Agricultura Scientia 22: No 1-2 11-20(2025)
https://doi.org/10.18690/agricsci.22.1-2.2

University of Maribor Press

Advanced Navigation and Artificial Intelligence Techniques: Team Carbonite's Winning Strategies at the Field Robot Event 2023

Samuel MANNCHEN*, Jonas MAYER, Janis Lion SCHÖNEGG, Klara FAUSER

Schülerforschungszentrum Südwürttemberg e.V. Standort Überlingen, Obertorstraße 16, 88662 Überlingen, Germany

ABSTRACT

An approach to address current challenges in agriculture caused by climate change, the increasing global population and the loss of biodiversity is precision farming, for which agricultural robotics is a key enabler. The Field Robot Event (FRE) 2023 has challenged student teams to develop and improve autonomous agricultural robots. This paper presents the improvements to our field robot "Carbonite," which is developed at the Schülerforschungszentrum (SFZ) Südwürttemberg. Our lightweight and compact robot design, supported by our advanced and efficient navigation algorithm, enabled our robot to quickly move through fields. Additionally, we introduced our newly developed system for targeted and precise application of water, fertilizer and herbicides, based on an intelligent gap detection algorithm to avoid wasting resources. Also, we trained an object recognition AI model based on the You Only Look Once (YOLO) models, allowing the robot to appropriately respond based on the type of obstacle. Carbonite managed to secure the first place in both the navigation task and the plant treatment task, benefiting from the lightweight design and the resulting high robot driving speed, enabling us to win the overall FRE 2023 contest.

Keywords: agricultural robotics, precision agriculture, sustainability, artificial intelligence

INTRODUCTION

Agriculture faces major challenges, such as climate change continuing to change farming conditions globally (Nelson et al., 2009), the increasing global population creating higher food demand (Foley et al., 2011) and the loss of biodiversity undermining the resilience of agricultural ecosystems (Cardinale et al., 2012). These issues underscore the need for advances in farming to ensure sustainable food production and environmental stability.

Robotics and artificial intelligence (AI) offer promising avenues to improve sustainability and efficiency in agriculture. As shown in recent studies, precision farming technologies, such as targeted herbicide application in maize fields, can significantly reduce chemical usage and environmental impact (Malhi et al., 2021). However, many of these systems are hampered by high costs, complex maintenance requirements, and limited adaptability to the diverse conditions encountered in real-world farming (Gil et al., 2023).

In this paper, we present the advances of Carbonite, building upon our prior work (Mayer et al., 2022) to enhance its efficiency and practical utility in field applications. Carbonite integrates multiple sensors and AI-based object recognition capabilities designed to overcome the shortcomings of current systems. Our central hypothesis is that by optimizing our robot, Carbonite can achieve superior performance in critical tasks–such as object detection and gap recognition–thereby increasing its adaptability and precision.

To rigorously test this hypothesis, we are evaluating Carbonite's performance at the Field Robot Event (FRE) (Field Robot Event, 2022), a competition that challenges student teams to compete in tasks inspired by real-world farming. The dynamic conditions of the FRE provide valuable insights into the operational strengths and limitations of our system.

**Correspondence to:*
E-mail: samuel.mannchen@sfz-bw.de

MATERIALS AND METHODS

Robotic Base

The Carbonite robot is an autonomous agricultural platform designed for maize field operations, integrating precision farming techniques for irrigation, fertilization, and weed control. Utilizing sensor-based navigation and targeted liquid application, Carbonite maximizes resource efficiency while reducing waste. The robot's chassis features a three-layered design (see Fig. 1) that ensures structural stability, accessibility, and modularity. Constructed from carbon fiber-reinforced plywood, the chassis balances durability with a lightweight frame, crucial for minimizing soil compaction in agricultural environments. With a total weight of just 16 kg and chassis walls only 2 mm thick, the structure remains robust while maintaining mobility and efficiency in the field.

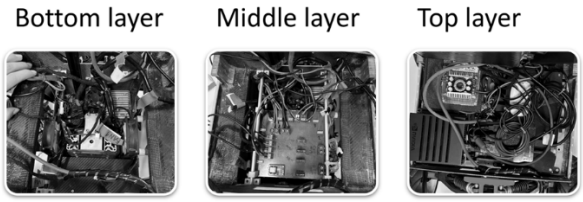

Figure 1: Photo of the three chassis layers

A centrally mounted Robitronic Platinium Brushless Motor 1/8 10.5T (Robitronic Electronic Ges.m.b.H., n.d.) transmits torque through a gearbox and motor shaft to the front and rear differentials, powering the drivetrain. This system ensures balanced traction and stable movement across varied terrain. Each of the four wheels is individually controlled by servo motors, enabling precise maneuverability and advanced turning maneuvers in structured maize rows. The independent steering system enhances adaptability, allowing Carbonite to efficiently navigate tight field layouts and adjust its path dynamically.

