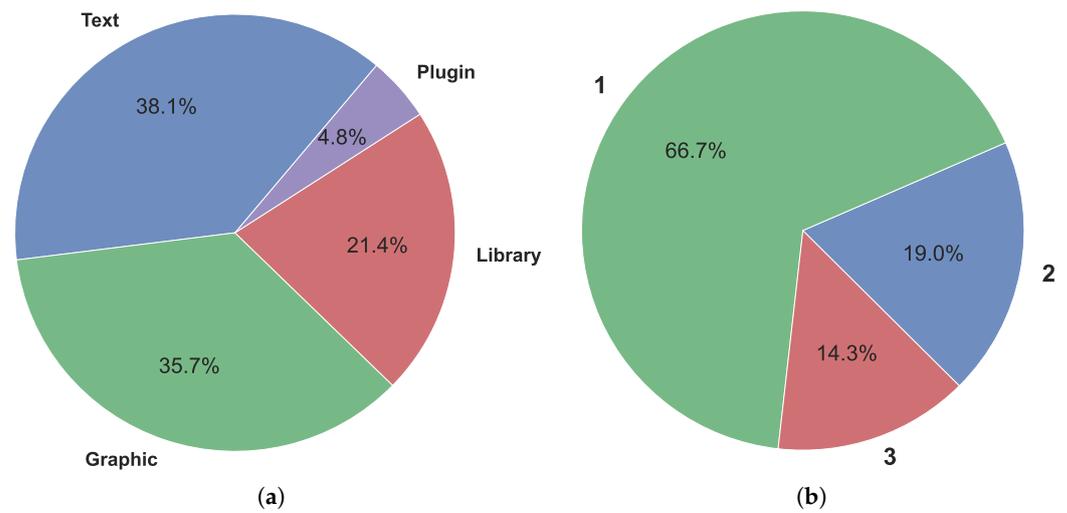


Figure 3. User interfaces implemented in the selected software packages: (a) by percentage; and (b) by number of interfaces used in a single package. Regarding (a), it was defined that, when a package implements $n > 1$ interfaces, each interface contributes $1/n$ to its total usage for all packages.

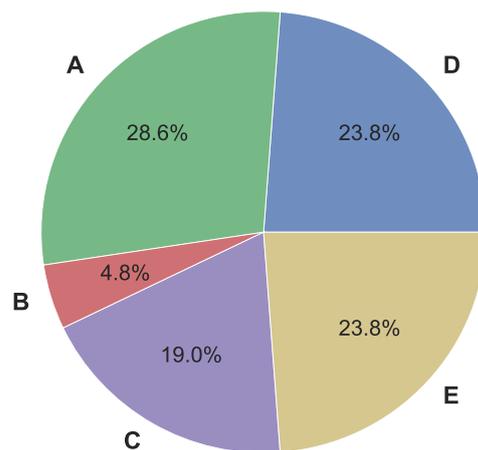


Figure 4. Representation of works divided by categories or application area, namely: A—crop and climate protection and diagnosis; B—nutrition and fertilization of crops; C—crop irrigation; D—soil management, planting, growing and harvesting crops; and, E—production management, machinery and equipment.

