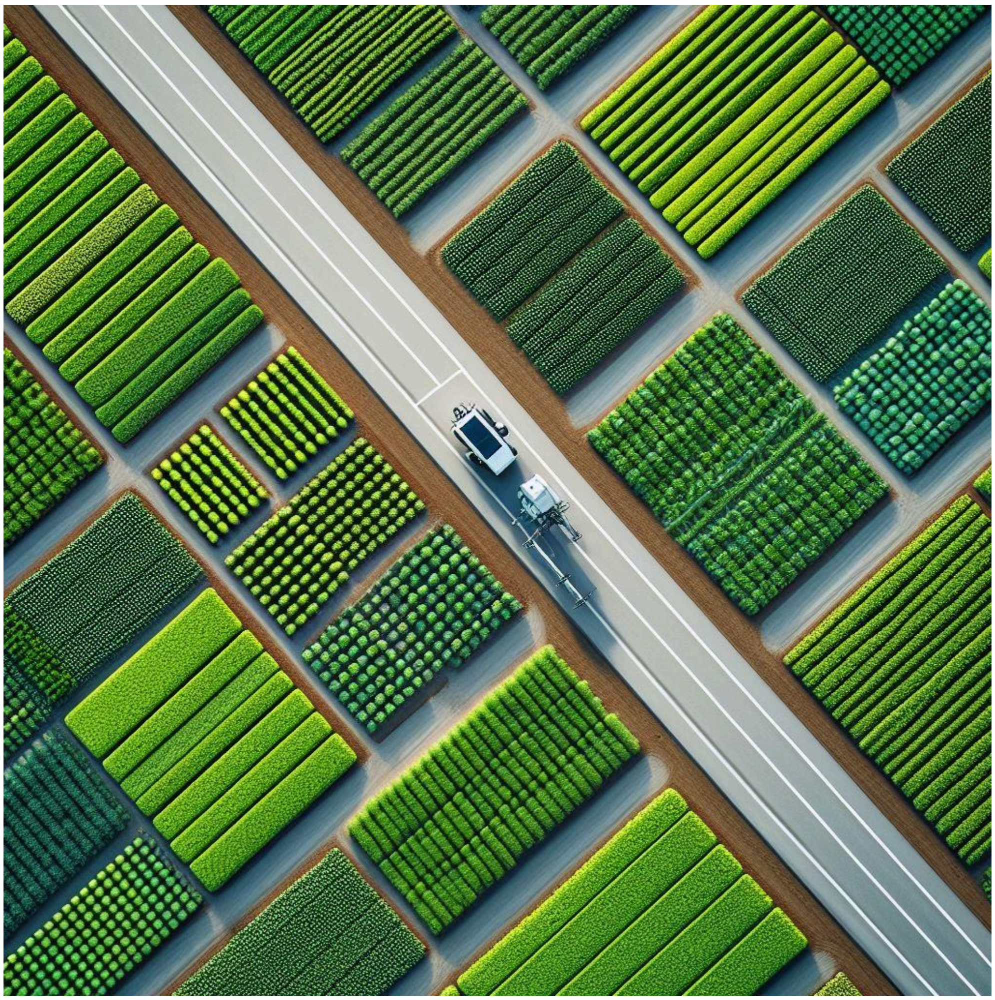
Research on the application of distributed architectures in the Smart Agriculture field



Rares Ioan Opariuc Victor Fânaru Leonard Rumeghea

14 May 2024

Table of Contents

Abstract 3
Introduction 3
Hardware 4
First approach 4
Introduction of the robotic arm 7
Final hardware design 8
Robot Operating System (ROS) and ArduPilot 9
Robot Operating System (ROS) 9
ArduPilot 10
Image Processing Algorithms 11
Data acquisition 12
Image pre-processing 12
Excess Green Index 13
Classification algorithms 14
Convolutional Neural Networks 14
Machine Learning 19
Deep Learning 19
Experimental Design 20
Description 20
Efficiency (Victor) 23
Conclusions 24
References 25

Abstract

The agriculture industry needs help fulfilling global food demand while reducing environmental effects. Weed infestations impede crop development and productivity, demanding effective and sustainable management measures. This article investigates the possibilities of robots in weed management, focusing on their use in precise weed elimination. These cutting-edge innovations use artificial intelligence, computer vision, and robots to find and eliminate weeds in non-invasive ways, minimizing the need for pesticides and physical labor. The paper uses case studies and empirical facts to illustrate the effectiveness of robotic weed management in increasing agricultural yields, improving resource consumption, and promoting sustainable practices.

Introduction

Agriculture has seen significant changes in recent years, with digitization functioning as a vital answer to issues such as climate change, increasing global population, and a lack of trained personnel. Companies like Continental are pushing advances in agricultural digitization as the world's population grows and food demand grows. According to a Continental study [11], farmers' top expectations from technology partners include lower prices for new machinery and equipment, more user-friendly technologies, upgrading or retrofitting existing devices and machinery rather than purchasing expensive new ones, and training and further education on how to use these technologies correctly.

Continental's latest trade show presentation had a particularly sustainable highlight: a solution that allows for herbicide-free weed management in fields. This system employs optical sensor technology, automotive-grade software, and artificial intelligence (AI) to properly detect weeds from crops and eliminate them with boiling water. Continental Engineering Services (CES), which specializes in building customer-specific technical solutions for businesses in a variety of sectors, designed and engineered the system.

Farmers' primary expectation from their technology partners is low-cost new machinery and equipment, with 64% of respondents globally and 71% of German farmers highlighting this. There is an interest in more user-friendly technology, with around 43% of respondents globally and 59% in Germany agreeing. Farmers also prefer to modify or adapt current gadgets and machinery rather than make costly new purchases, which is shared by around 37% of respondents globally and 48% in Germany.

According to predictions by the United Nations, the world's population will be over 8.5 billion by 2030, half a billion higher than in 2023. This population rise needs novel techniques for farming. Agriculture no longer relies only on physical labor; rather, cutting-edge technology is used. Optional alternatives for enhancing agriculture include utilizing a drone to detect the optimal time to distribute fertilizer, robotic solutions tailored to greenhouses, optimized tires, and optimized tire pressure. organic and environmentally friendly weed control. Continental and its partners are actively working further to develop this smart farming solution for widespread industrial use.

Hardware

To begin developing our robot focused on agriculture, we must first create a hardware design. This will be the basis for the physical design of our robot. In addition, we must design a reliable communication system that combines the hardware and software elements in a good manner. Many scientific articles have been reviewed, and a great deal of study has improved our understanding of these features.

The article written by Du, Yayun [1], presents a functional robot that was tested on the fields in North Dakota, U.S., and showed some statistics about weed removal, accuracy, and precision. Even the most recently marketed robots, like Ecorobotix, are still meter-sized, and the guided sprayers of today are big tractors. In order to operate along pre-planned paths, these robots are typically outfitted with highly precise but pricey navigation systems such as Multi-Global Navigation Satellite System (GNSS) or Real-Time Kinematics (RTK) GPS.

First approach

In this section, we will analyze the example robot get the main functionalities, and see some potential fixes. In the image below is the robot from the article. This is a very good starting point for our robot.

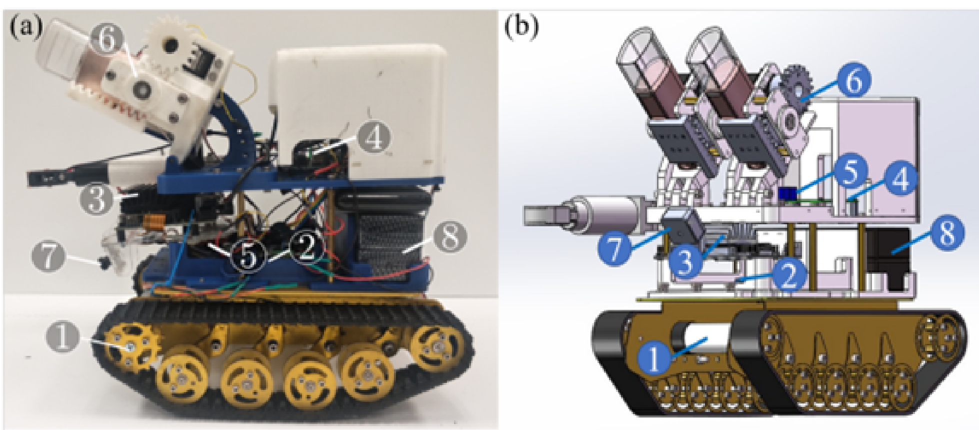


Fig. 1 Hardware layout of the agriculture robot (a - actual robot used for experiments) (b - the robot design in SOLIDWORKS computer-aided design software

For the robot from the article, they chose a **monocular camera**, because it is a far more practical and affordable solution for both weed detection and navigation. The two primary functions of monocular camera-based computer vision weed spraying robots are **crop row navigation** and **weed separation** from crops.

