Swarm Robotics for Sustainable Weed Management: Integrating Algorithm and Design

Zixin Yu

Shanghai World Foreign Language Academy

China

Section 1 Abstract

This study focuses on the development of a weeding robot with algorithmic and physical design, using swarm robotics to offer a solution for sustainable weed management. Traditional weed management methods, such as herbicides and mechanical removal, often harm soil fertility, water sources, and ecosystems. To address these challenges, this paper integrates a strategic weed removal algorithm with the design of a weeding robot equipped with vision and communication functions. The methodology consists of two components: firstly, the development of a strategic collective weeding algorithm aimed at minimizing total runtime in field operations, verified through computer simulations; secondly, designing a weeding robot equipped with vision and the swarm algorithm, using 3D modeling software and Arduino ESP32 board for information synchronization. Results show that strategic weeding method consistently outperforms random walks, especially when the field scale increases, evidenced by both qualitative and quantitative analyses. Another notable finding was the exponential decay in runtime with the addition of more agents, indicating efficiency gains will eventually hit a point of diminishing returns. Sensitivity analyses revealed that the distance threshold for repulsion calculation affect full coverage time to some extent, suggesting the importance of parameter selection. The integration of algorithmic and physical design demonstrates the effectiveness of swarm robotics as a robust solution for weed removal. Future work will explore parametric optimization and expanded testing to refine the approach. The solution is expected to be used in agricultural weed control, especially in organic farms where ecological protection is highly valued.

Section 2 Introduction and Goal

Weeds pose a significant challenge to global agricultural productivity, not only by competing or resources but also through non-competitive interactions that can lead to substantial losses in crop yields [1]. Traditional weed management approaches, including the application of herbicides and large-scale mechanical control, have been associated with loss in soil fertility, contamination of natural resources, and the disruption of ecosystems [2]. In light of these challenges, the adoption of sustainable weed management presents a viable pathway to ensuring food security and restoring ecosystem balance [3]. The shift towards sustainable practices underscores the need for integrated approach that considers both agricultural and environmental goals.

In sustainable farming, current weed management strategies can be broadly categorized into physical and biological approaches [4]. The physical method involves removing weeds using tools or machinery, primarily carried out by workers, which can lead to high labor costs and low efficiency over large areas. On the other hand, the biological approach uses natural enemies and biocontrol agents, which, while environmentally friendly, often require specific conditions to be effective. In this context, this paper introduces a swarm robotics strategy designed to achieve efficient and effective weed removal and environmental protection simultaneously.

'Swarm' refer to a large group of locally communicating individuals with common goals [5]. In the context of weed removal, the swarm robotics approach is exemplified by strategic collective weeding behavior. Verlekar, H. et al. [6] proposed a method based on the chain formation behavior of ants and division of labor, implemented on two robots with the aid of computer vision. The approach is open to practical uses such as in warehouses and space exploration, but a specific application scenario has yet to be determined. Albani, D. et al. [7] [8] presented a decentralized multi-agent system for a weed mapping problem, which enables efficient coverage through biasing random walks, with a roadmap towards bringing swarm robotics to farmland. However, the algorithm focuses only on weed labeling in a field and does not address long-term weed removal; the specifics regarding swarm robots equipped with the algorithm remains undefined. Identifying these gaps, this paper aims to develop a comprehensive product integrating strategic weeding algorithm, vision training and hardware design to bring forward the efficient and effective application of swarm robotics in fields.

The work of this paper includes two components: (1) Developing a strategic weed removal algorithm that facilitates collective weeding behavior with reduced field coverage time, with results demonstrated through computer simulations, and (2) designing a compact weeding robot capable of communication and data synchronization, adaptable to various field conditions. The integration of these two components forms a comprehensive solution that addresses the current weaknesses in practicality and completeness of weed management methods, advancing sustainable agricultural weed control.

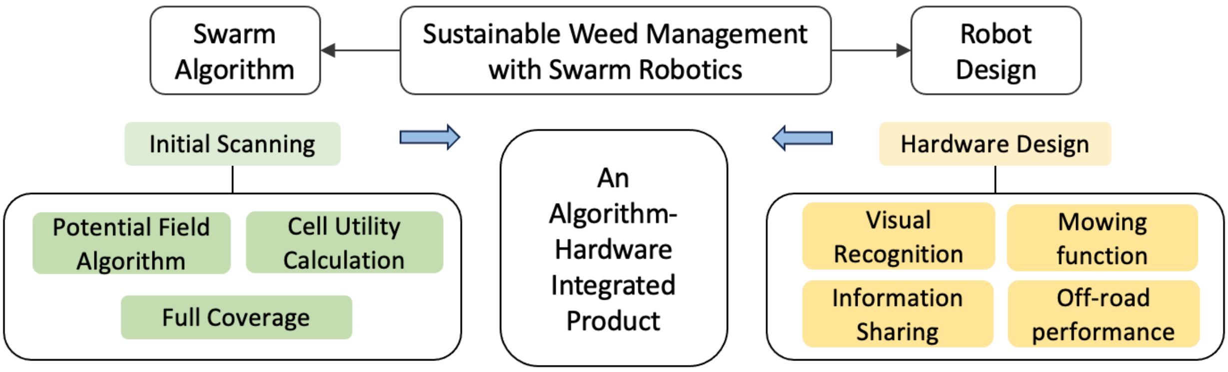


Figure 1: A schematic of the work done in this paper. The final product consists of robot design and swarm algorithms which work together to achieve sustainable weed management with high appicability.

Section 3 Method

§ 3.1 Algorithmic Design: Strategic Collective Weeding

The strategic collective weeding algorithm aims to minimize energy consumption of the swarm robots, thereby maximizing the goal of achieving sustainable weed clearance. Lower energy consumption indicates extended battery life and increased efficiency, which, in the context of weed removal, suggests a shorter total operational distance in one or multiple runs given a uniform robot speed. The algorithm described in this work considers weighting farmland units (henceforth referred to as cells), drawing robots to areas with a higher likelihood of weed emergence, while also constantly monitoring the cells in the vicinity as they move towards their destination and uploading the latest data. Meanwhile, since swarm robotics is organized as a system, we aim to prevent a cell from being revisited and to avoid collisions between two robots. This is done by creating a repulsion between individual agents within an effective range. Based on these insights, we construct an attraction-repulsion system that works together to determine the directional bias of each agent, which can be explained by a potential field [9]. Before clarifying the detailed working mechanism, we simulate a farmland and establish a format for representing weed detection.

