Daffodil Robot Navigation

An AI/ML-Powered Computer Vision Application for use with a Custom Agricultural Robot

Authors: Matt Watkins, Daniel Zhang, Ben Green, Jamie Covell, Josh Beckett & Maya Rogers

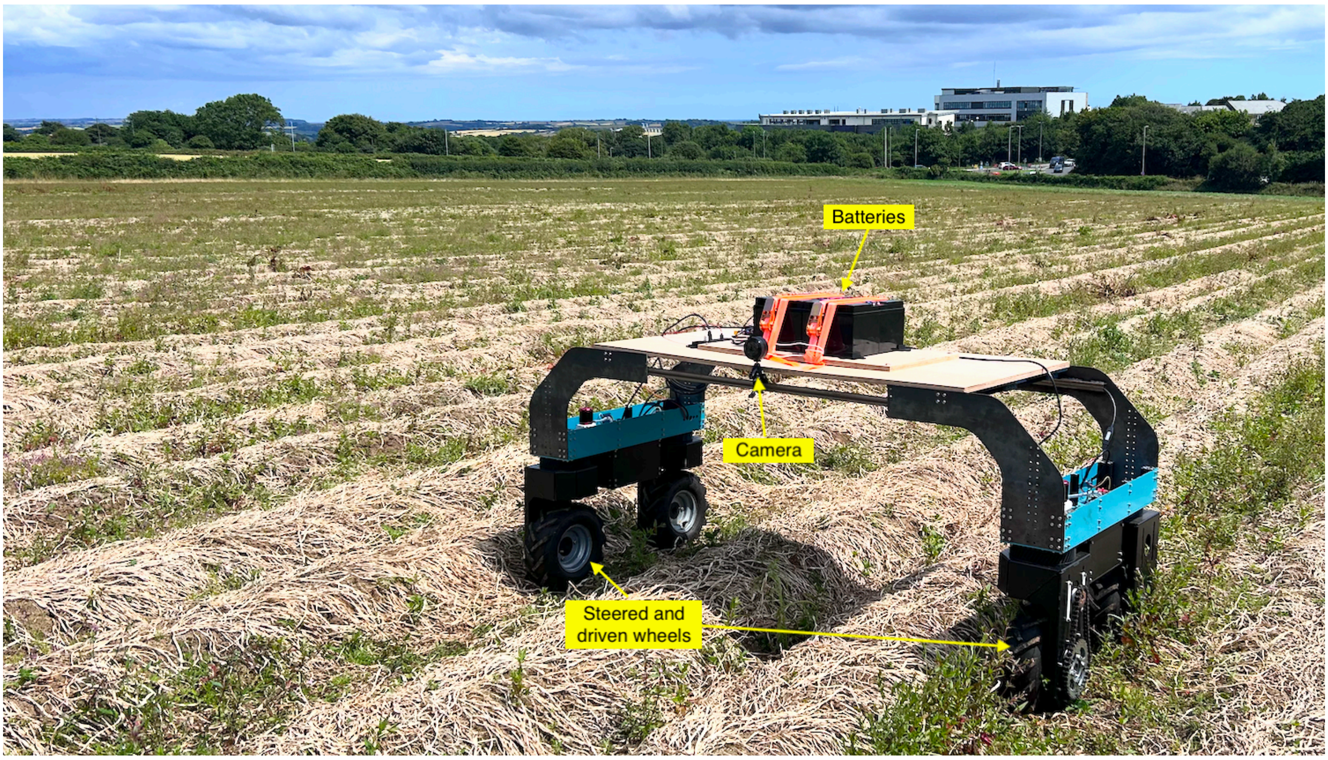

Figure 1: Robot in Field

Introduction

This project is a feasibility study to develop the next generation of 'smart tools' to help vegetable and ornamental farmers increase productivity and reduce waste.

This research builds upon work started 18 months ago to develop a daffodil-classifying and picking robot for use in growing fields. The current phase focuses on a specific subset of tasks within the broader project.

A critical challenge for agricultural robots is navigation—specifically, maintaining a steady course along furrows and executing precise turns to cover adjacent rows across an entire field. This research aims to develop a system utilising:

- 4-wheel synchro and independent drive for stability, precise furrow navigation, and tight turning
- Machine vision using cameras to guide path planning along furrows and during turns
- Supervised to train the robot in categorising and responding to various furrow conditions and turning scenarios
- GPS to log routes and enhance navigation accuracy
- Control methods to ensure smooth movement and precise positioning

Hypotheses

1. A wheeled robot can learn to traverse furrows while maintaining consistent positioning over the growing area without causing disturbance
2. A wheeled robot can navigate from one furrow to an adjacent one by utilising landscape features and GPS guidance
3. A wheeled robot can successfully navigate an entire test field, executing all necessary turns while staying within the furrows

The primary aim of the project is to build a proven autonomous vehicle for rural applications, initially focusing on supporting the needs of a local agricultural partner. An additional aim is to strengthen knowledge exchange between robotics and AI researchers within the computing subject area.