Due to the robot's limited battery capacity, power endurance is another issue for autonomous robots, necessitating the investigation of outside recharging techniques. These autonomous robots need to have a **charging plan**, either by wire or return to an indoor recharging station with the help of GPS.

The drivetrain consists of two 1 DC motors (labeled as 1) rated at 3.0 kg-cm torque with encoders integrated. The motors are connected to an **Adafruit Motorshield** v2 motor controller (labeled 2) which is, in turn, connected to the **NVIDIA Jetson Nano** (labeled 3) that serves as the main controller for the robot, low cost but with efficient image processing.

The encoders of the motor are connected to an **Arduino Uno. A** (labeled 4) **BNO055 Absolute Orientation Sensor** (labeled 5) is connected to **Jetson Nano**. This IMU (Inertial Measurement Unit) measures and computes the orientation of the robot during navigation.

Attached to the very front of the chassis is a **60 FPS, 136 field-of-view (FOV) wide-angle CSI camera** (labeled 7) with an unobstructed view of the path before the robot. As shown in Fig. 1, this camera is connected to the Jetson Nano with an adjustable perspective (**0-120 degrees** relative to the vertical line), 12 cm above the ground.

The robot has two **herbicide spraying units** installed close to its front. Herbicide leaking into the camera is avoided by properly arranging these sprayers in relation to the camera. The sprayers may be adjusted in angle, providing for more operational flexibility.

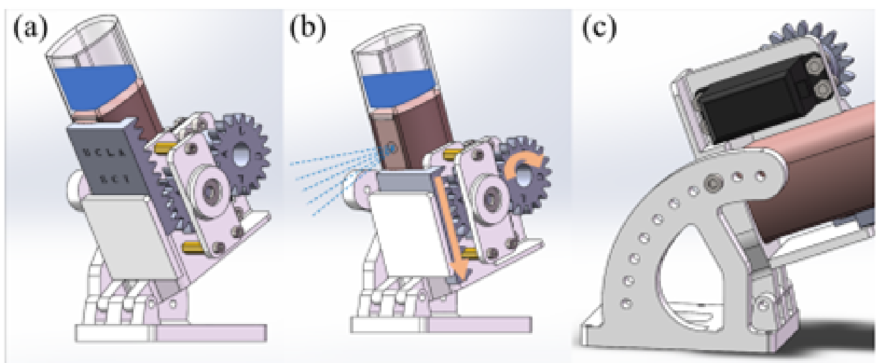


Fig. 2 Spraying system when (a) turned off and (b) turned on. (c) Close-up of sprayer holder from side view of (a))

We have determined the essential hardware parts for our agricultural robot by using the provided example as a guide. The Adafruit Motorshield v2, the motor controller of our choice, will power the robot's motor and function as an indicator for when the battery needs to be recharged.

We will use an Arduino Uno and a GPS sensor, namely the BNO055 Absolute Orientation Sensor, to locate the charging station and identify the robot's position. With this configuration, we can program the map of the field into the system and direct the robot to the charging station as needed. They will be connected to the NVIDIA Jetson Nano that serves as the main controller for the robot, low cost but with efficient image processing.

An alternative hardware and design was found in this article written by Li Xiaodong [3] which presents a ZED camera (www.stereolabs.com). It was used to gather new crop row data, and an NVidia Jetson TX2 embedded computer (www.nvidia.com) was used to execute the algorithm. Both devices were mounted on a mobile, manually remote-controlled robot platform. However, the images are taken from really far and there could be a possibility that the robot can not make the difference between a weed and a crop.

Fig. 3 Another robot example and the images taken by the camera

The system will also have a camera that takes pictures on a regular basis to track the weeds' presence. The robot will take action by turning on its pesticide spraying units when it detects weeds. In the field, our all-inclusive design guarantees effective and automatic weed management.

The robot could become stranded because it will be navigating different kinds of terrain. We will include a tracked wheel system in our design to avoid this. A tracked wheel system can more uniformly distribute the robot's weight over a greater surface area, much like the design used in tanks. This keeps the robot from sinking into soft soil and makes it easy for it to move over uneven or rugged terrain.

There may be a design flaw in our robot: the camera can only take pictures facing forward in its current configuration. If the crops get too tall, they can block the robot's view of the ground and possibly miss the existence of weeds, which could be an issue. We must investigate ways to examine the plants from above or to get a bird's-eye viewpoint of the ground in order to solve this difficulty.

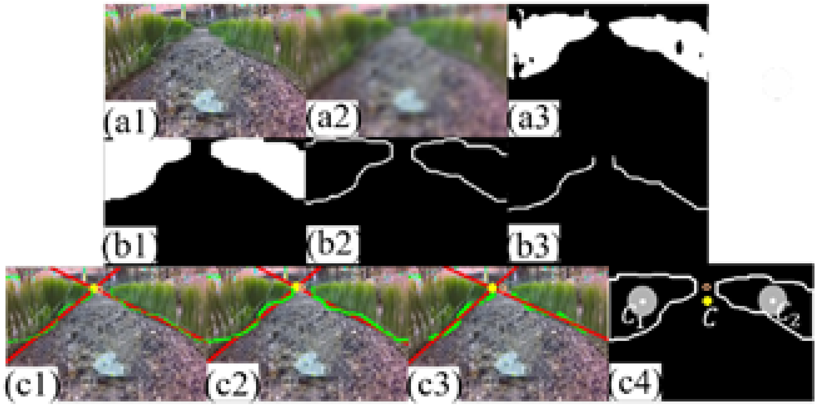


Fig. 4 What is the point of view of the robot

Introduction of the robotic arm

This article written by Kansal, Isha [2][14], presents a theoretical robot that uses the IoT for processing data but also offers a hardware design.

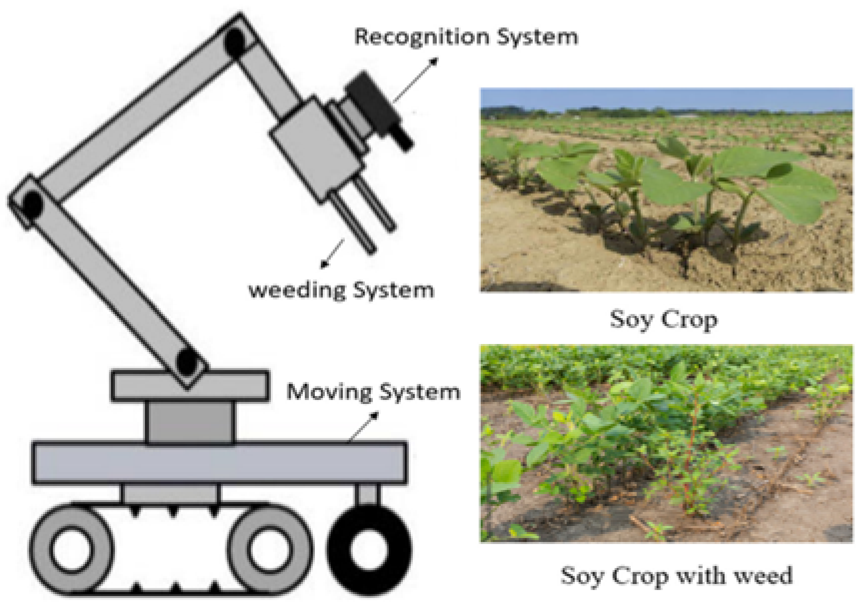


Fig. 5 The robotic arm of the robot

IoT devices [13] often generate huge volumes of data, even though there may already be millions of web units. The cloud storage system will be strained and bandwidth will be used by all of this data movement via the Internet. In large-scale Internet of Things applications, fog computation is a potentially revolutionary technology that can lower networking and processing obstacles. Fog computing is thought to provide a revolutionary approach to processing massive amounts of important and time-sensitive data.