Electronics and Power Distribution

The electrical system is distributed across two primary levels within the chassis. The middle level houses a custom power distribution board, responsible for regulating power from a lithium polymer battery to all robotic components. A DC-DC converter ensures a stable 12V supply for the servo motors, while integrated fuses protect the system from short circuits caused by electrical faults. The board also manages power distribution for the sensors, ensuring uninterrupted operation.

Sensors and Navigation

To achieve fully autonomous field operations, Carbonite is equipped with a comprehensive sensor suite. Two Sick Tim 571 LiDAR (SICK AG, n.d.) sensors, mounted at the front and rear, generate high-resolution 2D point clouds that allow the robot to detect obstacles and precisely determine the structure of maize rows. An Intel RealSense™ Depth Camera D455 (Intel Corporation, n.d.), mounted on a mast, captures detailed visual data, enabling accurate plant and weed recognition. A BNO055 Gyroscope (Bosch Sensortec GmbH, n.d.) monitors the robot's orientation and movement, ensuring stable navigation across uneven terrain. Additionally, a WiFi-Bridge ASUS EA-AC87 AC1800 (ASUSTeK Computer Inc., n.d.) facilitates real-time communication, allowing for remote monitoring and operation in open fields. For a better understanding of the robot, we have installed a WS2812B LED strip (Worldsemi, n.d.) around it, which signals to the farmer what state the robot is in and what it is currently doing.

System for Targeted Application of Water, Fertilizer, and Herbicides

Beyond autonomous movement, Carbonite is designed to enhance precision agriculture through its targeted liquid application system. Equipped with an integrated custom-built tank and a dual-pump (Keenso, n.d.) mechanism, the robot can precisely distribute water, fertilizers, or herbicides via side-mounted nozzles.

General software structure

To navigate and manage behavior, our robot uses Robot Operating System (ROS) 1, version noetic (Open Robotics, 2024b) with a modular software concept using ROS-Actions (Open Robotics, 2024a). There are two types of ROS-Actions: Actions servers and action clients. In our case, we mostly use action servers. An action server can accept a goal and will then try to reach it. During this process it can continuously send feedback, for example if there are delays in reaching the goal. After it either reached the goal or had to cancel it, it returns a result. Action clients can send goals to action servers, receive feedback and the results. This gives us flexible, reliable, and modular software for most processes on the robot. Another advantage is that an action client can request an action server to terminate its activities, so we can stop the robot any time if necessary. We also use a graphical user interface (GUI) made with Qt5 (The Qt Company, n.d.) to set parameters during runtime (see Fig. 2). The different software parts can access the parameters via ROS services

and can adapt accordingly. This makes parameter adjustment and testing on the field faster and more flexible.

For each task the robot should be able to perform, we have a task specific action server, each containing an individual state machine. When the software is started and after initially starting all needed action servers, the state machine in the task specific action server administrates the behavior of the robot. The task specific action server is provided with the input from the game pad and the GUI, with which we can also drive the robot or change its state manually. To see the current state of the robot while it is performing a task, we use the LED strips on the front and back of the robot.

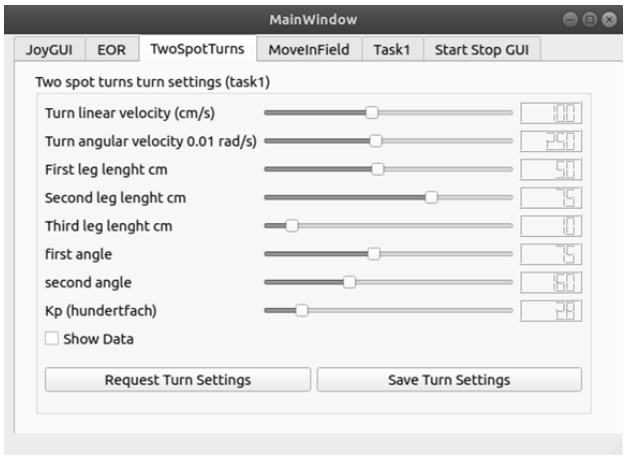

Figure 2: GUI for turn settings

Algorithm for basic navigation through maize rows

Our navigation software is structured into several action servers: The MoveInField action server handles driving between maize rows. Individual Turn action servers are responsible for executing different types of turns to change rows. The DetectEndOfRow action server identifies the end of a maize row.