The project is being conducted by a team of researchers and student interns from the University of Falmouth during the summer of 2025.

Team

1. **Project Lead** - Matt Watkins
2. **Robotics Lead** - Ben Green
3. **AI/ML Lead** - Dr Daniel Zhang
4. **Technical/Programming** - Jamie Covell
5. **Student Interns/Researchers** - Josh Beckett & Maya Rogers
6. **Administration** - Michelle Glover

Robot Development

Overview

The aims of the robot development was to create a robot suitable for the specific conditions of navigating a daffodil field and to provide power and a drive system that can be integrated into a machine learning model so the robot effectively learns to navigate the features of the field.

An existing version of the agricultural robot was already available to the project. However, unlike other agricultural settings, daffodils are grown in double ridges (see figure 2). To meet the requirements of this growing environment, the robot's span needed to be increased from its original 1200mm to 2000mm.

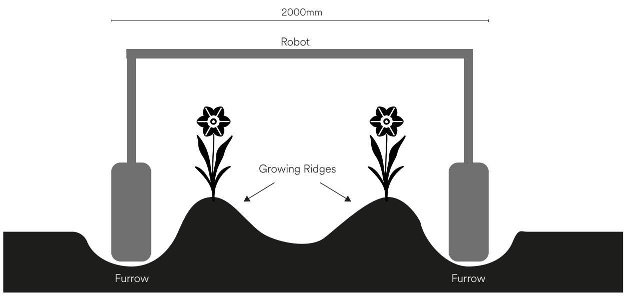

Figure 2: Cross section of the growing environment

Due to the short project delivery time and the long lead times for component orders and fabrication it was determined that the project would have 2 development tracks:

1. Reconfigure the existing robot (version 1) to have the necessary width/span to move along a row and so that a camera could be mounted on front of the adapted version for data gathering in the field to start training an AI model to achieve a proof of concept (version 1.2)
2. Begin the procurement and design process with the plan being to have the wider 4 wheel drive version 2 ready for training with the model at the end of the project.

Version 1.2

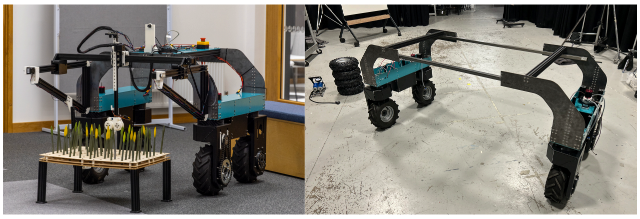

Figure 3: Figure 3: Version 1 and Version 1.2 of the robot after the width increase

This version of the robot maintains many of its existing features but the crossbars were cut in half and metal rods were welded between the 2 halves to create a rough prototype of the version 2 robot.

Version 1.2 like its predecessor is **non-holonomic**[4] only **2 wheel drive** with **front wheel steer and drive**. This limits it's manoeuvrability. It has 4 motors – 2 controlling steering on either side and 2 providing direct drive to the wheels.

Construction

The robot is made from laser cut anodised steel parts and extruded aluminium for supporting lengths and uprights. It is driven by 4 x Brushless motors, 4 x gearboxes with 2 x ODrive[3] motor controllers and a motorbike spindle and chain to power the wheels. It is powered by 2 x 12volt lead acid batteries (24volts).

Sensors

The robot has a Razer Kiyo Pro **camera** [5] mounted on the front to collect image data, 2 motor **encoders** to measure odometry values and 2 x end stop **limit switches** for steering calibration.

Actuation and Steering

The wheels are accurately controlled using **Ackermann steering angles**[2] which describes how the front wheels of a vehicle should be angled differently when turning to ensure they follow concentric circular paths and minimise tyre wear.

For a vehicle with:

- W = wheelbase (distance between front and rear axles)
- T = track width (distance between left and right wheels)
- R = turning radius to the centre of the rear axle
- θ_i = inner wheel steering angle
- θ_o = outer wheel steering angle

The principle Ackermann relationship is:

$$tan(\theta_i) - tan(\theta_o) = (T/W) \times tan(\theta_i) \times tan(\theta_o)$$

Therefore individual wheel angles are given for the inner wheel as:

$$tan(\theta_i) = W/R$$

and the outer wheel:

$tan(\theta_o) = W/(R+T)$

The inner and outer wheels are denoted by the wheel that is on the inside on a given turn.