Our original concept called for a single camera with a 120-degree field of view (FOV) to power the device. Nevertheless, in our most recent version, we have greatly improved this configuration. With three cameras now installed, our robot can see everything in a 360-degree field of view.

This improvement makes it possible for the robot to see around it and identify weeds that are growing higher than it is in the air in addition to those that are lying on the ground. Furthermore, this configuration enables the robot to stay on course as the robotic arm moves independently to remove identified weeds.

Final hardware design

Fig. 5, is the design architecture for standby and detecting a weed. Instead of one camera, it will have the 3 cameras put on a pillar which will create the 360-degree view. Besides this, it will have a robotic arm that can rotate.

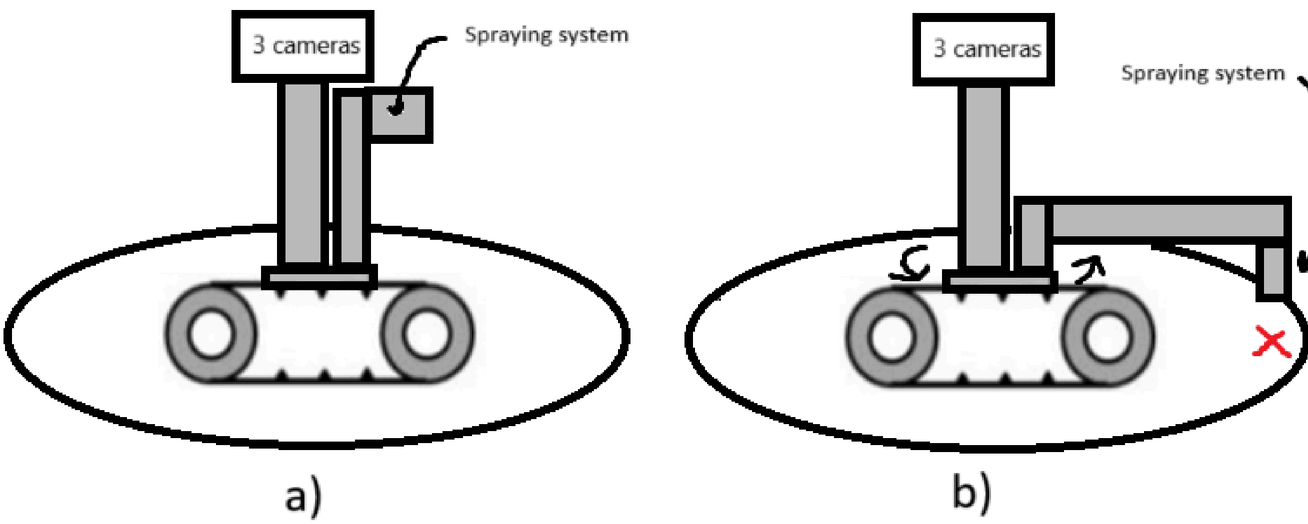


Fig. 6 The robot in standby a) and when detecting the weed

Using the Copilot, we generated an image that could be an example of our robot in the real world. (Fig. 7)

Fig. 7 An example of our robot in the real world

Robot Operating System (ROS) and ArduPilot

Robot Operating System (ROS)

The Robot Operating System, or ROS, is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. It's important to note that despite its name, ROS is not an actual operating system. It's a software framework that operates on top of a traditional operating system like Linux.

ROS was initially developed in 2007 by the Stanford Artificial Intelligence Laboratory to support the Stanford AI Robot (STAIR) project. After its initial development, the system was further supported and improved by Willow Garage, a robotics research lab and technology incubator. Today, ROS is maintained by the Open Source Robotics Foundation.

ROS provides a robust framework for building complex robot applications. It provides a communication layer over the host operating system, allowing different parts of a robot to interact in a flexible and unified manner. This includes everything from low-level device drivers to high-level algorithms for tasks like navigation and perception. ROS also provides tools for visualizing what the robot is doing, debugging robot behavior, and managing data.

ROS is used in a wide variety of robotic applications. For example, it's used in the PR2 robot, which was developed by Willow Garage. PR2 is a general-purpose robot that's used in research and for tasks like folding laundry or fetching drinks. ROS is also used in the TurtleBot, which is a low-cost, personal robot kit with open-source software. In addition, ROS has been used in numerous robotic competitions, including the DARPA Robotics Challenge.

In the field of agriculture, ROS has been used to develop autonomous farming robots. For example, a company called Blue River Technology developed a robot called LettuceBot that uses machine learning, computer vision, and ROS to selectively spray weeds on lettuce farms. Another example is the AgBot II, developed by researchers at Queensland University of Technology in Australia, which uses ROS to navigate fields, identify weeds and pests, and take action to eliminate them. - [Blue River Technology Uses Robots, Artificial Intelligence to Kill Weeds (agriculture.com)]

ROS offers a wide range of functionalities such as hardware abstraction, device drivers, inter-process communication over multiple machines, and tools for testing and visualization, among others.

One of the main features of ROS is its communication system. This system allows developers to design intricate software without needing to understand the specifics of how certain hardware operates. ROS achieves this by connecting a network of processes, known as nodes, to a central hub. These nodes can operate on multiple devices and connect to the hub in various ways.

The network is primarily created by offering services that can be requested, or by defining publisher/subscriber connections between nodes. Both methods communicate using specific message types. While some types are provided by the core packages, individual packages can also define their own message types.

This system allows developers to build a complex system by linking existing solutions for smaller problems. The implementation of the system offers several advantages:

- It allows for the replacement of components with similar interfaces on the fly, eliminating the need to stop the system for various changes.
- It enables the outputs of multiple components to be multiplexed into one input for another component, facilitating the parallel resolution of various problems.
- It allows components written in different programming languages to be connected by simply implementing the appropriate connectors to the messaging system. This makes it easy to develop software by linking existing modules from various developers.
- It allows nodes to be created over a network of devices, without worrying about where the code is run and implementing Interprocess Communication (IPC) and Remote Procedure Call (RPC) systems.
- It allows for direct connection to feeds on demand from remote hardware without writing any extra code, by employing the previous two points.

ArduPilot

ArduPilot is an open-source autopilot system that supports a wide variety of vehicle types, including multi-copters, traditional helicopters, fixed-wing aircraft, boats, submarines, rovers, and more. It's a comprehensive suite of tools suitable for almost any vehicle and application. It provides advanced functions such as autonomous flight, data logging, real-time viewing of flight data, and more.

The ArduPilot project was started in 2007 by members of the DIY Drones community. Its first code repository was created in 2009. Over the years, it has grown into a robust and versatile autopilot system with a large community of contributors.

ArduPilot has been used in various agricultural applications:

- **Crop Spraying**: ArduPilot supports a feature for crop spraying. This feature allows an autopilot connected to a PWM-operated pump and (optionally) spinner to control the rate of flow of liquid fertilizer based on the vehicle speed. This can be particularly useful for large-scale farms where manual spraying would be time-consuming and labor-intensive.
- **Precision Agriculture**: ArduPilot can be used in precision agriculture, where drones can monitor crop health using various sensors and apply treatments only where needed. This can lead to significant savings in water, fertilizer, and other inputs, and can also reduce environmental impact.
- **Agricultural Drones for Areca Nut Farms**: In a specific example, a quadcopter drone with a payload capacity of 500 g was developed with a Pixhawk Flight Controller (which uses ArduPilot). This drone was used to semi-autonomously spray fungicide solution on areca nut trees. This helped farmers avoid the dangerous and labor-intensive task of manually spraying the trees.
- **FlytOS**: FlytOS is a software framework that provides Drone APIs and SDKs for building high-level drone applications such as aerial delivery, precision agriculture, surveys, photography, industrial inspections, and disaster management. It is designed to enable drone developers to build advanced drone applications using its open APIs and is compatible with ArduPilot.