When the start button on the game pad or the GUI is pressed, the task specific action server sends a goal to the MoveInField action server (arrow 1 in Fig. 3) to start the navigation. After a distance set in the GUI is driven, MoveInField sends a goal to the DetectEndOfRow action server (arrow 2 in Fig. 3) to start searching for the end of the row while it still navigates through the row. This delay before starting the search for the end of the row minimizes the risk of misidentifying a gap in the row as the end of the row and additionally saves us compute load. We mainly use values between 5 and 10 meters, depending on the length of the rows. This assumes that the row has at least a certain minimum length. As soon as the end of the row is found, DetectEndOfRow sends a result back to MoveInField (arrow 3 in Fig. 3). MoveInField then terminates (arrow 4 in Fig. 3) and the task specific action server sends a goal to the Turn action server (arrow 5 in Fig. 3), which then performs a turn to either the left or the right. It then sends a result back to the task specific action server (arrow 6 in Fig. 3), restarting the cycle (see Fig. 3). The cycle is repeated until the task has been completed.

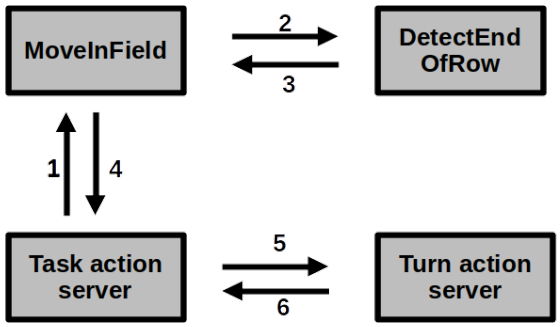

Figure 3: Sketch of carbonite's software structure

The MoveInField action server

In the following, the x-axis describes the direction the robot drives if it would not steer, and the y-axis describes the direction perpendicular to the x-axis to the left and right of the robot (see Fig. 4).

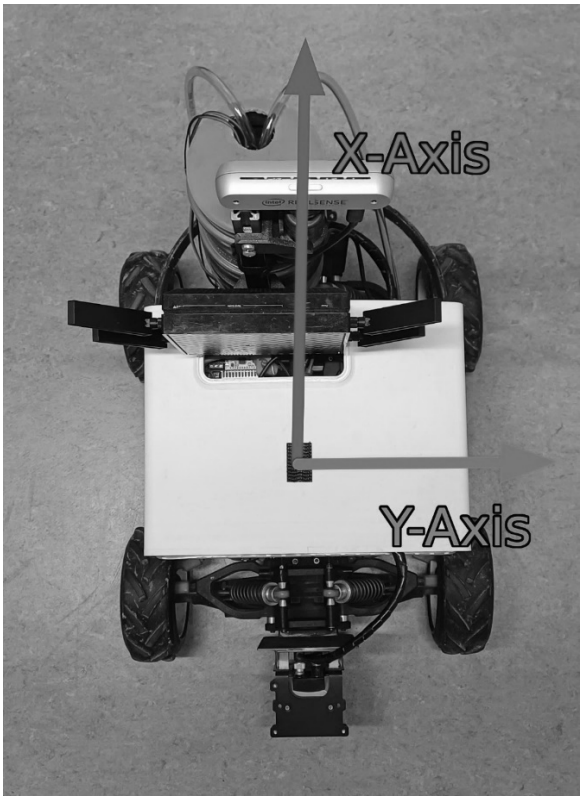

Figure 4: Photo of Carbonite with described axes

The MoveInField action server navigates our robot using its laser scanners. The laser scanner data is first converted from its published format ("sensor_msgs/LaserScan" ROS message type) to a point cloud format ("pcl_ros/point_cloud" ROS message type) using the Point Cloud Libary (PCL) (Rusu & Cousins, 2011) to facilitate algorithm processing. The resulting point cloud data is stored in a circular buffer, which continuously merges the most recent point clouds into one point cloud containing the points of all recent point clouds. This ensures that temporary gaps in the data, for example when the scanner points upwards while driving over a rock, do not disrupt the algorithm.

Afterwards the point cloud data is cropped to a small rectangle in front of the robot to save computing power and to only consider data points that are interesting for navigation. Inside this rectangle, we now search for the optimal direction to drive through the rows. For this, only the y-coordinates of the point cloud's points are taken, which shows the offset of the points to the left and right to of the robot. Now we apply a function (see Fig. 5) that gives each point a weight, depending on how well it fits the standardized row distance. The sum of all those points shows how well the current function matches the point cloud. If we start to calculate those scores for multiple offsets of the function, we can find the function offset with the highest score, which is the direction to the center of the current row in the field. With this, we know in which direction the robot must steer to stay in the row. We set the parameter of the standardized distance between two maize rows to 78 cm.