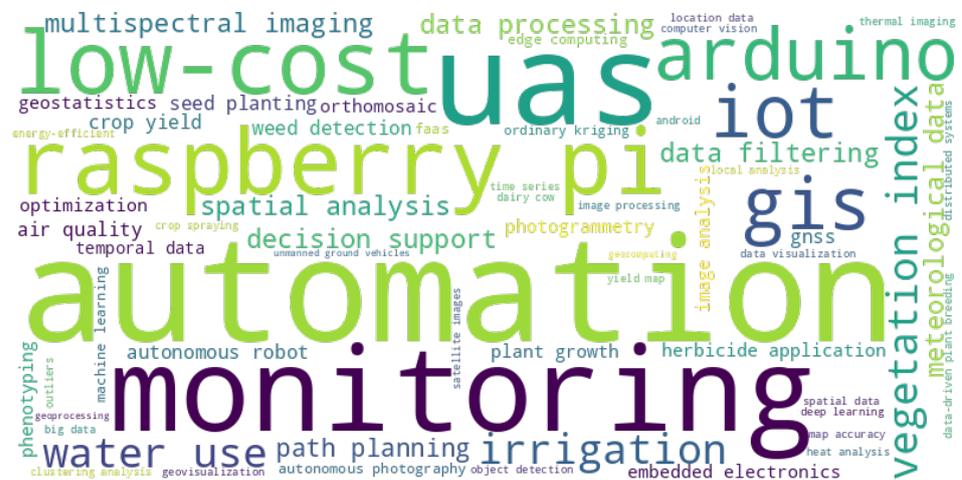


Figure 5. Word cloud highlighting the most frequent keywords found in the selected works (see Table A1).

4. Discussion

A total of 21 projects were presented and summarized in the previous section. The information reported in this discussion is necessary to understand the standards and guidelines that free and open-source software aimed at precision agriculture point to, outlining the results obtained in this research. Figure 1 displays a growth trend of works published on the theme of this research. This increasing trend suggests that free and open-source software will continue to have a significant impact on precision agriculture in the near future. Although this trend shows some clear advances in the provision of open codes in free repositories, making them available to academic communities, there are still some issues. For example, many of the repositories are poorly managed, with commits uploaded directly to GitHub, repositories containing ZIPs and executables instead of pure code, as well as incomplete documentation. Items 6, 12, 13, 16, and 19 in Table 4 are a case in point.

The prevalence of specific programming languages in precision agriculture software can provide valuable insights into the tools, resources, and skills that are important in the field, as well as emerging trends and job opportunities. While the programming language in which a specific application is developed may not be relevant for farmers following precision agriculture practices, it is nonetheless an important consideration for the development community—programmers, developers, and tinkerers. As noted in Figure 2a, the surveyed works are mostly developed in Python, and, at some distance, C/C++ and Java share the second position. While these results reflect the TIOBE index (which measures the popularity of programming languages on a monthly basis) for the period in which this review was conducted [55], it should be highlighted that Python and C/C++ in particular are very common choices for IoT solutions with Arduino and Raspberry Pi [56,57], as implemented in several of the works present in this review. Additionally, Figure 2b highlights that experience in multiple programming languages may be a factor in being involved with several open-source projects for precision agriculture.

The distribution of interfaces which end users can interact with is shown in Figure 3. As discussed in Section 2.4, four types of user interface were defined in this research: *text*, *graphic*, *library*, and *plugin*, and software packages can expose several of them simultaneously—indeed, approximately 33% of the surveyed works offer more than one interface, as highlighted in Figure 3b. Observing Figure 3a, it appears that there is a balance between the different interfaces available, namely text, graphical and library—plugins are the exception, accounting for less than 5% of the total. This data suggests that there are different types of end users with different aims, technical capabilities, and support requirements. Users who prefer graphical interfaces may be less technically proficient and/or seek applications for direct practical use in the (literal) field, while those who use text-based interfaces or libraries may be more technically apt, require more control or customization options, and/or are simply part of precision agriculture development community.

The application context by category is presented in Figure 4, in which we observe that the category of soil management, planting, growing and harvesting (category D) corresponds to 23.8% of the works presented, while the protection, diagnosis of cultivation, and climate (category A) appears with 28.6% of the tools found. In turn, the management of production, machinery and equipment (category E) is—similarly to category D—being represented by 23.8% of the tools found. The irrigation of crops (category C) appears represented with 19%, and crop nutrition and fertilization (category B) with 5%. There is an almost homogeneous distribution between most of the categories, with the exception of category B, indicating a growth space to be explored by future projects.

The word cloud displayed in Figure 5 highlights some of the core concepts underlying the selected works. Themes such as **automation**, **monitoring**, and **UAS** clearly stand out, and are the main focus in most of the surveyed software packages. The importance of hardware such as the **Raspberry Pi** and **Arduino**, and the focus on **low-cost** and/or **IoT** solutions—possibly involving **GIS**—is also emphasized in several of the selected works. Although this is a more subjective discussion, this keyword analysis is also notable due to

some important concepts which do not seem to be addressed or considered in the selected works, therefore pointing to opportunities for future research into open source precision agriculture, as is the case of security [58], interoperability [59], quality assurance [60], and the use of open data [61].