ArduPilot is a platform that facilitates the development and utilization of reliable, autonomous, unmanned vehicle systems for the benefit of everyone. It offers a wide range of tools that can be adapted to nearly any vehicle or application. As an open-source initiative, it continually evolves, driven by feedback from its extensive user community.

While ArduPilot doesn't produce any hardware, its firmware is compatible with a broad array of hardware used to control various types of unmanned vehicles. When paired with ground control software, vehicles running on ArduPilot can perform sophisticated functions, including real-time interaction with operators. ArduPilot is backed by a large online community that provides support by answering queries, solving problems, and sharing solutions.

Image Processing Algorithms

Weeds obstruct irrigation, interrupt pesticide application, and propagate diseases, reducing the growth of crops and agricultural costs. Traditional approaches like chemical and physical eradication result in overuse and labor-intensive cultivation. Site-specific weed Management (SSWM) solves these concerns by identifying weed areas and performing spot spraying or mechanical removal. SSWM may be divided into two categories: prescription map-based approaches and real-time monitoring methods, which identify and control weeds by spraying weedicide on the spot. This method greatly cuts production expenses while improving crop quality.

Data acquisition

Mobile robots or UAVs are frequently used to acquire photographs of plantations with multispectral or hyperspectral sensors. Multispectral cameras catch many bands, including R, G, B, and NIR. Hyperspectral sensors, on the other hand, may give hundreds or thousands of bands, as shown by Lu et al. (2019). Hyperspectral sensors are characterized as point-scan, area-scan, or line-scan based on how they scan 3D hyperspectral cubes. Area-scanning cameras acquire a 2-D grayscale image of a single band, whereas point-scanning cameras catch a single point throughout a two-dimensional area. Line-scanning cameras produce a two-dimensional picture, expanding point-scanning. However, hyperspectral pictures are in little supply and perform similarly to multispectral photos in terms of estimating vegetation features.

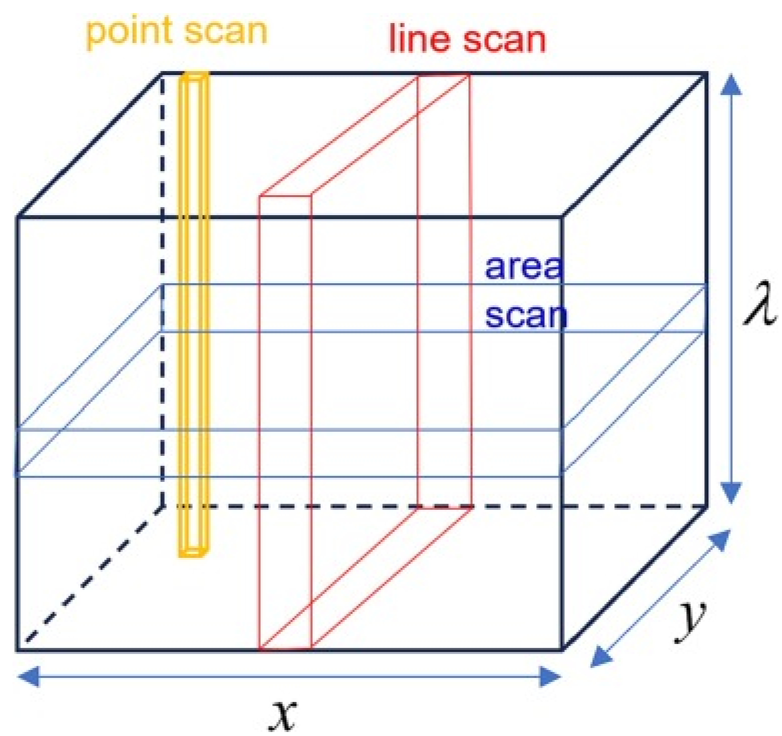


Fig. 8 Examples of point-scan, area-scan, and line-scan

Image pre-processing

Imagine enhancement is essential for enhancing picture quality during feature extraction and classification tasks. Shadows, stones, water, and dirt are examples of disruptions in manually captured photographs. These pictures are often multispectral, having four channels: red (630-690 nm), green (510-580 nm), blue (450-580 nm), mid-infrared (300-500 nm), and near-infrared (700-1100 nm). Vegetation Indices (VIs) distinguish vegetation from disturbances since each component has a unique reflectance to a certain band. VIs are classified as either without infrared channels (RGB) or with infrared channels (RGB + NIR/MIR). There are two types of VIs: first-generation and second-generation. First-generation VIs are derived empirically, with no consideration for atmospheric impacts, soil brightness, or soil color. Second-generation VIs are developed using mathematical and physical reasoning, logical experimentation, and simulation.

VIs can be classified as slope-based, distance-based, orthogonal transformation, or Red Edge Inflection Point (REIP). Slope-based approaches employ the ratio of NIR to R, whereas distance-based methods use the perpendicular distance between each pixel point and the soil line. Orthogonal transformation generates new uncorrelated bands, but REIP detects the red edge, a rapid shift in reflectance seen in the green plant spectrum at the transition between visible and near-infrared wavelengths.

Excess Green Index

The **Ex**cess **G**reen **I**ndex (**ExGI**), developed by Gée et al. (2008) and Woebbecke et al. (1995), is critical for remote sensing and precision agriculture. It uses satellite photography to assess the health and vitality of plants, as well as to identify chlorophyll and crop stress. This statistic supports decision-making in agriculture and environmental management. Gée et al. (2008) improved its use, stressing physiological insights, while Woebbecke et al. (1995) presented ExGI as a growth indicator. Their contributions illustrate ExGI's long-term relevance in changing agricultural research and promoting sustainable land management globally.

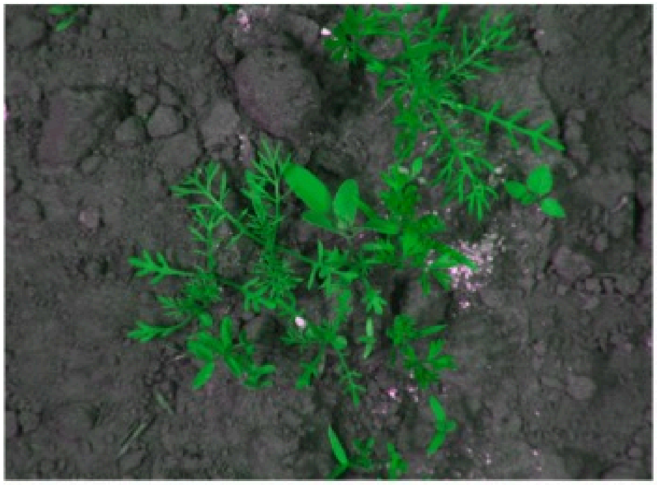
- $ExGI = 2g - r - b$

Where: $g = \frac{G_n}{R_n+G_n+B_n}$, $r = \frac{R_n}{R_n+G_n+B_n}$, $b = \frac{B_n}{R_n+G_n+B_n}$

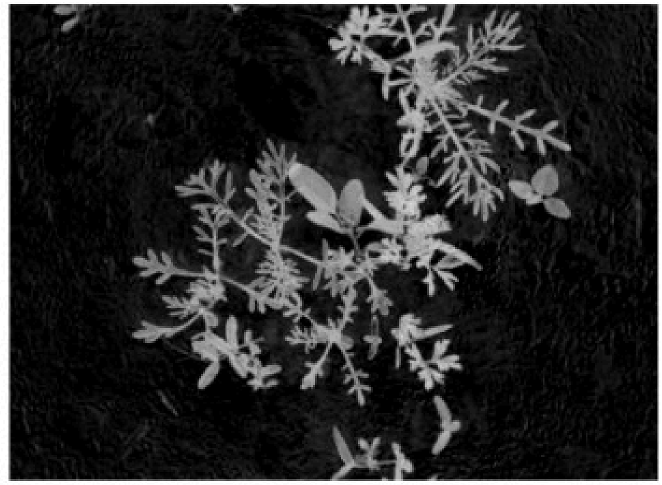
- R_n, G_n, B_n are the normalized RGB coordinates ranging between 0 and 1. The normalized coordinates are given by

$$G_n = \frac{G}{G_m}, \qquad R_n = \frac{R}{R_m}, \qquad B_n = \frac{B}{B_m}$$

- R, G, and B are the true coordinates, and R_m, G_m, and B_m are the maximum true coordinates. Note that $R_m = G_m = B_m = 255$ for 24-bit color images. The figure below depicts the photos before and after ExGI preprocessing.