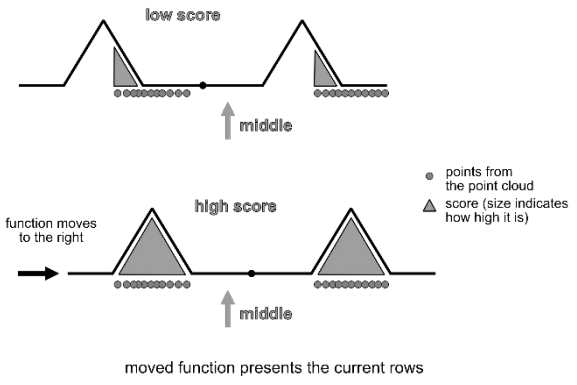

Figure 5: Illustration for the scores for navigation

The DetectEndOfRow action server

The DetectEndOfRow action server also uses the same conversion for the laser scanner data to the point cloud format that is described in the beginning of the MoveInField action server section. To detect the end of a row, we cut the space in front of the robot into multiple stripes parallel to the y-axis, along the x-axis. If the stripe contains more than a defined number of points, it is labeled as "occupied." In our system, the distance to the end of the row is estimated as twice the average distance of the occupied stripes from the robot. When this calculated distance is low, or if no stripes are detected as occupied, the *DetectEndOfRow* action server concludes its operation and signals this to the *MoveInField* action server. The algorithm regarding the number of stripes, the point count threshold for a stripe to be considered occupied, and the distance at which the end of the row is announced has been empirically adjusted for our robot by repeatedly testing and tuning it until we were satisfied.

Turn action servers

Since our robot has two laser scanners, one at the front and one at the back, we can drive forwards and backwards. For turns into adjacent rows, we use this to our advantage by performing a y-turn at the end of the row: Instead of turning the robot by 180 degrees, we simply drive backwards into the next row.

When rows must be skipped, simple U-turns are used. For this we use the average row width multiplied with the number of rows to skip to calculate how far we need to drive in the headland before turning back into the field. To drive straight in the headland, we use a gyro sensor. We read the sensor's yaw-value and then adapt the driving direction with a simple p-controller algorithm. The target value for this algorithm is calculated with the robot's yaw-value during the last few meters in the row + 90°.

Precision Irrigation Through Intelligent Gap Detection

To maximize irrigation efficiency, our robot utilizes a custom-designed tank and spray system that applies liquid directly to the maize plants. This system, combined with our specially developed algorithms, enables us to distribute water with optimal precision, ensuring minimal waste and maximum effectiveness in the field.

To avoid wasting resources, our system is designed to recognize gaps in plant coverage and automatically adjust spraying accordingly. For our gap recognition, we again use the point cloud from our laser scanner data. To detect gaps, we analyze the y-dimension, which runs laterally (left–right) relative to the robot. Each point is assigned a weight proportional to its absolute y-value, representing its lateral distance from the robot. The smaller the distance, the higher the weight. All weights are then summed up to estimate the presence of plants alongside the robot.

To determine whether a gap is present, the algorithm compares the current sum of weights with two dynamic thresholds. These thresholds are computed as fixed percentages of the moving average of the weight sums from

the most recent measurements. If the current sum of weights falls below a gap detection threshold, it is interpreted as a gap. In this case, the spraying process is interrupted. If the sum exceeds an end of gap detection threshold, the system assumes sufficient plant presence, and spraying is resumed.

This dynamic thresholding mechanism enables robust gap detection under varying environmental and crop conditions. By continuously adapting to the recent distribution of plant density, the algorithm remains sensitive to changes in the field while avoiding false detections caused by short-term fluctuations. The use of separate thresholds for stopping and resuming irrigation further ensures stable system behavior by preventing rapid switching between states.

All parameters are customizable using the GUI and must be adjusted for the specific environment, task and robot steering behavior. We do so by repeatedly testing the robot in the specific environment and for the specific tasks and tuning the parameters until we are satisfied with the performance.

The object recognition

For many agricultural tasks, it is essential that the robot can detect objects in the camera images. To fulfil this requirement, Artificial Intelligence (AI) is used. To be able to perform the tasks 3 (Field Robot Event, 2023d, p. 3) and 4 (Field Robot Event, 2023e, p. 4) of the FRE, we trained an image recognition model which can detect humans and deer. Like in the FRE competition 2022, we used customized YOLO (Redmon & Farhadi, 2018; Wang et al., 2022) models for AI image recognition. We trained and inferenced the models using Darknet (Redmon, 2013). Darknet is an open-source library for convolutional neural networks (CNNs) written in C and CUDA. In our work, we used two versions of the library: the original version developed by Joseph Redmon (Redmon, 2013) (hereinafter referred to as the original version) and a fork created by Alexey Bochkovskiy (Bochkovskiy, 2013) (hereinafter referred to as the forked version). The forked version, as opposed to the original version, includes performance improvement and additional features, a metrics plot during the training process and support for Microsoft Windows.