While we analyzed and discussed several data items of interest, spanning a number of free and open source projects for precision agriculture, there are some limitations to consider. Rapid literature reviews, such as the one presented in this paper, provide a quick overview of the current state of knowledge on a particular topic. These can be useful for gaining a general understanding of a subject, but have a number of limitations, namely: (1) limited depth when compared with comprehensive scoping and systematic reviews; (2) risk of bias, since the search was carried out by the first author and non-methodologically revised by the second author, which may have led to the exclusion of relevant studies, or a lack of rigor in the selection and appraisal of included studies; and (3) quality of evidence, since limiting the search to the sources mentioned in Section 2.1 may have excluded relevant works, including unpublished or gray literature. Overall, we believe this paper will be a useful tool for quickly assessing the use of free and open source technologies on precision agriculture, but these limitations should be carefully considered before drawing conclusions or making decisions based on the presented findings.

5. Conclusions

In this article, a rapid review of the literature on free and open-source technologies and software for precision agriculture was carried out. The works were selected, mapped, and presented by their information source, source code repository, year of publication, programming language(s) used in development, offered user interface(s), category within precision agriculture, and relevant keywords. A total of 21 works were found, and the associated references argued for their effectiveness in precision agriculture, especially in situations where there is more than one success case for each software and when the existence of libraries and open-source software leads to the creation of even more and better solutions. The reviewed literature shows that the use of free and open-source software can lead to excellent results in the area, and that its use is expected to grow in the future, with hopefully more software available in public repositories, fostering testing and improvements by the community involved. Analysis of the extracted data showed that Python is currently the most used programming language in these works, and that their most common themes include automation, monitoring, IoT, and low-cost solutions, highlighting technologies such as UAS and GIS in general, and the Raspberry Pi and Arduino in particular. Free and open-source software can be used to leverage these technologies, creating precision farming systems. It is freely available and can be modified and redistributed, making it an attractive option for farmers who want to customize their precision farming systems to their specific needs. Additionally, open-source software can be used to create tools that allow farmers to cost-effectively collect, analyze, and visualize data. The review carried out in this article—although having some limitations due to its rapid nature—contributes to the understanding of open-source software aimed at precision agriculture and its availability.

Author Contributions: Conceptualization, R.P.d.S., N.F. and V.R.Q.L.; methodology, R.P.d.S., N.F. and V.R.Q.L.; software, N.F.; validation, N.F., M.B. and V.R.Q.L.; investigation, R.P.d.S.; resources, M.B. and V.R.Q.L.; writing—original draft preparation, R.P.d.S. and N.F.; writing—review and editing, N.F., M.B. and V.R.Q.L.; visualization, R.P.d.S. and N.F.; supervision, N.F., M.B. and V.R.Q.L.; project administration, V.R.Q.L.; funding acquisition, M.B. and V.R.Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Fundação para a Ciência e Tecnologia under grants UIDB/05064/2020 (VALORIZA—Research Centre for Endogenous Resource Valorization) and UIDB/04111/2020 (COPELABS).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ICT	Information and communication technology
AI	Artificial intelligence
IoT	Internet of Things
ITALLIC	Integrated Tool for AgData Lat Long Imputation and Cleaning
UAS	Unmanned Aerial System
OWL	OpenWeedLocator
BovHEAT	Bovine Heat Detection and Analysis Tool

Appendix A. Keywords for Selected Works

Table A1. Five keywords for each of the works listed in Table 4. Keywords were agreed upon by the first and second authors, and obtained from one or more of the following sources, ordered by relevance: explicit keywords in referenced article, if available; extracted from abstract; extracted from full article; and/or, README file in code repository.