(a) Original image



(b) ExGI image

Fig. 9 The original image and the image after the algorithm

Classification algorithms

Convolutional Neural Networks

Convolutional Neural Networks (**CNN**) have been used for weed classification and recognition, however, they require a significant number of manually labeled photos for model training, making field weed detection difficult due to the time-consuming data labeling. Below we will check a CNN-based semi-supervised learning strategy for weed and crop recognition using minimal labeling data. The method uses a huge amount of unlabeled data to train on a small amount of labeled data, hence enhancing learning accuracy. The GCN feature is extracted from all weed CNN features, and the GCN graph is created using weed features and Euclidean distances. The graph propagates characteristics between labeled and unlabeled graph vertices using graph convolution layers, allowing unlabeled vertices to update their properties. The final characteristics are put into the labeled vertices' softmax classifier to predict the label. The approach beats other cutting-edge algorithms and has been made accessible in weed recognition benchmark datasets.

Powerful CNNs have been utilized to extract powerful weed characteristics that are more representative and insensitive to lighting or soil backdrop. Recent research indicates that such characteristics can increase weed detection accuracy. For example, a CNN model was trained and validated to identify weeds in winter wheat fields with significant leaf occlusion, with a recall of 46.3% and 86.6%. Similarly, a k-means unsupervised feature learning method was presented to increase weed detection accuracy. Recent research has also employed the Inception-v3 and ResNet-50 deep learning models to obtain average weed classification accuracy of 95.1% and 95.7%, respectively. However, most current deep learning algorithms concentrate on specific datasets and ignore the impact of light and soil backdrop.

CNN networks are used to extract CNN features due to their strong feature extraction ability. They use original images as input and convolution of layers to acquire features. Three types of CNN features are extracted from AlexNet, VGG16, and ResNet-101. Each network is fine-tuned with the dataset, and the final CNN feature set is used to construct the GCN graph.

These three approaches (AlexNet, VGG16, and ResNet-101) produced the following outcomes on four separate databases containing three different species of plants (corn, lettuce, and radish).

Fig. 10 Database image examples

Table 1.
Description of the datasets that were utilized in the research investigations.

Dataset	Number of Images	Description
Corn	Train: 4200 (840 corn and 3360 weed) Test: 1800 (360 corn and 1440 weed)	Major variations in illumination and soil background
Lettuce	Train: 560 (350 lettuce and 210 weed) Test: 240 (150 lettuce and 90 weed)	Major variations in illumination
Radish	Train: 280 (140 radishes and 140 weed) Test: 120 (60 radishes and 60 weed)	Minor variations in illumination
Mixed	Train: 5040 (840 corn, 350 lettuce, 140 radishes, and 3710 weed) Test: 2160 (360 corn, 150 lettuce, 60 radishes, and 1590 weed)	Multiclass crops have minor variations in illumination

Table 2.
Training parameters for networks.

Parameters	Value	Remarks
Batch size	32	Compute the gradient at a reasonable speed
Training epochs	20,000	Make sure the network is well-trained
Learning rate	0.00001	Learn a more optimal set of weights
Initial weights	Random	No influence of pre-trained weights

The testing results for all techniques were put together into confusion matrices, which included true positive (**tp**), true negative (**tn**), false positive (**fp**), and false negative (**fn**). In this context, tp represents properly recognized weeds; tn represents correctly identified crops; fp represents crops wrongly identified as weeds; and fn represents weeds incorrectly identified as crops. To assess performance, five commonly utilized metrics were calculated: accuracy (ACC), precision, recall, specificity (SPC), and F1 score.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad Precision = \frac{tp}{tp + fp} \quad Recall = \frac{tp}{tp + fn}$$

$$Specificity = \frac{tn}{tn + fp} \quad F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Table 3.
Performance comparison of the approaches on the corn weed dataset (%).

Methods	ACC	Precision	Recall	SPC	F1	Time (s)
AlexNet	93	97.9	97.76	91.98	97.83	0.18
VGG16	95.8	98.63	98.92	94.7	98.78	0.21
ResNet-101	96.5	98.89	98.83	95.3	98.81	0.28

Table 4.
Performance comparison of the approaches on the lettuce weed dataset (%).

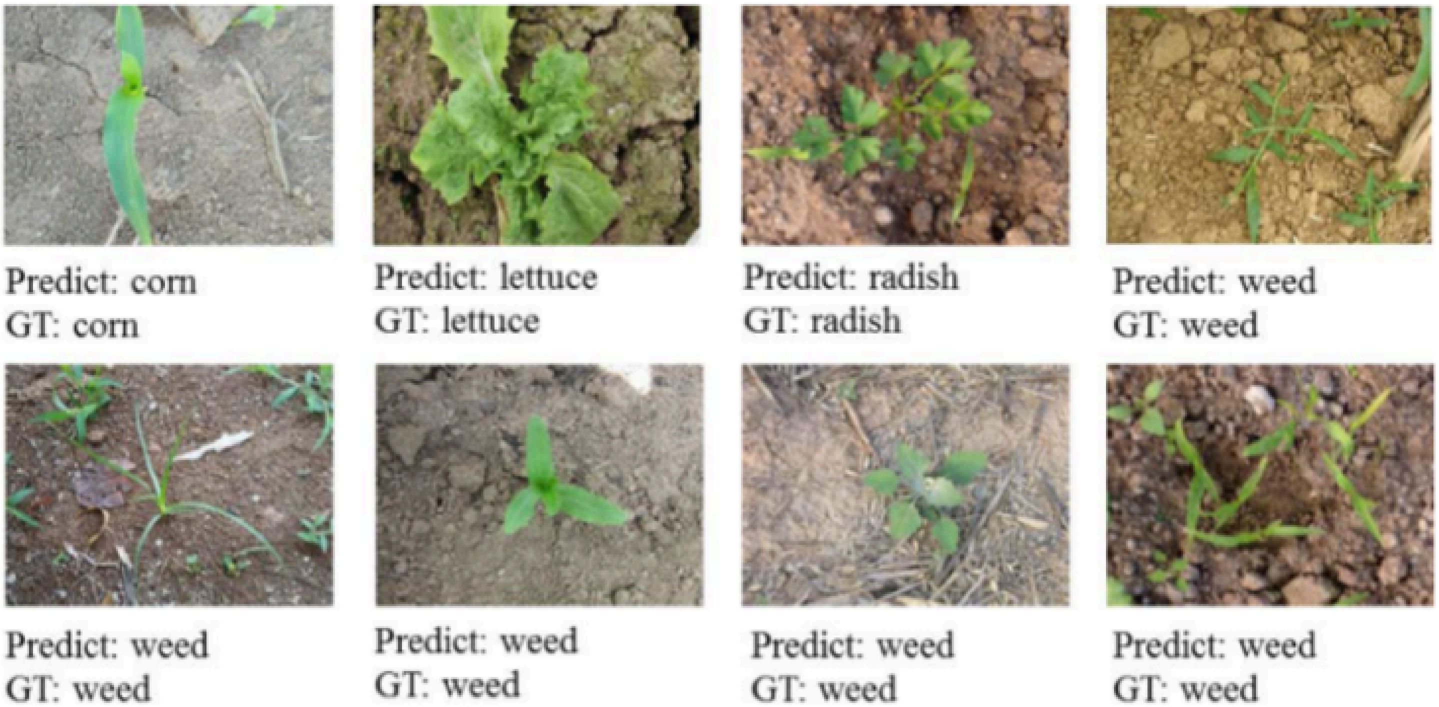
Methods	ACC	Precision	Recall	SPC	F1	Time (s)
AlexNet	96.65	93.59	97.33	96	95.42	0.16
VGG16	97.61	95.77	98	97.4	96.87	0.19
ResNet-101	98.25	96.73	98.83	98	97.52	0.20

Table 5.
Performance comparison of the approaches on the radish weed dataset (%).