We used the ROS package darknet-ros (Bjelonic, 2016) to integrate the trained AI models into our robot control software. In darknet-ros, the original version of Darknet is used. The darknet-ros package, which inferences the trained AI models, runs on a Nvidia Jetson AGX Xavier Developer Kit (NVIDIA Corporation, n.d.) using CUDA. Darknet-ros receives the images via ROS messages from the camera and publishes the detection results as a ROS topic. During our development and testing, we could use configuration and model files created by one Darknet version with the other without compatibility issues, if only functionality supported by both versions is used. This ROS topic is subscribed to by our robot control software, which then controls the robot based on the detection result.

The models were trained using the forked version and transfer learning. Transfer learning means that a model is trained using weights from an existing, pre-trained model. The main advantage of transfer learning is that the duration of the training is significantly shorter and significantly less training data is required, as opposed to training a model from scratch with random weights, which requires large amounts of data and computational power. For the YOLO models, transfer learning can be done by reusing the convolutional weights of the models provided by the developers of Darknet (Bochkovskiy, 2013; Redmon, 2013) and YOLO (Redmon & Farhadi, 2018; Wang et al., 2022). For the above-mentioned reasons and because the approach worked well for the AI for the FRE competition 2022, we decided to build our AI using that approach. To teach the YOLO model to recognize humans and deer, we needed a labeled dataset containing images of humans and deer. Because we could not find a dataset online which fulfilled our requirements, we decided to compile our own dataset using images we considered suitable from public sources, mainly from the Microsoft Common Objects in Context (COCO) (Lin et al., 2015) dataset and Google's Open Images Dataset (Benenson et al., 2019; Kuznetsova et al., 2020). Some of the images included labels, while we had to label other images ourselves.

For our initial training run, we trained the model with the method described above and a dataset consisting of 50% images containing humans and 50% images containing deer. The resulting AI model then had an issue of incorrectly detecting humans or deer in images that contain neither of them.

To fix this problem, we added images not containing either human or deer. This causes the AI to learn that it is possible for images to not contain any object, solving the problem of the first run. The adapted dataset consisted of 25% images showing humans, 25% images showing deer, and 50% images showing other objects (neither human nor deer).

Another problem was still present after our dataset refinement: The AI also detected humans standing outside the field (e.g. spectating the field) instead of the relevant object in front of the robot. Our first idea to solve this problem was tailored to the FRE tasks, where the relevant objects were images printed on paper. It involved training a second YOLO model to detect paper and then programming our post-processing logic to disregard all objects not on paper. To build a paper detection model, we synthetically generated a dataset consisting of images without paper and images where perspective projections of rectangles (simulating paper) were added. The addition of images

without paper was necessary to prevent incorrect detection of paper on every input image. However, in addition to not being applicable for real-world application, the approach did not work due to many other white surfaces (e.g. tents, buildings, walls) being incorrectly detected as paper, making the filtering based on paper detection useless. Even after adding images containing other white surfaces, the model still was not reliable enough.

Thus, we instead decided to utilize the depth sensor of the Intel Real Sense camera. Before the camera image is forwarded to the AI, all pixels outside a specific distance range are filtered out by making them black. This approach successfully prevented our AI system from recognizing irrelevant objects, limiting image recognition to relevant objects in front of the robot.

Use of AI to improve usability

We developed a program for the robot that supports giving the robot instructions using voice in natural language, allowing easy programming of the robot without technical knowledge. We used OpenAI's API Platform (OpenAI, n.d.) as AI provider. First, the program records the voice command and transcribes it using the Whisper API. Then, the text is passed along with a command prompt to the ChatGPT API, which converts the natural language instruction to machine-readable instructions for the robot path. The command prompt describes the machine-readable driving path format used in Task 1 of the FRE (Field Robot Event, 2023b, p. 1). After ChatGPT outputs a valid driving path, the robot starts following the desired path.

RESULTS AND DISCUSSION

Our robot's performance was evaluated through participation in the Field Robot Event 2023 at the University of Maribor, Slovenia (Field Robot Event, 2022). The competition consisted of five tasks designed to test the robot's capabilities. That year, 14 other robots also participated in that competition (Field Robot Event, 2023a).

Task 1 (Field Robot Event, 2023b, p. 1) assessed basic navigation skills. The robots were required to autonomously navigate a maize field following a specific pattern, as shown in Fig. 6. During the 3-minute runtime, Carbonite successfully covered nearly 9 out of 10 rows, securing first place in this task. Our main advantage over other teams were Carbonite's lightweight and compact design, which allowed the stable and reliable MoveInField algorithm to make small deviations from the optimal course without damaging the plants, and our fast y-turn behavior for adjacent rows. Second place was secured by FRED from Technische Universität Braunschweig, while the Wageningen Robotic Bulls Eye team finished in third place (Field Robot Event, 2023a).