No.	Ref.	Keywords
1	[28]	gnss; raspberry pi; autonomous robot; seed planting; low-cost.
2	[29]	automation; arduino; air quality; iot; monitoring.
3	[30]	crop yield; low-cost; weed detection; herbicide application; image analysis.
4	[31]	low-cost; multispectral imaging; uas; embedded electronics; optimization.
5	[32]	photogrammetry; phenotyping; uas; gis; orthomosaic.
6	[33]	automation; arduino; monitoring; raspberry pi; plant growth.
7	[34]	arduino; iot; irrigation; water use; automation.
8	[35]	decision support; geostatistics; spatial analysis; gis; temporal data.
9	[36]	uas; autonomous photography; faas; machine learning; edge computing.
10	[37]	automation; ordinary kriging; clustering analysis; yield map; data filtering.
11	[38]	iot; meteorological data; irrigation; water use; android.
12	[15]	thermal imaging; uas; vegetation index; image processing; raspberry pi.
13	[39]	data-driven plant breeding; data processing; data visualization; location data; big data.
14	[40]	object detection; crop spraying; energy-efficient; uas; computer vision.
15	[41]	automation; geovisualization; time series; vegetation index; satellite images.
16	[42]	spatial data; local analysis; map accuracy; outliers; data filtering.
17	[43]	decision support; vegetation index; monitoring; multispectral imaging; uas.
18	[44]	monitoring; dairy cow; data processing; automation; heat analysis.
19	[45]	irrigation; distributed systems; meteorological data; automation; water use.
20	[46]	deep learning; unmanned ground vehicles; monitoring; path planning; automation.
21	[47]	gis; geoprocessing; spatial analysis; geocomputing; path planning.

References

1. The Free Software Foundation. What is Free Software? 2022. Available online: <https://www.gnu.org/philosophy/free-sw.html> (accessed on 6 March 2023).
2. Doering, D.; Vizzotto, M.; Bredemeier, C.; da Costa, C.; Henriques, R.; Pignaton, E.; Pereira, C. MDE-based development of a multispectral camera for precision agriculture. *IFAC-PapersOnLine* **2016**, *49*, 24–29.
3. Pearce, J.M. Emerging business models for open source hardware. *J. Open Hardw.* **2017**, *1*, 2. [CrossRef]
4. Aravind, K.; Subramanian, R.B.; Subramanian, V.S.; Srivassram, V.; Hayakawa, Y.; Pandian, S. An affordable build-your-own computer control system for electropneumatics education. In Proceedings of the 2017 Conference on Information and Communication Technology (CICT), Gwalior, India, 3–5 November 2017; pp. 1–6. [CrossRef]
5. Pedersen, S.M.; Lind, K.M. *Precision Agriculture: Technology and Economic Perspectives*; Springer: Cham, Switzerland, 2017. [CrossRef]