Consider a field $\mathcal{F}$ of dimensions $D \times D$, divided into an $M \times M$ grid for simulation purposes. The cells in this grid are denoted as c_{ij}, where i and j index the cell's position within the matrix, and each cell has a side length l with $Ml = D$. For each cell c_{ij}, we assume a weed density ρ_{ij}. In the simulation, weeds are hypothesized to be distributed unevenly across the field, forming clusters. These clusters are modeled using a Gaussian distribution across N patches. The vision algorithm returns $\hat{\rho_{ij}}$, where $\hat{\rho_{ij}} \leq \rho_{ij}$ due to undetected weed units at the blurry edges within the photographic range. The confidence value for each cell is represented as v_{ij}. The estimations for detected weed density and confidence values for agent m in this experiment are derived as follows:

$$\hat{\rho_{ij}} = \rho_{ij}(1-\epsilon), \quad \epsilon \sim \mathcal{U}(0, \epsilon_{max}) \tag{1}$$

$$\hat{v_{ij}} = 1 - \epsilon_v, \qquad \epsilon_v \sim \mathcal{U}(0, \epsilon) \tag{2}$$

where $\mathcal{U}(a, b)$ denotes a uniform distribution in the interval $[a, b]$.

Our proposed algorithm includes an initial scanning phase in which robots traverse the field to map out weed densities and perform weed removal. In subsequent run $k+1, k = 2, 3, 4 \ldots$, it is hypothesized that cells identified with higher weed densities in the kth run are biased, as the robots did not fully uproot the weeds but instead engaged in a more gentle form of removal. Since the roots of the weeds were not eradicated, areas where weeds were previously detected are more likely to experience regrowth, even after surface-level clearance. During the field coverage process, it is possible for some cells to be visited multiple times. Consequently, for each cell's adjusted weed density $\hat{\rho_{ij}}$, the highest value of $\hat{v_{ij}}$ recorded across all visits should be selected:

$$< \hat{\rho}, v >_{ij} = \max_{v}\{< \hat{\rho_m}, v_m >_{ij}, k = 1, 2, \ldots, t\} \tag{3}$$

where t denotes the total number of visits. Given that our robots are equipped with wide-angle cameras, in our simulated field, we simplify their visual range to include the left, front, and right cells based on the direction of their frontage. This means that these three cells are documented and their data uploaded during a single visit. During initial scanning, the robots directly proceed to the cells where weeds have been detected and record the weed density and confidence data of all cells monitored along the way. This process continues until all cells have been effectively covered. In each subsequent run, the swarm robotics use this established weed density map to conduct collective weeding, but also continuously update the new weed density in the current operation.

The goal of our strategic weeding algorithm is to minimize the energy consumption while ensuring effective coverage of the field. Therefore, we aim to minimize the total running time required to achieve full coverage in each operation. To achieve this, we have designed two types of forces: attraction and repulsion. Attraction guides the robots towards cells with a higher probability of weed emergence, while repulsion prevents too many robots from congregating in a single patch and discourages multiple visits to the same cell. It is expected that areas with a lower likelihood of weed growth will be covered by the peripheral vision of the robot. The idea of potential field is hence introduced to construct the attraction-repulsion system.

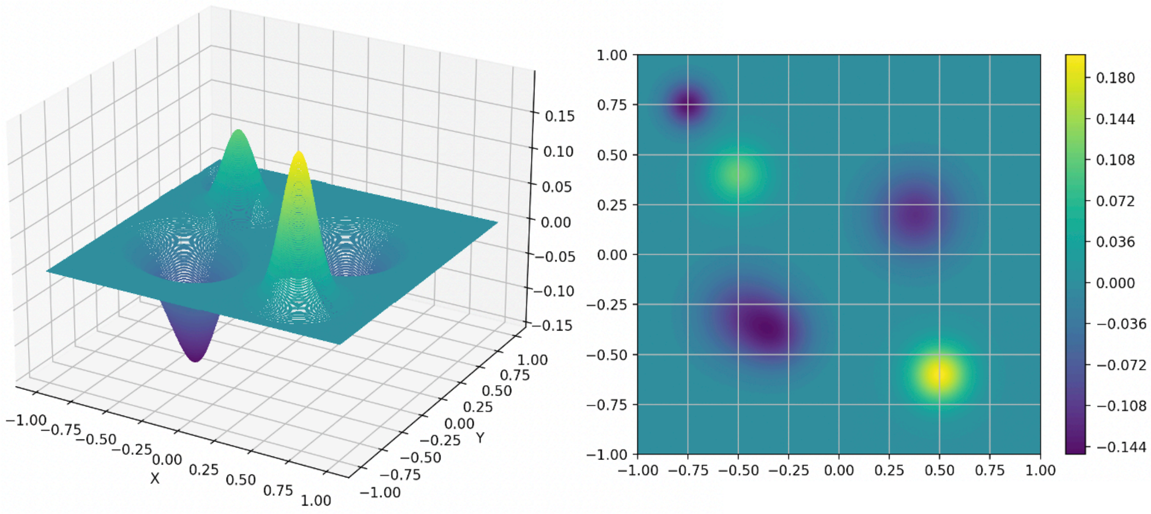


Figure 2: Left: A sample potential field in a 8×8 field. Right: 2D representation of potential mapping.

Figure 2 illustrates a potential field within an 8×8 area. The concave areas represent cells that had higher weed density during the most recent collective operation, whereas the convex areas denote zones with high agent density. As an object always moves from a region of high potential to a region of low potential due to the natural tendency to minimize energy, our weeding robot is predisposed to travel toward cells with assumed higher weed density and avoid cell revisiting. The summation of attractive and repulsive field results in a comprehensive field where directional bias is finally determined. Following this, we methodically define the attractive and repulsive field and compute the resultant forces respectively.