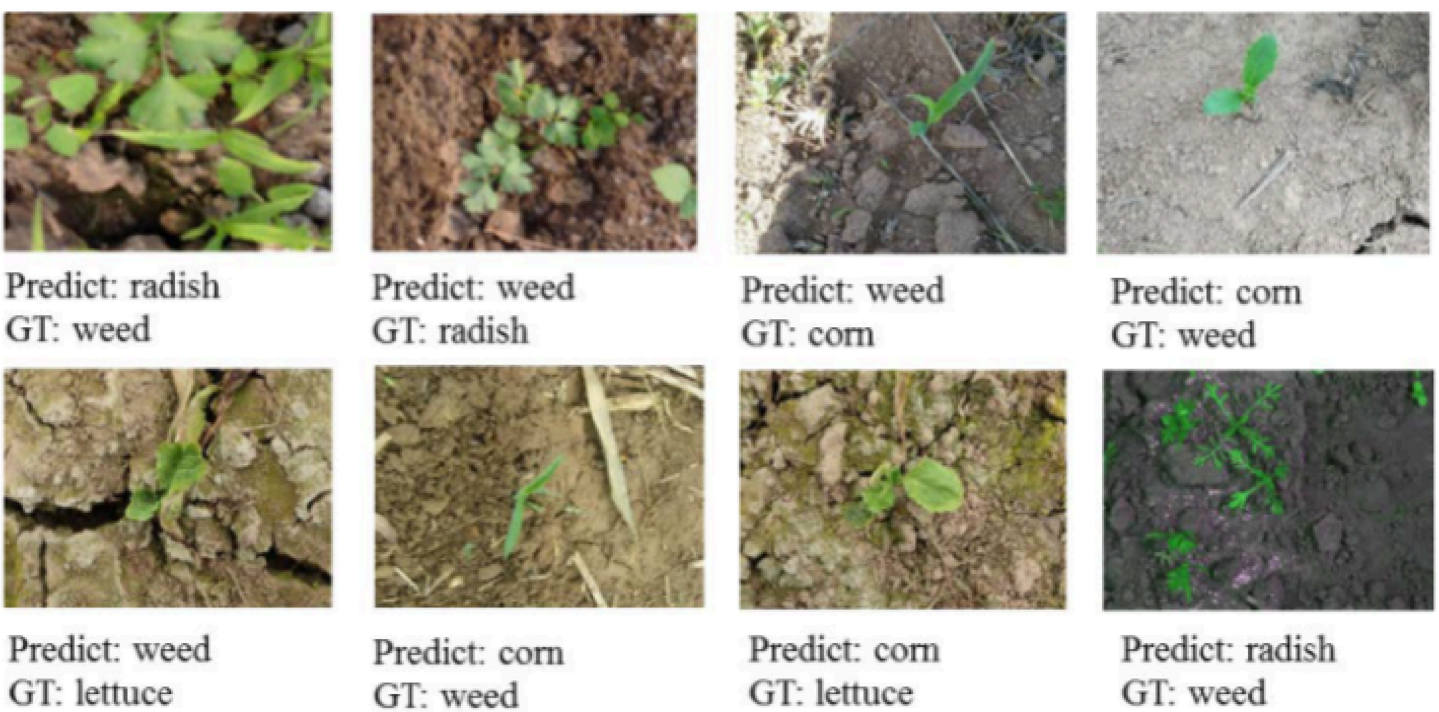
Methods	ACC	Precision	Recall	SPC	F1	Time (s)
AlexNet	95.85	95	96.7	95	95.89	0.15
VGG16	97.54	97.83	97.86	97.22	97.55	0.18
ResNet-101	97.95	97.83	98.08	97.82	97.95	0.21

Table 6.
Performance comparison of the approaches on the mixed dataset (%).

Methods	ACC	Precision	Recall	SPC	F1	Time (s)
AlexNet	90.3	95.65	95.53	90.17	95.58	0.19
VGG16	91.37	96.03	95.98	91.03	96.05	0.23
ResNet-101	94.95	98.37	97.58	94.35	96.9	0.29



(a) Correct recognition examples



(b) False recognition examples

Fig. 11 Results obtained after training on the mixed database.

Machine Learning

Machine learning-based classifiers are used to improve precision agricultural systems by merging many features. Traditional approaches, which need handmade features and depend on image quality, pre-processing, and learning algorithms, have advantages such as small sample sizes and low graphics processing unit requirements, making agricultural machinery and equipment more affordable. These approaches, however, may be inefficient in real-time applications due to continuous soil conditions and light.

Crop/weed discrimination tasks are widely performed using a variety of machine learning classification methods, including K Nearest Neighbor (**KNN**), Artificial Neural Networks (**ANNs**), decision trees, and Support Vector Machines. KNN is a non-parametric method that measures the distance between a new data sample and its neighbors before assigning it to the class with the smallest distance. ANNs are intended to emulate the principles of the human brain, consisting of layers of connected artificial neurons, with the perceptron serving as the essential structure.

SVM is a linear classifier that constructs a hyperplane to separate two classes optimally. Error-Correcting Output Coding (**ECOC**) can be utilized for multiclass classification issues, while kernel techniques such as Radial Basis Function (**RBF**) can be used for non-linear data distribution. A decision tree is a supervised machine learning method with a tree-like structure that can do classification and regression. Its components are root nodes, splitting nodes, branch/sub-tree nodes, decision nodes, and terminal nodes. Various techniques have been utilized when building the decision tree, with the entropy measure being used to choose the most beneficial characteristic to split data at each node.

Deep Learning

The methods of deep learning (**DL**), which utilize huge amounts of photos for recognition and classification, have grown in popularity due to their capacity to extract multidimensional and spatially important information about crops and weeds using Convolutional Neural Networks (**CNNs**). CNNs are made up of a stack of convolutional layers that filter input, simplify it for improved processing, and extract characteristics. DL algorithms outperform traditional learning techniques in applications such as picture classification and object identification and recognition.

Weed-crop identification strategies are divided into two categories: pixel-level (or pixel-wise) and image-wise classifications. Pixel-wise categorization involves separating each pixel in a picture and labeling them as weed or crop pixels. Researchers employed pixel-wise solutions to decrease manual pixel annotation efforts while achieving a per-class **average accuracy of 91.3%**.

McCool et al. proposed using Deep CNN (**DCNN**) to train lightweight models for pixel-level crop-weed classification. They proposed three phases for training the classification model: applying a pre-trained DCNN model, creating a lightweight DCNN student model, and merging several lightweight models to get better classification performance. The lightweight models combined to produce an

88.9% classification rate, which was quicker than the cutting-edge DCNN model's **93.9% accuracy at 0.12 fps**.

Haug and Ostermann (2015) suggested a plant classification model that does not use segmentation and instead classifies pixels using a Random Forest classifier using statistical features. This model overcame segmentation issues such as overlapping and complicated backdrops, obtaining an **average classification accuracy of 85.9%** in the CWFID benchmark dataset for binary classification.

U-Net is a famous deep learning (DL) technique for segmentation, notably in medical representation. In recent experiments, models like InceptionV3 have demonstrated great accuracy in differentiating between bell peppers and weeds, ranging from **94.5%** to **97.7%**. These models do not use segmentation, but instead identify vegetative objects as weeds or crops.