6. Bhat, S.A.; Huang, N.F. Big data and AI revolution in precision agriculture: Survey and challenges. *IEEE Access* **2021**, *9*, 110209–110222. [CrossRef]
7. Addicott, J.E. *The Precision Farming Revolution: Global Drivers of Local Agricultural Methods*; Palgrave Macmillan: Singapore, 2019. [CrossRef]
8. Kpienbaareh, D.; Kansanga, M.; Luginaah, I. Examining the potential of open source remote sensing for building effective decision support systems for precision agriculture in resource-poor settings. *GeoJournal* **2019**, *84*, 1481–1497. [CrossRef]
9. Niethammer, U.; Rothmund, S.; Schwaderer, U.; Zeman, J.; Joswig, M. Open source image-processing tools for low-cost UAV-based landslide investigations. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *XXXVIII-1/C22*, 161–166. [CrossRef]
10. Matilla, D.M.; Murciego, Á.L.; Bravo, D.M.J.; Mendes, A.S.; Leithardt, V.R.Q. Low cost center pivot irrigation monitoring systems based on IoT and LoRaWAN technologies. In Proceedings of the 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Trento, Italy, 4–6 November 2020; pp. 262–267. [CrossRef]
11. Sargent, A. *If You Want to Go Far, Go Together: The Future of Open Source Agtech*; Technical Report 1909; Nuffield Australia: Kyogle, NSW, Australia, 2020.
12. Raeth, P.G. Transition of soil-moisture estimation theory to practical application. *J. Eng. Comput. Innov.* **2021**, *6*, 1–10. [CrossRef]
13. Lenarduzzi, V.; Taibi, D.; Tosi, D.; Lavazza, L.; Morasca, S. Open source software evaluation, selection, and adoption: A systematic literature review. In Proceedings of the 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Portoroz, Slovenia, 26–28 August 2020; pp. 437–444. [CrossRef]
14. Cisternas, I.; Velásquez, I.; Caro, A.; Rodríguez, A. Systematic literature review of implementations of precision agriculture. *Comput. Electron. Agric.* **2020**, *176*, 105626. [CrossRef]
15. de Oca, A.M.; Flores, G. A UAS equipped with a thermal imaging system with temperature calibration for crop water stress index computation. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 15–18 June 2021; pp. 714–720. [CrossRef]
16. Martini, B.G.; Helfer, G.A.; Barbosa, J.L.V.; Espinosa Modolo, R.C.; da Silva, M.R.; de Figueiredo, R.M.; Mendes, A.S.; Silva, L.A.; Leithardt, V.R.Q. IndoorPlant: A model for intelligent services in indoor agriculture based on context histories. *Sensors* **2021**, *21*, 1631. [CrossRef]
17. dos Santos, R.P.; Beko, M.; Leithardt, V.R. Modelo de machine learning em tempo real para agricultura de precisão. In Proceedings of the Anais da XXII Escola Regional de Alto Desempenho da Região Sul, Curitiba, PR, Brazil, 18–20 April 2022; SBC: Porto Alegre, RS, Brazil, 2022; pp. 69–70. [CrossRef]
18. Cartaxo, B.; Pinto, G.; Soares, S. Rapid Reviews in Software Engineering. In *Contemporary Empirical Methods in Software Engineering*; Felderer, M., Travassos, G.H., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 357–384. [CrossRef]
19. Tricco, A.C.; Lillie, E.; Zarin, W.; O'Brien, K.K.; Colquhoun, H.; Levac, D.; Moher, D.; Peters, M.D.; Horsley, T.; Weeks, L.; et al. PRISMA extension for scoping reviews (PRISMA-ScR): Checklist and explanation. *Ann. Intern. Med.* **2018**, *169*, 467–473. [CrossRef]
20. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Technical Report EBSE-2007-01; School of Computer Science and Mathematics, Keele University: Keele, UK, 2007.
21. Cartaxo, B.; Pinto, G.; Soares, S. The Role of Rapid Reviews in Supporting Decision-Making in Software Engineering Practice. In Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, EASE'18, Christchurch, New Zealand, 28–29 June 2018; ACM: New York, NY, USA, 2018; pp. 24–34. [CrossRef]
22. Martín-Martín, A.; Thelwall, M.; Orduna-Malea, E.; Delgado López-Cózar, E. Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: A multidisciplinary comparison of coverage via citations. *Scientometrics* **2021**, *126*, 871–906. [CrossRef]
23. Exterman, D. GitLab vs GitHub—A 2022 Comparison. Incredibuild. 2021. Available online: <https://www.incredibuild.com/blog/gitlab-vs-github-comparison> (accessed on 7 March 2023).
24. Escamilla, E.; Klein, M.; Cooper, T.; Rampin, V.; Weigle, M.C.; Nelson, M.L. The Rise of GitHub in Scholarly Publications. In Proceedings of the Linking Theory and Practice of Digital Libraries, TPD L 2022, Padua, Italy, 20–23 September 2022; Silvello, G., Corcho, O., Manghi, P., Di Nunzio, G.M., Golub, K., Ferro, N., Poggi, A., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 187–200.
25. Pace, L. 4G: History, Origin, and More. History-Computer. 2022. Available online: <https://history-computer.com/4g-guide/> (accessed on 5 March 2023).
26. Alkobi, J. The Evolution of Drones: From Military to Hobby & Commercial. Percepto. 2019. Available online: <https://percepto.co/the-evolution-of-drones-from-military-to-hobby-commercial/> (accessed on 5 March 2023).
27. Kaur, R.; Chahal, K.K. Exploring factors affecting developer abandonment of open source software projects. *J. Softw. Evol. Process* **2022**, *34*, e2484. [CrossRef]
28. Rogers, H.; Fox, C. An open source seeding agri-robot. In Proceedings of the 3rd UK-RAS Conference, 2020, UKRAS '20, Lincoln, UK, 17 April 2020; pp. 48–50. [CrossRef]
29. Winkler, R. MeteoMex: Open infrastructure for networked environmental monitoring and agriculture 4.0. *PeerJ Comput. Sci.* **2021**, *7*, e343. [CrossRef] [PubMed]
30. Coleman, G.; Salter, W.; Walsh, M. OpenWeedLocator (OWL): An open-source, low-cost device for fallow weed detection. *Sci. Rep.* **2022**, *12*, 170. [CrossRef] [PubMed]