The attractive field for agent m defined as:

$$U_{att,ij} = -k_{att} \cdot \rho_{ij} \cdot f(d_{ij}) \tag{4}$$

$$f(d_{ij}) = \frac{1}{1 + d_{ij}^2},\ d_{ij} = \|\vec{x_{ij}} - \vec{x_m}\| \tag{5}$$

where k_{att} is a scaling factor, and ρ_{ij} stands for c_{ij}'s weed density measured in the last run. $\vec{x_{ij}}$, $\vec{x_m}$ denote position vectors of cell c_{ij} and agent m respectively. d_{ij} hence represents the distance between c_{ij} and agent m. The function $f(d)$ is included in the expression to give preference to cells that are closer to the current position of agent m. This approach ensures that movements do not overlook cells in close proximity, avoiding the skipping of nearby cells in a single move. To define the repulsive field for agent m, our goal is not only to prevent revisiting and high agent density in one area but also to avoid the influence from cells that are significantly distant from agent m. Therefore, we introduce a distance restriction, d_0, and formulate the repulsive field, U_{rep}, through equation (6):

$$U_{rep,ij} = \begin{cases} \frac{1}{2} k_{rep} (\frac{1}{d_{ij}} - \frac{1}{d_0})^2, & d_{ij} < d_0,\ c_{ij} = c_k \\ 0, & d_{ij} \geq d_0,\ c_{ij} \neq c_k \end{cases} \tag{6}$$

where k_{rep} is a scaling factor, and c_k represents the cells all agent $k \neq m$ are located. $U_{rep,ij}$ returns a positive potential for other agents within a specified range, thereby guiding agent m towards areas with a lower density of agents. The resultant field constructed by attractive and repulsive potential is hence:

$$U_{total,ij} = U_{att,ij} + \sum_{c}^{\mathcal{F}} U_{rep,ij} \tag{7}$$

where $\mathcal{F}$ denotes the field we're examining and c represents the cells in this field. To model the resultant force generated by the total potential field, the relationship between force and potential is derived. Under a physical context, force is the negative gradient of potential energy with respect to position. Mathematically, $\vec{F} = -\nabla\vec{U}$, where ∇ denotes the gradient operator, which measures the rate and the direction of change in a field. The force of c_{ij} with respect to m is hence:

$$F_{total,ij} = -\nabla U_{att,ij} - \nabla \sum_{c}^{\mathcal{F}} U_{rep,ij} \tag{8}$$

Based on the calculations of force, we ultimately assign a value to each cell, representing their corresponding utility. However, prior to this, we introduce a hierarchy-based cell selection. In the algorithm, the shorter the distance moved in a single step, the smaller the likelihood of backtracking and energy consumption. Therefore, to achieve this goal, the Manhattan distance between agent m and the cells is considered, as shown in figure 3. Within this hierarchy constructed by the Manhattan distance, agent m will first consider cells with a Manhattan distance of 1 and select the cell with the highest utility value. If all cells at this level have already been covered, it then considers distances of 2, 3, and so forth, prioritizing cells based on their proximity and utility in a hierarchical manner.

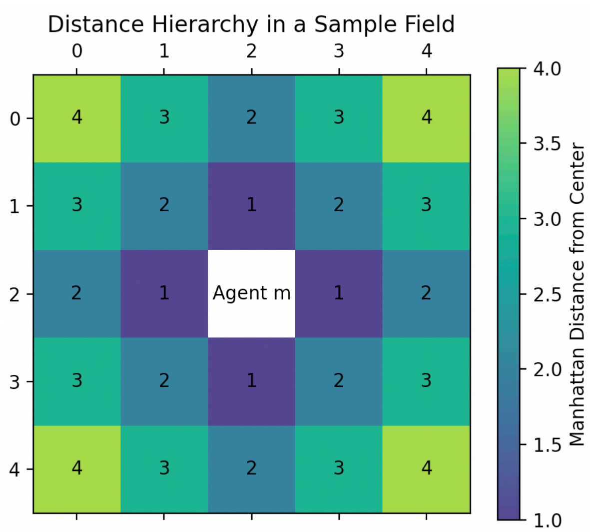


Figure 3: Distance hierarchy for cells around agent m.

The utility value is computed according to equation (9) (10):

$$\theta_{ij} =< F_{total,ij}, \vec{d_{ij}} >, \ \vec{d_{ij}} = \vec{x_{ij}} - \vec{x_m} \tag{9}$$

$$u_{ij} = C(\theta_{ij}, \gamma), \quad C(\theta, \gamma) = \frac{1}{\pi} \cdot \frac{1}{\gamma + \theta^2} \tag{10}$$

where $C(\cdot)$ is a Cauchy probability density function, and γ refers to the Median Absolute Deviation

(MAD) from the mean in this distribution.The reasoning behind this approach is that we want the agent to move towards the cell that has the smallest angle with the directional bias, essentially the cell closest to the directional bias within the hierarchy. After reaching this cell, agents will repeat the process of scanning, weeding, and selecting, until all cells have been effectively covered. The cell selection and coverage process is encapsulated by **Algorithm 1**.

Algorithm 1 Weeding Simulation with Hierarchical Utility Calculation

Initialize: Field $\mathcal{F}$ with size $M \times M$, N patches with uniform distribution, n agents with random positions, parameters k_{att}, k_{rep}, d_0, and γ.
function MoveAgent($\mathbf{a}, \mathcal{F}$):
    Current cell and check front, left, and right cells for weeds
    Best move **m** $\leftarrow$ None
    if exactly one of front, left, or right cells has weeds **then**:
        m $\leftarrow$ cell with weeds
        Mark front, left, and right cells as covered
    else
        for $d = 1$ to M **do** ▷ Iterate over increasing Manhattan distances
            Initialize list of candidate cells C at distance d
            Compute utility u for each cell in C
            Filter C to include only uncovered cells
            if C is not empty **then**:
                m $\leftarrow$ cell in C with maximum utility
                Break the loop
            end if
        end for
        Mark **m** as covered
    end if
    return m
end function
while not all cells in $\mathcal{F}$ are covered **do**:
    for each agent **a do**:
        Mark the current cell **a** is in as covered
        m $\leftarrow$ MoveAgent($\mathbf{a}, \mathcal{F}$)
        if m $\neq$ None **then**:
            Update **a** position to **m**, mark **m** as covered
        end if
    end for
end while
Output: Maximum runtime among all agents

In our simulation, the runtime is calculated based on a fixed computational delay of 2 seconds, and a travel delay of 3 seconds for each unit of Manhattan distance traveled. To validate our algorithm, we conducted four sets of experiments as outlined in Table 1.