Experimental Design

Description

The design we propose encompasses an intelligent robotic system, specifically engineered to assist agricultural practitioners in their daily tasks by managing the extermination of weeds through the application of pesticides.[12]

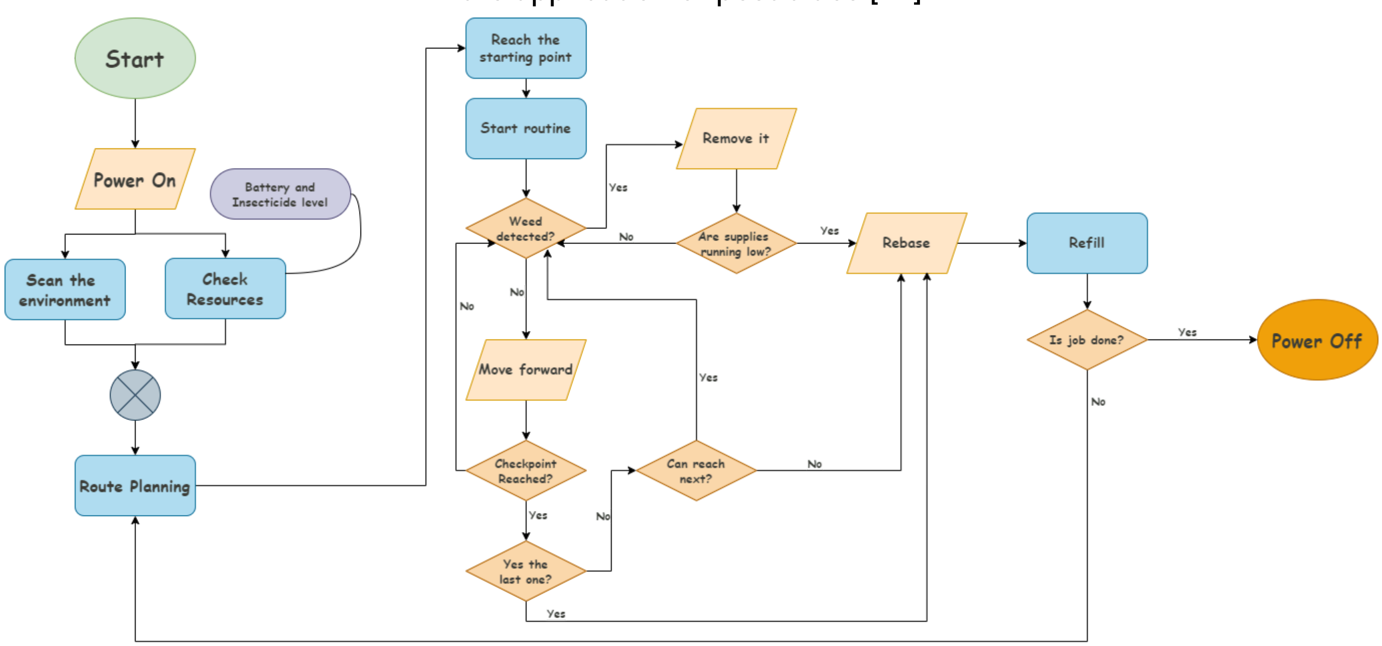


Fig. 12 The diagram of our robot

The robotic system is comprised of the following components:

- A pair of track mechanisms are designed to facilitate movement and overcome potential obstacles that may arise during the robot's operation.

- A 360-degree camera, configured to detect weeds within a specified range and pinpoint the location for pesticide application.
- An extendable and mobile arm, tasked with the precise delivery of the pesticide to the base of the weed.
- A pesticide reservoir serves as the storage unit for the pesticide, enabling the robot to eliminate multiple weeds during a single crop inspection.
- A rechargeable battery provides the necessary power for the robot's operation.

In addition to the components of the robot, we have also designed a docking station where the robot can replenish its energy and pesticide supply. The docking station consists of:

- A wireless charging pad, upon which the robot is positioned for recharging.
- A larger pesticide reservoir, designed to refill the robot's reservoir as needed.

Prior to the deployment of the robot, the client is required to utilize auxiliary software to conduct an aerial scan of the crops. This process facilitates the creation of a map detailing potential routes, which is subsequently integrated into the robot's system.

Utilizing this map, the robot is capable of conducting a systematic inspection of the crops, thereby ensuring an efficient use of time and resources. The robot is programmed to always commence its operation from a consistent starting point and is capable of memorizing the areas inspected within a day. This information is utilized to determine the most efficient route back to the point where it ceased operation at the end of the day or prior to returning for recharging or pesticide refilling

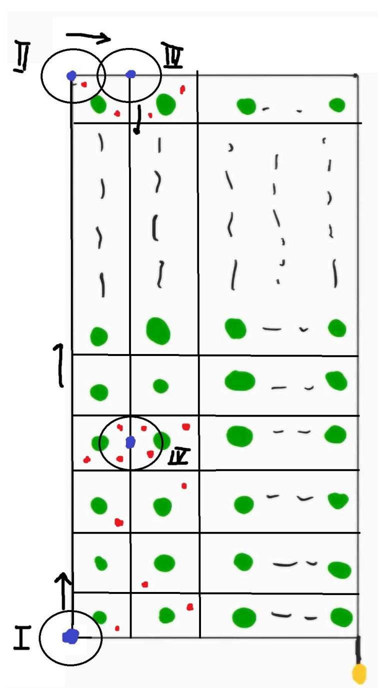


Fig. 13 How the robot will walk on a field

The typical operational flow of the robot is as follows: Upon departure from the docking station, the robot utilizes the predefined map to inspect the crops. It commences from the uppermost left corner of the crops and proceeds from left to right horizontally. Upon reaching the boundary of the map, it initiates the inspection of the next row of crops.

Upon detection of one or multiple weeds using the 360-degree camera, the robot halts and applies pesticide to all detected weeds, ensuring that no weed remains untreated within its path. Once the camera confirms that no untreated weed remains, the robot resumes its course.

During the crop inspection process, on the map, there are some checkpoints where the robot will calculate the energy required to return to the docking station for recharging. This calculation is based on the fastest route back to the docking station, irrespective of whether the route has been previously inspected or not, as the robot ceases all work during its return journey. If the energy required to reach the docking station equals the robot's current energy level, the robot will return to the docking station, saving the coordinates of its stopping point using GPS. This approach is also employed for refilling the pesticide tank.

Upon recharging or refilling, the robot calculates the fastest route back to its previous stopping point and proceeds there without applying pesticide to any weeds encountered en route. Upon arrival, it resumes its task, ensuring that no weeds are left untreated.

At the end of the day, the robot saves the coordinates of its stopping point and returns to the docking station for recharging and refilling, enabling it to resume work from the same point the following day.

Upon the completion of a 100% crop inspection, the robot is programmed to return to the docking station for a full recharge and pesticide refill. Concurrently, it resets the map, thereby preparing itself for a fresh start when it is next scheduled to commence operation. This ensures that the robot is always ready for efficient and effective operation, starting anew with each cycle.

Efficiency

Our agricultural robotic system is designed with efficiency at its core. It incorporates several components and strategies to ensure optimal performance in weed extermination tasks. Here are the key factors that represent the efficiency of our robot:

1. Coverage Efficiency

Our robot is engineered to cover a large area of crops in a given time period. This high coverage efficiency ensures that no area is left untreated, contributing to a healthier crop yield.

2. Energy Efficiency

The robot is designed to do more work with less energy. It uses an optimized route planning algorithm to minimize travel distance and time, thereby reducing energy consumption.

3. Pesticide Efficiency

Our robot applies pesticides with precision, ensuring that each weed is effectively treated with the least amount of pesticide. This not only conserves resources but also minimizes environmental impact.

4. Time Efficiency

Time is a crucial factor in agricultural practices. Our robot is designed to inspect and treat a large area in less time. It uses a systematic approach to inspect the crops, ensuring that no time is wasted.

To further enhance the efficiency of our robot, we have incorporated several strategies:

- **Optimized Route Planning**: Our robot uses advanced algorithms to plan its route, minimizing travel distance and time.
- **Improved Weed Detection**: The robot's 360-degree camera is enhanced with AI algorithms to better identify and locate weeds.
- **Upgraded Battery Technology**: We use a battery with higher energy density and faster charging speed to increase the robot's working time.
- **Refined Pesticide Application**: The precision of the mobile arm and the control of pesticide output are improved to ensure the effective treatment of each weed with the least amount of pesticide.
- **Automated Map Updating**: The robot can automatically update its map based on real-time changes in the crop field, helping it adapt to new situations and maintain high coverage efficiency.