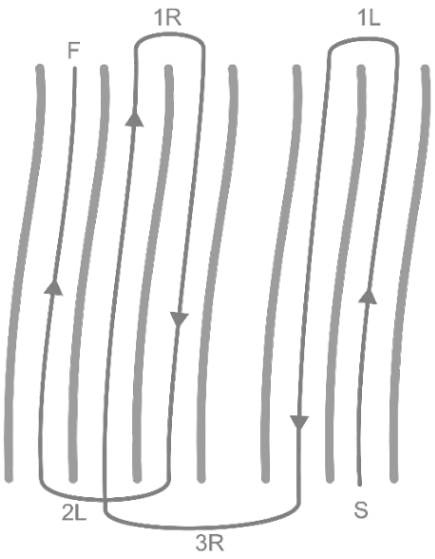

Figure 6: Example path for the robot

Task 2 (Field Robot Event, 2023c, p. 2) involved driving through every second row of the maize field while spraying water from an onboard tank to cover gaps in the row. Despite a significant delay at the start, likely due to network issues that prevented the initial start command from executing correctly, the robot began operating normally after approximately 30 seconds. Our gap detection algorithm proved to be reliable and precise, ensuring accurate spraying. The small nozzles on our robot allowed us to conserve water, enabling continuous operation throughout the 3-minute runtime without running out of water. Our decision to reduce the driving speed paid off, as it resulted in stable and consistent performance for the navigation. This allowed us to secure first place in this task as well, ahead of Wageningen Robotic Bulls Eye in second place and FRED from Technische Universität Braunschweig in third place (Field Robot Event, 2023a).

Task 3 (Field Robot Event, 2023d, p. 3) required identifying images of deer and humans displayed on signs placed in front of the robot. Our AI performed well overall but struggled with edge cases, such as an image of an animal resembling a deer that was from the "unknown" category and a drawn deer, which was misclassified as "unknown." This issue likely stemmed from our AI being primarily trained on photos of real deer. To improve accuracy in future events, we plan to create a more diverse training dataset. Despite these challenges, we finished in fourth place behind KAMARO Betelgeuse from Karlsruhe with first place, TH OWL on second place, and a tie between Grasslammer from Milan, Team Acorn from Osnabrück, and DTU Maizerunners from TU Denmark on third place (Field Robot Event, 2023a).

Task 4 (Field Robot Event, 2023e, p. 4) involved recognizing images and responding by either waiting or reversing into the next row. Our last-minute solution was to stop the robot upon detecting an obstacle. This allowed us to cover the first row, turn, and shut down before the obstacle. Since many teams faced difficulties with this task, our simple approach was enough to secure sixth place. The task was won by DTU Maizerunners from TU Denmark, followed by FRED from Technische Universität Braunschweig and Team FloriBot from Hochschule Heilbronn (Field Robot Event, 2023a).

With two first, one fourth and one sixth place we could secure the first place overall followed by Wageningen Robotic Bulls Eye on second place and FRED from Technische Universität Braunschweig on the third place (Field Robot Event, 2023a).

The freestyle task (Task 5) (Field Robot Event, 2023f, p. 5) allowed us to develop and present a custom solution to a jury and was not considered for the overall ranking. Our presented integration of AI to improve usability secured sixth place. The task was won by Wageningen Robotic Bulls Eye, with KAMARO Betelgeuse from Karlsruhe in second place and Team Acorn from Osnabrück in third (Field Robot Event, 2023a).

CONCLUSION

Carbonite managed to secure the first place in the navigation and plant treatment tasks and won the competition. Our main advantages were the light and small form factor that allowed us to drive fast and left room for errors, the well tested and tailored navigation algorithm and the y-turn that saved us a lot of time in turns. However, there is still room for improvement. We were unable to implement the required state machine behavior for Task 4 in time, primarily due to a rigid code structure and insufficiently abstracted robot control. Currently, for example, the system requires manual differentiation between forward and reverse driving, which means that commands like "drive left" must be adjusted depending on the driving direction. Future improvements will focus on abstracting these inputs and outputs so that the algorithm, such as the MoveInField action server, can execute the same commands consistently, regardless of the robot's driving direction.

Acknowledgment

We want to thank our sponsors, Mikro Makro Mint, the Wilhelm Stemmer Stiftung, Sick AG and our institution, the Schülerforschungszentrum Südwürttemberg e.V. at our high school Gymnasium Überlingen, for supporting us and making our participation in the FRE possible. Furthermore, we want to thank our supervisor, Lukas Locher, for his time and support of Team Carbonite's FRE participation. We also want to thank all current and previous team members for creating a strong foundation for us to build on.