Standard Condition	**Method: Strategic Collective Weeding Algorithm,** $M = 32, N = 6, n = 4, k_{rep} = 0.02$**,** $k_{att} = 0.1, d_0 = 8, \gamma = 0.6$		
Group Number	Dependent Variable	Independent Variable	Value Specification
1	Total Runtime (In a single operation)	Method, M	Strategic, $n = 4$; Random, $n = 4$; 'Segmentation', $n = 1$ $M = 8, 16, 24, 32, 40, 48$ $d_0 = M/4$
2		n	$n = 2, 3, 4, 5, 6, 7, 8, 9, 10$
3		k_{rep}	$k_{rep} = 0.02, 0.05, 0.08, 0.11, 0.14, 0.17, 0.20$
4		d_0	$d_0 = 5, 6, 7, 8, 9, 10, 11$

Table 1: Simulation Parameter Specification

The experiments consist of four control groups and examine the effect of different variations on total runtime in a single operation, i.e., full coverage and weed removal, in our simulated field $\mathcal{F}$. The first control group aims to demonstrate the effect of our strategic collective weeding algorithm in comparison to the conventional random walk strategy. In this scenario, robots are initialized at random starting points and proceed to survey the cells within their vision range. Upon identification of weeds, a robot moves to the corresponding location; otherwise, it selects a neighboring cell at a Manhattan distance of one for its next movement. This process repeats until all cells are effectively covered. We hypothesize that the 'Strategic' method will demonstrably decrease runtime, particularly in large-scale fields. The subsequent three control groups are dedicated to conducting a sensitivity analysis on the proposed algorithm, focusing on the variables n, k_{rep}, and d_0. The examination of these variables aims to provide insights into improving the 'Strategic' algorithm and increasing our understanding of how it works.

The simulations are carried out using Python 3.9.13 on a MacBook Pro 2019 with an Intel Core i9 processor.

§ 3.2 Physical Design and Vision Training

In the robot design section, we aim to achieve several objectives: (1) Robots should possess sufficient off-road capabilities to handle various terrain conditions and obstacles, such as rocks blocking the path. We aim for the vehicle to easily overcome obstacles, avoiding rollovers and other mishaps. (2) In addition to basic controls such as forward movement and turning, robots should possess WiFi and communication functionalities to enable algorithm deployment and information sharing with other robots, forming a swarm robotics system. (3) Robots must be equipped with cameras to capture images of weeds and possess the ability to accurately identify them using well-trained visual models. Based on these objectives, we focus on constructing and presenting the three aspects of off-road performance, algorithm deployment, and vision training:

- **Off-road performance**: To achieve good off-road capabilities, we designed wheels that are compatible with the vehicle's size. These wheels use flexible materials, and feature a pattern of bilaterally symmetrical and raised rectangular treads, forming regular grooves. Such a pattern increases the friction between the tire and the ground during operation, ensuring better traction; additionally, the grooves facilitate water and mud evacuation, preventing accumulation on wet and muddy surfaces and improving the tire's grip in these conditions. Beyond the primary wheels, the chassis has been equipped with four omnidirectional wheels to augment the robot's multi-directional movement and stability, enabling it to navigate environments with numerous obstacles without the risk of overturning or slipping.

- **Algorithm Deployment**: The robot must possess communication capabilities to upload location and weed density information among all agents, allowing the designed algorithm to function effectively. Therefore, in addition to the main control board, we have installed Arduino ESP32 boards. These microcontroller development boards, based on the ESP32 chip—a powerful, Wi-Fi and Bluetooth-enabled system-on-chip—enable the ESP32 boards to connect to wireless networks and communicate with other agents. This feature harmonizes with our algorithm's requirements.

- **Vision Training**: For the visual training component, we utilized the YOLOv2 network [10] trained on the plant seedling dataset [11], focusing on recognizing maize and shepherd's purse. The choice of this dataset, featuring weeds and crop seedlings, is based on the premise that once the crops reach a certain growth stage, weeds can be accurately identified by height alone, eliminating the need for visual recognition based on shape or color. In terms of data augmentation, we randomly selected seedling images to compile into composite photos containing multiple plant groups; we also used image flipping and rotation to further expand the dataset. All images were annotated and resized to 224×224 pixels. The test set achieved an accuracy rate of approximately 80%, which will be shown in the figures below.

We have equipped the base plate with motors and installed cutting lines beneath them to remove weeds, using flexible materials to prevent breakage. In order to protect the components from damage due to collisions, we have also positioned a black casing around the main body. The designs are sketched in Fusion 360 and produced via 3D printing. Figures 4 and 5 showcase the design outcome and robot object.