31. de Oca, A.M.; Flores, G. The AgriQ: A low-cost unmanned aerial system for precision agriculture. *Expert Syst. Appl.* **2021**, *182*, 115163. [CrossRef]
32. Wang, H.; Duan, Y.; Shi, Y.; Kato, Y.; Ninomiya, S.; Guo, W. EasyIDP: A Python package for intermediate data processing in UAV-based plant phenotyping. *Remote Sens.* **2021**, *13*, 2622. [CrossRef]
33. Arunachalam, A.; Andreasson, H. RaspberryPi-Arduino (RPA) powered smart mirrored and reconfigurable IoT facility for plant science research. *Internet Technol. Lett.* **2022**, *5*, e272. [CrossRef]
34. Carrillo-Pasiche, P.; Miranda-Gutarra, A.; Ugarte, W. HydroTi: An Irrigation System for Urban Green Areas using IoT. In Proceedings of the 2022 IEEE XXIX International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Lima, Peru, 11–13 August 2022; pp. 1–4. [CrossRef]
35. Leroux, C.; Jones, H.; Pichon, L.; Guillaume, S.; Lamour, J.; Taylor, J.; Naud, O.; Crestey, T.; Lablee, J.L.; Tisseyre, B. GeoFIS: An open source, decision-support tool for precision agriculture data. *Agriculture* **2018**, *8*, 73. [CrossRef]
36. Boubin, J.; Stewart, C. Softwarepilot: Fully autonomous aerial systems made easier. In Proceedings of the 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C), Washington, DC, USA, 17–21 August 2020; pp. 250–251. [CrossRef]
37. de Freitas Coelho, L.A.; de Queiroz, D.M.; Valente, D.S.M.; de Carvalho Pinto, F.D.A. An open source spatial analysis system for embedded systems. *Comput. Electron. Agric.* **2018**, *154*, 289–295. [CrossRef]
38. Júnior, W.M.; Valeriano, T.T.B.; de Souza Rolim, G. EVAPO: A smartphone application to estimate potential evapotranspiration using cloud gridded meteorological data from NASA-POWER system. *Comput. Electron. Agric.* **2019**, *156*, 187–192. [CrossRef]
39. Onsongo, G.; Fritsche, S.; Nguyen, T.; Belemli, A.; Thompson, J.; Silverstein, K.A. ITALLIC: A tool for identifying and correcting errors in location based plant breeding data. *Comput. Electron. Agric.* **2022**, *197*, 106947. [CrossRef]
40. Qin, Z.; Wang, W.; Dammer, K.H.; Guo, L.; Cao, Z. Ag-YOLO: A real-time low-cost detector for precise spraying with case study of palms. *Front. Plant Sci.* **2021**, *12*, 2974. [CrossRef] [PubMed]
41. Jiménez-Jiménez, S.I.; Marcial-Pablo, M.d.J.; Ojeda-Bustamante, W.; Sifuentes-Ibarra, E.; Inzunza-Ibarra, M.A.; Sánchez-Cohen, I. VICAL: Global Calculator to Estimate Vegetation Indices for Agricultural Areas with Landsat and Sentinel-2 Data. *Agronomy* **2022**, *12*, 1518. [CrossRef]
42. Maldaner, L.F.; Molin, J.P.; Spekken, M. Methodology to filter out outliers in high spatial density data to improve maps reliability. *Sci. Agric.* **2022**, *79*, e20200178. [CrossRef]
43. Rentadrone, H.B. GitHub. 2020. Available online: <https://github.com/RentadroneCL/AI-Agro> (accessed on 6 March 2023).
44. Plenio, J.L.; Bartel, A.; Madureira, A.; Cerri, R.; Heuwieser, W.; Borchardt, S. Application note: Validation of BovHEAT—An open-source analysis tool to process data from automated activity monitoring systems in dairy cattle for estrus detection. *Comput. Electron. Agric.* **2021**, *188*, 106323. [CrossRef]
45. Raeth, P.G. Moving beyond manual software-supported precision irrigation to human-supervised adaptive automation. *Afr. J. Agric. Res.* **2020**, *16*, 1548–1553. [CrossRef]