REFERENCES

1. ASUSTeK Computer Inc. (n.d.). *EA-AC87—Support*. Retrieved March 29, 2025, from https://www.asus.com/supportonly/eaac87/helpdesk_knowledge/
2. Benenson, R., Popov, S., & Ferrari, V. (2019). Large-scale interactive object segmentation with human annotators. arXiv:1903.10830. https://doi.org/10.48550/arXiv.1903.10830
3. Bjelonic, M. (2016). *Leggedrobotics/darknet_ros* [C++]. Robotic Systems Lab - Legged Robotics at ETH Zürich. https://github.com/leggedrobotics/darknet_ros
4. Bochkovskiy, A. (2013). *AlexeyAB/darknet* [C]. https://github.com/AlexeyAB/darknet
5. Bosch Sensortec GmbH. (n.d.). *Smart Sensor BNO055*. Bosch Sensortec. Retrieved from https://www.bosch-sensortec.com/products/smart-sensor-systems/bno055/
6. Cardinale, B. J., Duffy, J. E., Gonzalez, A., Hooper, D. U., Perrings, C., Venail, P., Narwani, A., Mace, G. M., Tilman, D., Wardle, D. A., Kinzig, A. P., Daily, G. C., Loreau, M., Grace, J. B., Larigauderie, A., Srivastava, D. S., & Naeem, S. (2012). Biodiversity loss and its impact on humanity. *Nature*, *486*(7401), 59–67. https://doi.org/10.1038/nature11148
7. Field Robot Event. (2022, October 21). *Field Robot Event 2023 in Slovenia! – Field Robot Event*. https://fieldrobot.nl/event/index.php/2022/10/21/field-robot-event-goes-to-slovenia/
8. Field Robot Event. (2023a). *FRE2023-RESULTS_FINAL.pdf*. Field Contest 13.06.2023 – 15.06.2023 FRE 2023 Results. Retrieved from: https://fieldrobot.nl/event/wp-content/uploads/2023/06/FRE2023-RESULTS_FINAL.pdf
9. Field Robot Event. (2023b). *Task 1 Navigation – Field Robot Event*. Retrieved from: https://fieldrobot.nl/event/index.php/contest-hybrid/tasks-h/
10. Field Robot Event. (2023c). *Task 2 treating (spraying) the plants – Field Robot Event*. Retrieved from: https://fieldrobot.nl/event/index.php/contest-hybrid/task-h1/
11. Field Robot Event. (2023d). *Task 3 sensing and recognizing possible obstacles – Field Robot Event*. Retrieved from: https://fieldrobot.nl/event/index.php/contest-hybrid/task-h2/
12. Field Robot Event (2023e). *Task 4 Static and dynamic obstacles – Field Robot Event*. https://fieldrobot.nl/event/index.php/contest-hybrid/task-h3/