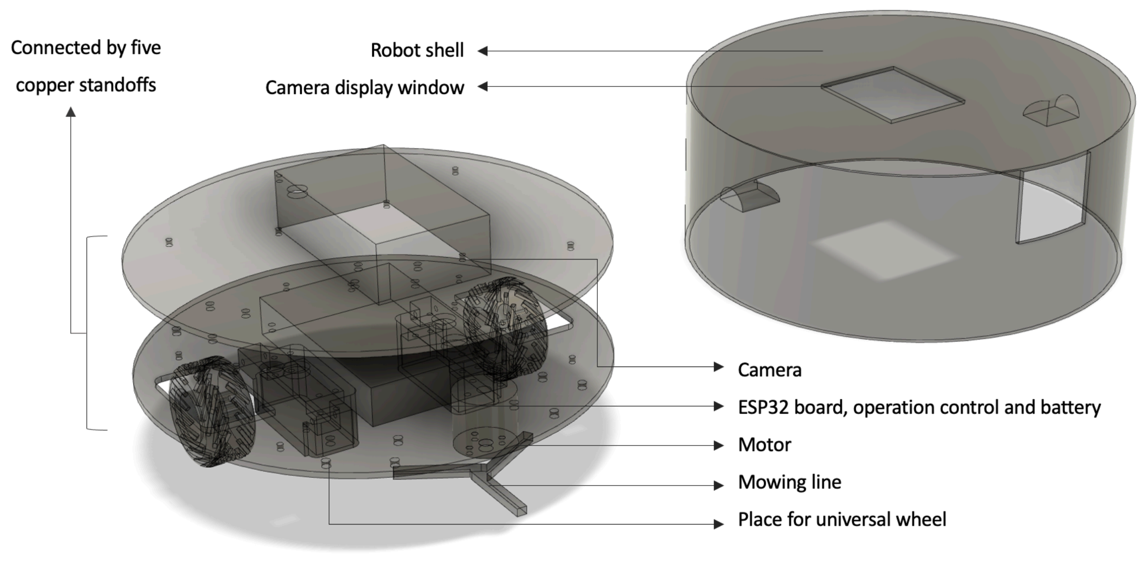


Figure 4: Robot design sketches and component labeling.

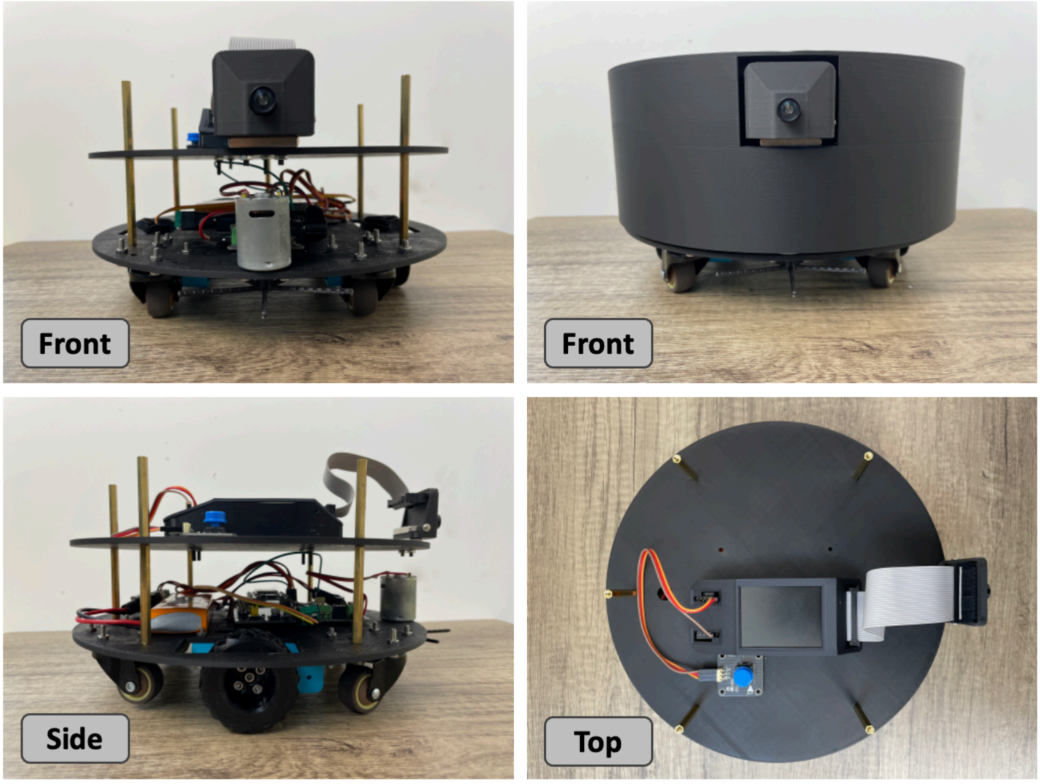


Figure 5: Front, side, and top views of the robot entity.

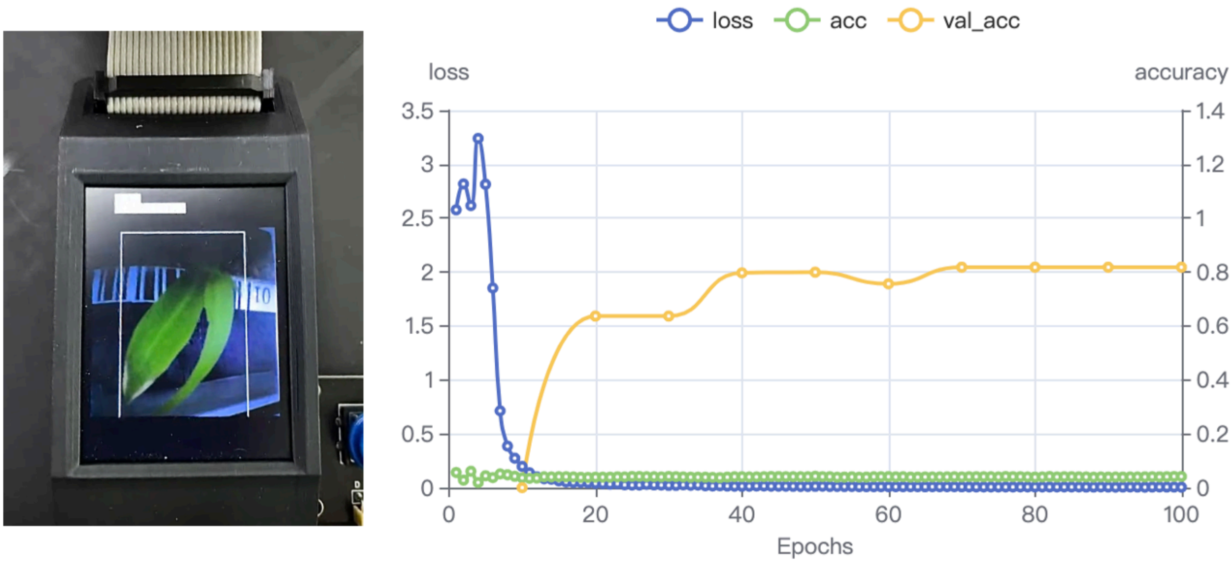


Figure 6: YOLOv2 vision training results. Left: Recognizing maize in a real scenario. Right: Loss, accuracy, and validation accuracy in 100 epochs.