In conclusion, our agricultural robotic system is a model of efficiency, designed to assist agricultural practitioners in their daily tasks in the most effective and resource-friendly way possible.

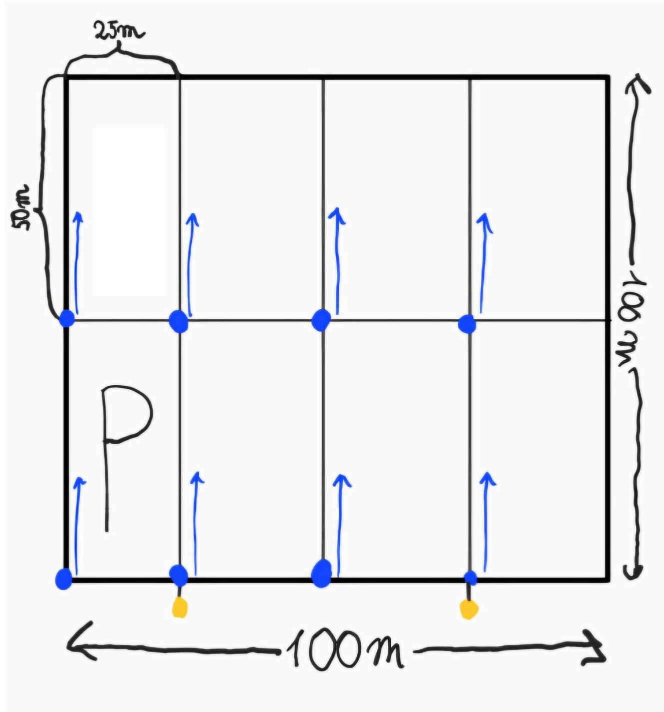


Fig. 14 An example of 1 hectare of corn

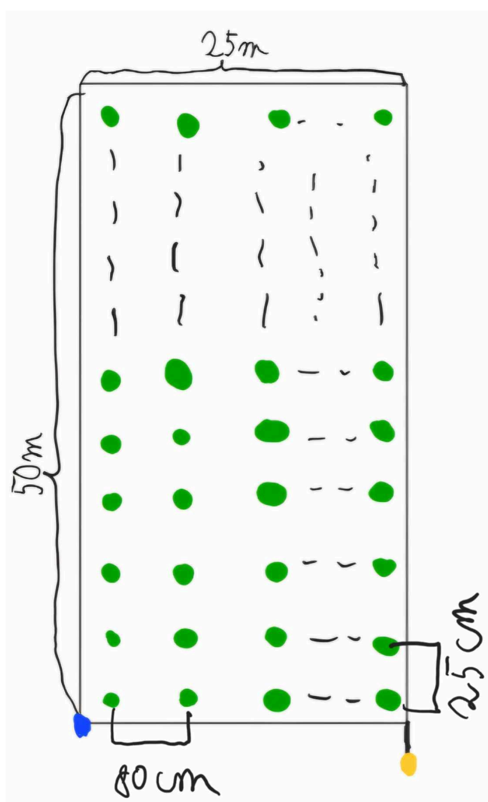


Fig. 15 The position of our corns

Conclusions

In conclusion, a potential answer to the urgent problems facing the agriculture sector is the incorporation of distributed architectures in the field of smart agriculture, especially in weed control. With its superior robotics, computer vision, and artificial intelligence capabilities, the intelligent robotic system presented in this study provides an effective and sustainable solution for managing weeds. Through accurate weed identification and removal, fewer toxic chemicals and labor-intensive methods are required, supporting resource efficiency and environmental sustainability.

Furthermore, the system's design guarantees continuous and independent operation and has a docking station for pesticide and energy replenishment. The robot's operational efficiency is further enhanced by the usage of a prepared map for systematic crop inspection and its capacity to memorize investigated locations.

This study emphasizes the potential of such creative solutions to address the dearth of trained workers in the agriculture sector, fight climate change, and fulfill the world's growing need for food. The demand for such clever farming solutions will only increase in tandem with the global population growth. To fully utilize these technologies and make them available to farmers everywhere, further research and development in this field are therefore vital. This will transform farming and open the door to a more sustainable and secure food supply in the future.

References

- [1] Du, Yayun, et al. "A low-cost robot with autonomous recharge and navigation for weed control in fields with narrow row spacing." 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021.
- [2] Kansal, Isha, Khullar, Vikas, Verma, Jyoti, Popli, Renu and Kumar, Rajeev. "IoT-Fog-enabled robotics-based robust classification of hazy and normal season agricultural images for weed detection" *Paladyn, Journal of Behavioral Robotics*, vol. 14, no. 1, 2023, pp. 20220105. https://doi.org/10.1515/pjbr-2022-0105
- [3] Li, Xiaodong, et al. "Robotic crop row tracking around weeds using cereal-specific features." *Computers and Electronics in Agriculture* 197 (2022): 106941.
- [4] How to Grow Corn Seeds - https://www.theseedcollection.com.au/growguide/corn-seeds
- Filbert H. Juwono and W.K. Wong and Seema Verma and Neha Shekhawat and Basil Andy Lease and Catur Apriono (2023). Machine learning for weed–plant discrimination in agriculture 5.0: An in-depth review, Artificial Intelligence in Agriculture, 10, 13-25. https://doi.org/10.1016/j.aiia.2023.09.002
- [5] Honghua Jiang and Chuanyin Zhang and Yongliang Qiao and Zhao Zhang and Wenjing Zhang and Changqing Song (2020). CNN feature-based graph convolutional network for weed and crop recognition in smart farming, Computers and Electronics in Agriculture,174, 105450. https://doi.org/10.1016/j.compag.2020.105450
- [6] Muhammad Hamza Asad and Abdul Bais (2020). Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network, Information Processing in Agriculture, 7, 535-545, https://doi.org/10.1016/j.inpa.2019.12.002
- [7] Punithavathi, R., Rani, A.D.C., Sughashini, K.R., Kurangi, C., Nirmala, M. et al. (2023). Computer vision and deep learning-enabled weed detection model for precision agriculture. Computer Systems Science and Engineering, 44(3), 2759-2774. https://doi.org/10.32604/csse.2023.027647
- [8] Kun Hu and Guy Coleman and Shan Zeng and Zhiyong Wang and Michael Walsh (2020). Graph weeds net: A graph-based deep learning method for weed recognition, 174, 105520. https://doi.org/10.1016/j.compag.2020.105520
- [9] Yangkai Zhang and Mengke Wang and Danlei Zhao and Chunye Liu and Zhengguang Liu (2023). Early weed identification based on deep learning: A review. Smart Agricultural Technology, 3, 100123. https://doi.org/10.1016/j.atech.2022.100123
- [10] Amin Nasiri and Mahmoud Omid and Amin Taheri-Garavand and Abdolabbas Jafari (2022). Deep learning-based precision agriculture through weed recognition in sugar beet fields. Sustainable Computing: Informatics and Systems, 35, 100759. https://doi.org/10.1016/j.suscom.2022.100759

- [11] Farms Are Becoming Smart: with Bits and Bytes as Today's Seeds Continental https://www.continental.com/en/press/studies-publications/continental-stories/smart-farming/
- [12] IoT-enabled smart agriculture: architecture, applications, and challenges, Quy, Vu Khanh and Hau, Nguyen Van and Anh, Dang Van and Quy, Nguyen Minh and Ban, Nguyen Tien and Lanza, Stefania and Randazzo, Giovanni and Muzirafuti, Anselme, 2022, MDPI
- [13] Sinha, B. B., & Dhanalakshmi, R. (2022). Recent advancements and challenges of Internet of Things in smart agriculture: A survey. *Future Generation Computer Systems*, *126*, 169-184.
- [14] Moraitis M, Vaiopoulos K, Balafoutis AT. Design and Implementation of an Urban Farming Robot. *Micromachines*. 2022; 13(2):250. https://doi.org/10.3390/mi13020250
- [15] ArduPilot Documentation: https://ardupilot.org/dev/docs/learning-ardupilot-introduction.html
- [16] ROS documentation: https://docs.ros.org/