13. Field Robot Event (2023f). *Task 5 Freestyle – Field Robot Event*. Retrieved from: https://fieldrobot.nl/event/index.php/task-5-freestyle/
14. Foley, J. A., Ramankutty, N., Brauman, K. A., Cassidy, E. S., Gerber, J. S., Johnston, M., Mueller, N. D., O'Connell, C., Ray, D. K., West, P. C., Balzer, C., Bennett, E. M., Carpenter, S. R., Hill, J., Monfreda, C., Polasky, S., Rockström, J., Sheehan, J., Siebert, S., Tilman, D., & Zaks, D. P. M. (2011). Solutions for a cultivated planet. *Nature*, *478*(7369), 337–342. https://doi.org/10.1038/nature10452
15. Gil, G., Casagrande, D. E., Cortés, L. P., & Verschae, R. (2023). Why the low adoption of robotics in the farms? Challenges for the establishment of commercial agricultural robots. *Smart Agricultural Technology*, *3*, 100069. https://doi.org/10.1016/j.atech.2022.100069
16. Intel Corporation. (n.d.). *Introducing the Intel® RealSense™ Depth Camera D455*. Intel® RealSense™ Depth and Tracking Cameras. Retrieved from: https://www.intelrealsense.com/depth-camera-d455/
17. Keenso. (n.d.). *Mini-Wasserpumpe, 12 V DC, 6 W, Tauchpumpe, ohne Bürste, energiesparend, für Aquarium, Brunnen, kleine Fischteiche, Solarsystem: Amazon.de: Haustier*. Retrieved from: https://www.amazon.de/Submersible-Without-Energy-Aquarium-Fountain/dp/B07VGQ8KJV
18. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., & Ferrari, V. (2020). The open images dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, *128*(7), 1956–1981. https://doi.org/10.1007/s11263-020-01316-z
19. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). Microsoft COCO: Common Objects in Context. *arXiv:1405.0312*. https://doi.org/10.48550/arXiv.1405.0312
20. Malhi, G. S., Kaur, M., & Kaushik, P. (2021). Impact of climate change on agriculture and its mitigation strategies: A review. *Sustainability*, *13*(3), 1318. https://doi.org/10.3390/su13031318
21. Mayer, J., Fauser, K., Schupp, J., & Locher, L. (2022). Carbonite–Team Carbonite. *Proceedings of the 18th Field Robot Event 2021*, 51–57. Retrieved from: https://www.fieldrobot.com/event/wp-content/uploads/2022/02/Proceedings_FRE2021.pdf
22. Nelson, G. C., Rosegrant, M. W., Koo, J., Robertson, R. D., Sulser, T., Zhu, T., Ringler, C., Msangi, S., Palazzo, A., Batka, M., Magalhaes, M., Valmonte-Santos, R., Ewing, M., & Lee, D. R. (2009). *Climate change: Impact on agriculture and costs of adaptation*. International Food Policy Research Institute. https://doi.org/10.2499/0896295354
23. NVIDIA Corporation. (n.d.). *Jetson AGX Xavier Developer Kit by NVIDIA | 945-82972-0045-000*. Arrow.Com. Retrieved March 29, 2025, from https://www.arrow.com/en/products/945-82972-0045-000/nvidia
24. Open Robotics. (2024a, January 9). *actionlib—ROS Wiki*. Retrieved from: https://wiki.ros.org/actionlib
25. Open Robotics. (2024b, August 31). *noetic—ROS Wiki*. https://wiki.ros.org/noetic
26. OpenAI. (n.d.). *Overview—OpenAI API*. Retrieved from: https://platform.openai.com
27. Redmon, J. (2013, 2016). *Darknet: Open Source Neural Networks in C*.
28. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv:1804.02767*. https://doi.org/10.48550/arXiv.1804.02767
29. Robitronic Electronic Ges.m.b.H. (n.d.). *Robitronic Platinium Brushless Motor 1/8 10.5T*. Robitronic RC Car Online Shop - Power for Winners. Retrieved from: https://shop.robitronic.com/en/robitronic-platinium-brushless-motor-r03204
30. Rusu, R. B., & Cousins, S. (2011). *3D is here: Point Cloud Library (PCL)*. IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China. Retrieved from: https://github.com/PointCloudLibrary/pcl
31. SICK AG. (n.d.). *TiM571-2050101—TiM | SICK*. Retrieved from: https://www.sick.com/us/en/catalog/products/lidar-and-radar-sensors/lidar-sensors/tim/tim571-2050101/p/p412444
32. The Qt Company. (n.d.). *Qt 5.15*. Retrieved from: https://doc.qt.io/qt-5/
33. Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv:2207.02696*. Retrieved from: https://doi.org/10.48550/arXiv.2207.02696
34. Worldsemi. (n.d.). *WS2812B.pdf*. Retrieved from: https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf

Napredne navigacijske tehnike in postopki umetne inteligence: Ključne strategije ekipe Carbonite na tekmovanju Field Robot Event 2023

IZVLEČEK

Eden od pristopov za reševanje trenutnih izzivov v kmetijstvu, ki jih povzročajo podnebne spremembe, naraščajoče svetovno prebivalstvo in izguba biotske raznovrstnosti, je precizno kmetijstvo, pri katerem igrajo robotski sistemi ključno vlogo. Tekmovanje poljskih robotov Field Robot Event (FRE) 2023 je zato izzvalo študentske ekipe, da razvijejo in izboljšajo avtonomne kmetijske robote. V tem prispevku predstavljamo izboljšave našega poljskega robota »Carbonite«, ki ga razvijamo na Schülerforschungszentrum (SFZ) Südwürttemberg. Naša lahka in kompaktna zasnova robota, podprta z naprednim in učinkovitim navigacijskim algoritmom, je omogočila hitro premikanje robota po polju. Poleg tega pa smo uvedli novo razvit sistem za ciljno in natančno uporabo vode, gnojil in herbicidov, ki temelji na naprednem algoritmu za zaznavanje prisotnosti rastlin, kar preprečuje nepotrebno porabo virov. Prav tako smo pripravili postopke umetne inteligence za prepoznavanje objektov, ki temeljijo na modelih You Only Look Once (YOLO), kar robotu omogoča ustrezno odzivanje glede na vrsto ovire. Carbonite je tako osvojil prvo mesto tako v nalogi navigacije kot tudi v nalogi obdelave rastlin, k čemur je prispevala lahka zasnova in posledično visoka hitrost premikanja robota, vse to pa je pripomoglo k skupni zmagi skupine na FRE 2023.

Ključne besede: kmetijska robotika, precizno kmetijstvo, trajnost, umetna inteligenca