Section 4 Results

Based on the parameter specifications in Table 1, we conducted four sets of experiments. The runtime for each data group is the average of 100 simulations. For data points and curves with clear functional relationships, curve fitting was conducted. The results are shown in figure 7, 8 and 9.

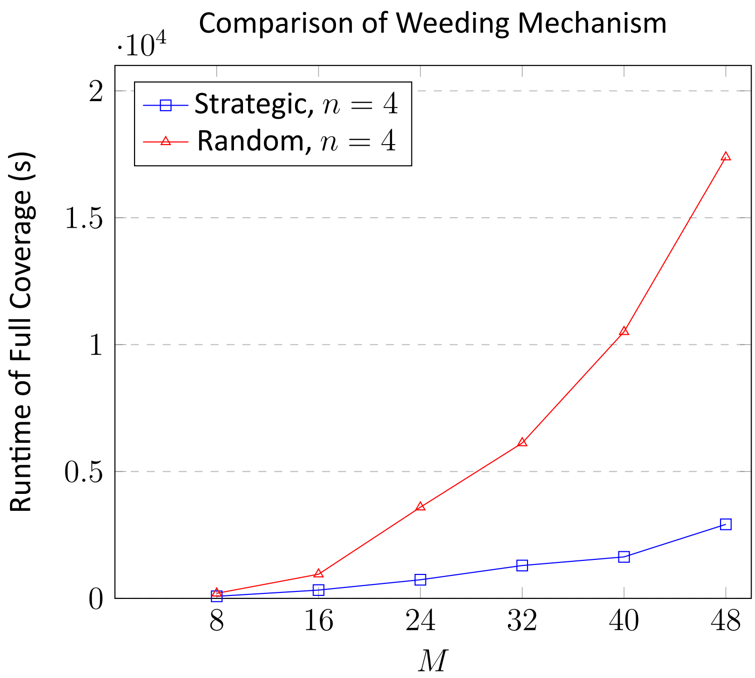
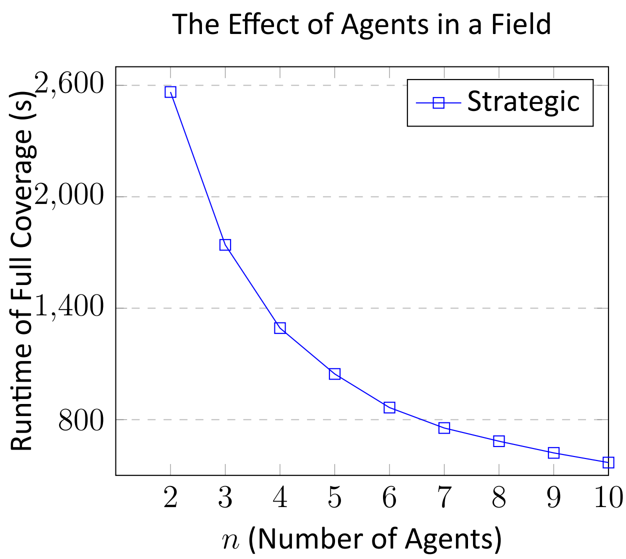


Figure 7: Runtime comparison of 'Strategic' and 'Random' when $M = 8, 16, 24, 32, 40, 48$. 'Runtime' represents the total time taken for the swarm robots to remove weeds and cover the entire simulated field.



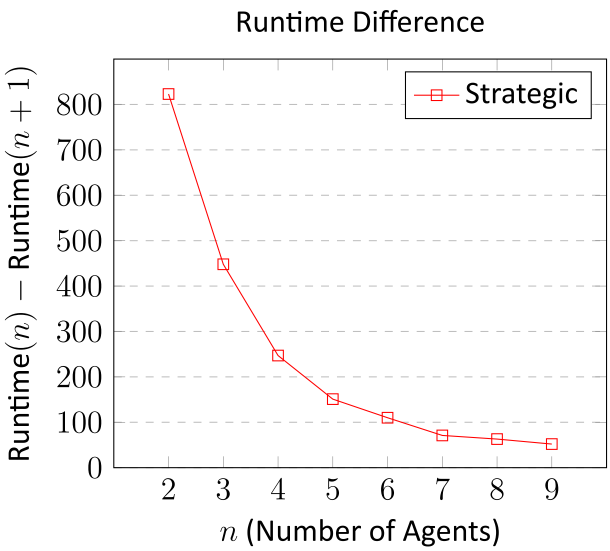


Figure 8: Left: The impact of agent quantity alongside other parameters under standard conditions. Right: The runtime difference between agent number n and $n + 1$.