46. Mazzia, V.; Salvetti, F.; Aghi, D.; Chiaberge, M. DeepWay: A deep learning waypoint estimator for global path generation. *Comput. Electron. Agric.* **2021**, *184*, 106091. [CrossRef]
47. Muenchow, J.; Schratz, P.; Brenning, A. RQGIS: Integrating R with QGIS for Statistical Geocomputing. *R J.* **2017**, *9*, 409–428. [CrossRef]
48. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2018.
49. Gräler, B.; Pebesma, E.; Heuvelink, G. Spatio-Temporal Interpolation using gstat. *R J.* **2016**, *8*, 204–218. [CrossRef]
50. Dos Santos, R.P.; Leithardt, V.R.Q.; Beko, M. Analysis of MQTT-SN and LWM2M communication protocols for precision agriculture IoT devices. In Proceedings of the 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain, 22–25 June 2022; pp. 1–6. [CrossRef]
51. Hernández, E.S.; García, A.G.; Izquierdo, L.R.; González, J.T.; Silva, L.A.; Ovejero, R.G.; Leithardt, V.R.Q. LoRaWAN applied to agriculture: A use case for automated irrigation systems. In Proceedings of the New Trends in Disruptive Technologies, Tech Ethics and Artificial Intelligence, DiTTEt 2021, Salamanca, Spain, 15–17 September 2021; Springer International Publishing: Cham, Switzerland, 2022; Volume 1410, pp. 308–316. [CrossRef]
52. Bradski, G. The OpenCV library. *Dr. Dobb's J. Softw. Tools Prof. Program.* **2000**, *25*, 120–123.
53. Harris, C.R.; Millman, K.J.; Van Der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef] [PubMed]
54. Fachada, N. Supplementary materials for “A Rapid Review on the Use of Free and Open Source Technologies and Software Applied to Precision Agriculture Practices”. *Zenodo* **2023**. [CrossRef]
55. TIOBE Software BV. TIOBE Index. 2023. Available online: <https://www.tiobe.com/tiobe-index/> (accessed on 3 March 2023).
56. Plauska, I.; Liutkevičius, A.; Janavičiūtė, A. Performance Evaluation of C/C++, MicroPython, Rust and TinyGo Programming Languages on ESP32 Microcontroller. *Electronics* **2023**, *12*, 143. [CrossRef]
57. Mohamed, K.S.; Mohamed, K.S. IoT physical layer: Sensors, actuators, controllers and programming. In *The Era of Internet of Things: Towards a Smart World*; Springer International Publishing: Cham, Switzerland, 2019; pp. 21–47. [CrossRef]
58. Yazdinejad, A.; Zolfaghari, B.; Azmoodeh, A.; Dehghantanha, A.; Karimipour, H.; Fraser, E.; Green, A.G.; Russell, C.; Duncan, E. A review on security of smart farming and precision agriculture: Security aspects, attacks, threats and countermeasures. *Appl. Sci.* **2021**, *11*, 7518. [CrossRef]

59. Roussaki, I.; Doolin, K.; Skarmeta, A.; Routis, G.; Lopez-Morales, J.A.; Claffey, E.; Mora, M.; Martinez, J.A. Building an interoperable space for smart agriculture. *Digit. Commun. Netw.* **2023**, *9*, 183–193. [[CrossRef](#)]
60. Chrismanto, A.R.; Purwadi, J.; Wibowo, A.; Santoso, H.B.; Delima, R.; Balisa, D. Comparison Testing Functional and Usability System Mapping Land Agriculture On Platform Web and Mobile. *IAIC Trans. Sustain. Digit. Innov. (ITSDI)* **2021**, *2*, 140–157. [[CrossRef](#)]
61. Jeppesen, J.H.; Jacobsen, R.H.; Jørgensen, R.N.; Toftegaard, T.S. Towards data-driven precision agriculture using open data and open source software. *arXiv* **2022**, arXiv:2204.05582.

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.