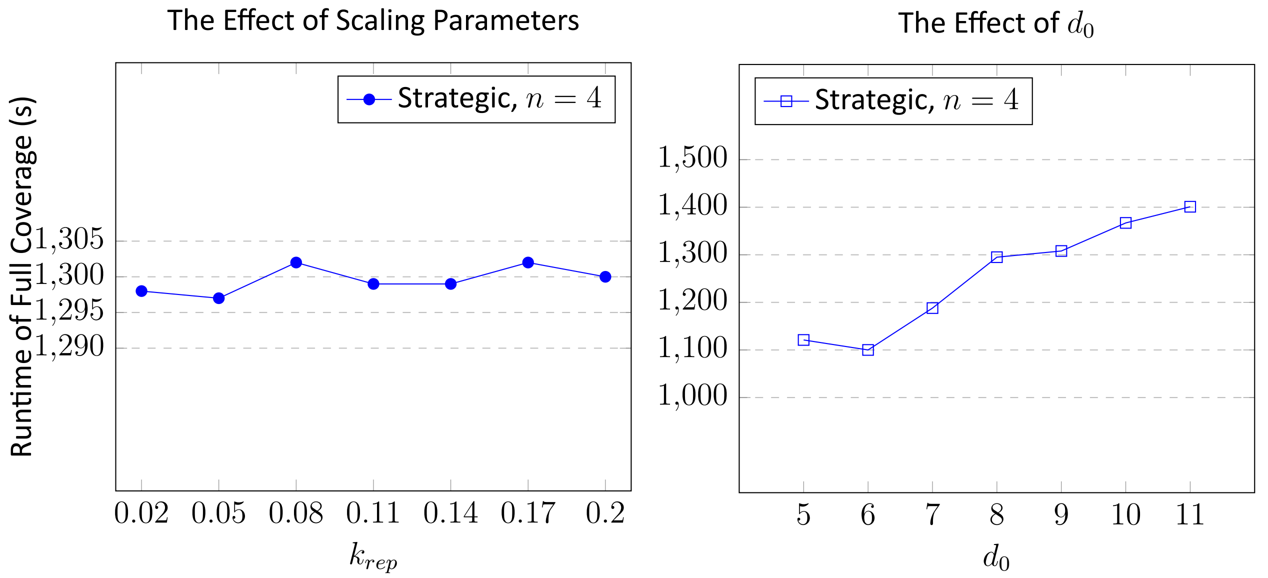


Figure 9: Left: The impact of k_{rep} other parameters under standard conditions. Right: The effect of d_0 alongside other parameters under standard conditions.

The data from Group 1 and Group 2 demonstrate strong functional relationships, as shown in Figure 7 and 8. Therefore, linear and exponential regressions were performed on the data points, producing the results depicted in Figure 10. In Group 1, the RMSE (Root Mean Square Error) for 'Strategic' is 256.52, and for 'Random' it is 3422.11, while for Group 2, it is 277.94. Due to limitations in data points and since the primary objective of the simulation is to observe quantitative relationships rather than qualitative ones, RMSE values below 500 are considered to be in a reasonable range. All the results will be discussed and interpreted in the next section.

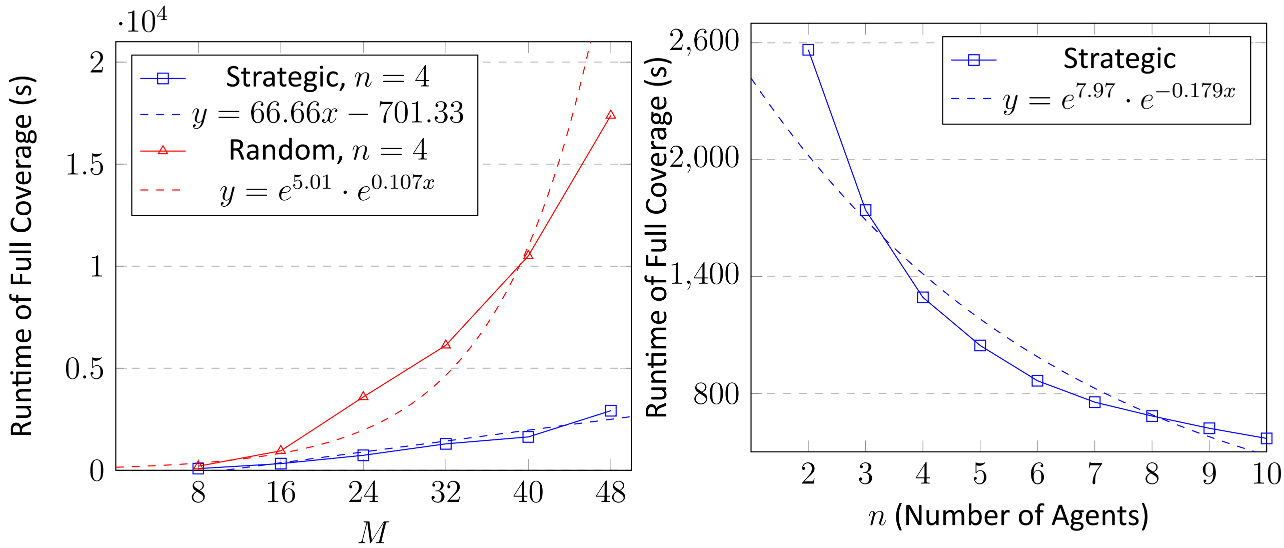


Figure 10: Left: Regression analysis for data points in group 1. Right: Regression analysis for data points in group 2.

Section 5 Discussion

Group 1: Runtime comparison of 'Strategic' and 'Random'

1. **Qualitative Analysis**: Figure 8 illustrates a clear trend: as the parameter M increases, both methods show longer runtimes. This is unsurprising, as larger fields or more complex tasks naturally require more time to complete. Initially, there is a slight difference in runtime between the two methods, likely due to the smaller field size reducing the complexity of full coverage and weed removal. The developed swarm algorithm, labeled 'Strategic', consistently outperforms the random walk method as M increases, which is in line with our expectations. The difference in performance between the two methods becomes more transparent as the task complexity M grows, suggesting that the benefits of the strategic method are amplified in more complex scenarios.

2. **Quantitative Analysis**: In the context of our established criterion that an RMSE below 500 is considered acceptable given the limitations in the number of data points, the RMSE value for 'Random' is notably high, recorded at 3422.11. Two explanations are postulated for this value: Firstly, the limited sample size may result in reduced reliability of the estimates. The inherent variability within small datasets can magnify the impact of outliers, thereby inflating RMSE. Secondly, there is a high degree of stochastic variability in 'Random', which implies that an average derived from 100 simulation runs may not sufficiently converge towards the true expected value. Meanwhile, it is noteworthy that curve fitting serves as a supplementary tool for interpretation as we highlight the linear nature of 'Strategic', demonstrating the effectiveness of our algorithm.

Group 2: The effect of agent quantity in same sizes of field

1. **Qualitative Analysis**: It can be observed from Figure 8 that, as the number of agents increases, the time taken for full coverage decreases. This trend suggests that the 'Strategic' algorithm's efficiency improves with more agents at work. Additionally, as the number of agents increments, the rate of runtime decrease slows down, indicating a diminishing return on the addition of more agents. The benefits gained by increasing the number of agents can be represented by the runtime difference between adjacent ns, as shown in Figure 8 (Right).

2. **Quantitative Analysis**: The trend is modeled by an exponential decay function with RMSE value of 277.94. Beyond the small volume of the dataset, employing discrete data points for exponential regression may introduce fitting issues and error. Employing techniques such as differencing or recoding variables may ameliorate these errors. Furthermore, the fit of the function provides insights into determining the optimal number of agents for each field size. This optimal n likely demands comprehensive consideration of factors such as energy consumption and actual runtime, giving the formulation of quantitative function high importance.

Group 3: Sensitivity analysis on the scaling parameters

The objective of Group 3 (Figure 9, left) is to investigate the impact of scaling parameters within the potential field algorithm. The decision to solely adjust k_{rep} arises from the importance of the relative relationship between k_{rep} and k_att, as the forces will ultimately undergo mapping via a

Cauchy distribution.T he runtime remains relatively constant across different values of k_{rep}, fluctuating around 1,300 seconds, suggesting that changes in the repulsion coefficient do not significantly impact runtime for full coverage in this scenario. The consistency of the runtime can be explained by the small number of agents, which likely results in a low level of repulsive interaction among them, regardless of the magnitude of $krep$. With restriction d_0, some repulsion from other agents may not be considered due to large separation and increasing k_{rep} values do not substantially alter the dynamics of their movement. The results suggests that for a small number of agents, the scaling factors are not critical in the strategic coverage time. Future work may seek to observe the effect of scaling parameters in fields with large agent quantity.

Group 4: Sensitivity analysis on d_0

As d_0 increases from 5 to 11, there is a general trend that the total runtime also increases, suggesting that a larger d_0 leads to a longer runtime. In this scenario, robot might be taking into account agents that are farther away, which are not as relevant to its immediate decision-making about avoiding high agent density areas. This result provides us with an insight that agents with large separation (greater distances apart) should not significantly affect the robot's behavior, and the value of d_0 should be carefully chosen.

Based on these findings, three future progress plans are outlined. Firstly, while the algorithm has been proven effective, future efforts can focus on proposing better methods for parameter selection based on the research results. When considering parameter optimization, further integration with data such as robot energy consumption, can be undertaken to find the optimal combination. Secondly, the number and scale of simulations can be further expanded, and running the robots in actual fields should also be attempted. Integrating actual performance with computer simulations not only reinforces the dataset for fitting quantitative relationships but also provides a better understanding of the model's performance. Lastly, the swarm system composed of algorithms and robot design can be further detailed, with promotion and trial runs at various farm units to gather insights for optimizing the algorithm and adjusting the model's stability, which contributes to the final application. These steps will serve as critical steps towards sustainable weed control with swarm robotics.

Section 6 Conclusion

Through qualitative and quantitative analysis, we demonstrated that the strategic weeding method outperforms the random approach, especially as field scale increases. At the same time, through robot design and visual training, we have developed a functioning entity, which not only provides a practical platform for the algorithm but also expands its application scope. The integration of these two components has proven the effectiveness of swarm robotics as a robust solution for sustainable weed removal. In the process of moving towards widespread application, future work will focus on parameter optimization and expanded testing. Ultimately, we expect this solution to be used in agricultural weed control, especially in organic farms where ecological protection and sustainable development are highly valued. Furthermore, this algorithm, optimized for reducing weed removal runtime, also has the potential to be applied in other areas, such as path planning and energy consumption minimization.

References

[1] Bagavathiannan, M., Beckie, H. J., Chantre, G. R., González-Andújar, J. L., León, R. G., Neve, P., Poggio, S. L., Schutte, B. J., Somerville, G. J., Werle, R., & Van Acker, R. C. (2020). Simulation models on the ecology and management of arable weeds: structure, quantitative insights, and applications. Agronomy, 10(10), 1611. https://doi.org/10.3390/agronomy10101611

[2] Monteiro, A., & Santos, S. R. D. (2022). Sustainable approach to weed management: The role of precision weed management. Agronomy, 12(1), 118. https://doi.org/10.3390/agronomy12010118

[3] Reddy, B. S. (2010). Organic Farming: Status, Issues and Prospects - A review §. Agricultural Economics Research Review, 23(2), 343–358. https://doi.org/10.22004/ag.econ.97015

[4] Schwarzländer, M., Hinz, H.L., Winston, R.L. et al. Biological control of weeds: an analysis of introductions, rates of establishment and estimates of success, worldwide. BioControl 63, 319–331 (2018). https://doi.org/10.1007/s10526-018-9890-8

[5] Nandan, R., S, L., Narasannavar, N., S A, P., & S, S. (2020). Modern Agriculture Using Swarm Robots. International Journal of Advances in Engineering and Management (IJAEM), 2(4). https://doi.org/10.35629/5252-02046770

[6] Verlekar, H., & Joshi, K. (2017). Ant & bee inspired foraging swarm robots using computer vision. 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT). https://doi.org/10.1109/iceeccot.2017.8284663

[7] Albani, D., IJsselmuiden, J., Haken, R., & Trianni, V. (2017). Monitoring and mapping with robot swarms for agricultural applications. IEEE AVSS 2017. https://doi.org/10.1109/avss.2017.8078478

[8] Albani, D., Manoni, T., Arik, A., Nardi, D., & Trianni, V. (2019). Field Coverage for Weed Mapping: Toward Experiments with a UAV Swarm. Springer eBooks, 132–146. https://doi.org/10.1007/978-3-030-24202-2_10

[9] Li, G., Yamashita, A., Asama, H., & Tamura, Y. (2012). An efficient improved artificial potential field based regression search method for robot path planning. In 2012 IEEE International Conference on Mechatronics and Automation (pp. 1227-1232). Chengdu, China: IEEE. https://doi.org/10.1109/ICMA.2012.6283526

[10] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 6517-6525). Honolulu, HI, USA: IEEE. https://doi.org/10.1109/CVPR.2017.690

[11] V2 Plant Seedlings Dataset. (2018). Kaggle. https://www.kaggle.com/datasets/vbookshelf/v2-plant-seedlings-dataset/data