The steering angle relationship is expressed as:

$\theta_i - \theta_o = \arctan(W/R) - \arctan(W/(R+T))$

Both steering and throttle is **normalised** to a value between −1 and 1. This is the number that will be fed back into the system once the AI model is trained and the system is a closed loop.

For testing the robot is steered manually using an Xbox controller.

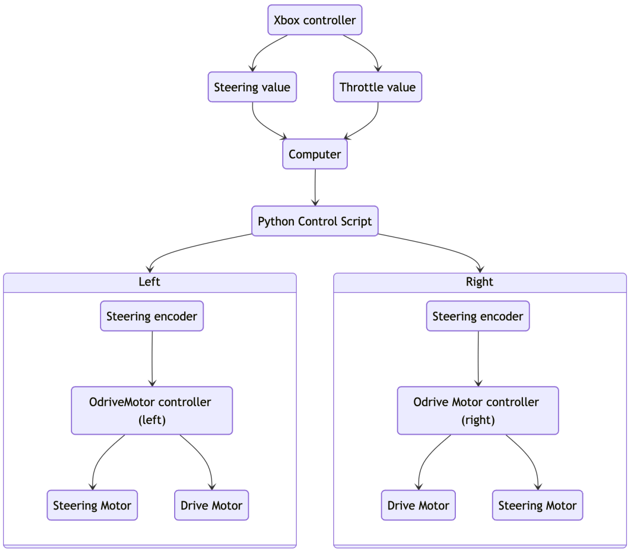

Figure 4: caption

As the above diagram shows the robot is controlled through a python **script** on a laptop **computer** (to be replaced with an Nvidia Microcontroller suitable for realtime AI model training) which instructs the Odrive to turn motors on either side of the robot to turn. The drive motors are controlled through the Xbox controllers throttle value and the Steering motor takes data from the specific encoder to determine its current position and the new change in position based on the steering value from the Xbox controller.

Software Libraries

- Odrive for Python: https://docs.odriverobotics.com/
- OpenCV in Python: https://opencv.org/

Version 2

This version of the robot will be **holonomic**[4] meaning it can move independently in any direction, including forward, backward, sideways, and rotating in place, without needing to turn its body first.

This is also referred to as **4 wheel synchro steering**. This can best be described as a shopping trolley where all wheels can steer and drive independently.

This will increase the torque of the robot and also allow it manoeuvre in any direction. It will be able to make 90° turns effectively moving sideways (crab like) between rows.

Machine Learning Development

Overview

The vision part of the Daffodil Robot research project aims to investigate the integration of artificial intelligence (AI) and machine learning (ML) in agricultural robotics, with a particular emphasis on computer vision systems for autonomous operation. It also seeks to develop a modular and extensible framework for robotics computer vision algorithms and systems, which can be employed to design and evaluate novel methods and approaches for agricultural robots capable of perceiving and interpreting their environment through AI/ML-powered computer vision.

This section presents details of the development process and preliminary experimental results, focusing exclusively on the computer vision aspect of the project.

The Approach

Software Libraries

- OpenCV in Python: https://opencv.org/
- PyTorch: https://pytorch.org/
- YOLO (You Only Look Once): https://pjreddie.com/darknet/yolo/

The vision system of the project utilises OpenCV, PyTorch, and YOLO. OpenCV is a widely used computer vision library for image processing and related tasks. To enable the robot to learn from its environment on the daffodil farm, the PyTorch library and the YOLO package are employed for training and making predictions based on images captured by the robot's mounted camera. PyTorch is a prominent deep learning framework for constructing neural networks, while YOLO is a real-time object detection system based on convolutional neural networks (CNNs).

The Workflow

To meet the project requirements, the workflow for the vision component is divided into two stages: offline and online. The offline stage involves collecting and preprocessing video and image data, training the YOLO model, and evaluating algorithms for applying morphological operations to refine masks and lines, as well as calculating the ridge angle. The online stage involves deploying the trained YOLO model to the robot, capturing live video, performing YOLO-based detection and prediction on the images, and returning

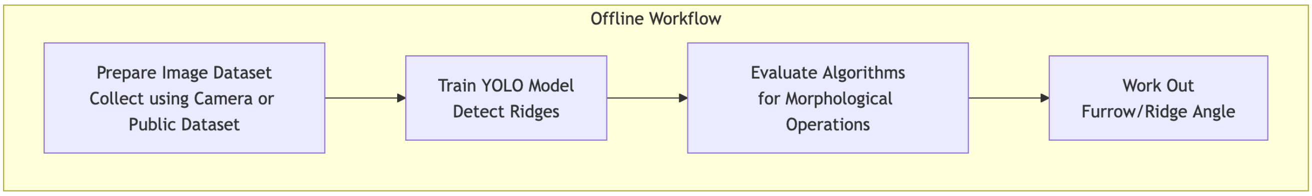

Figure 5: Online Workflow

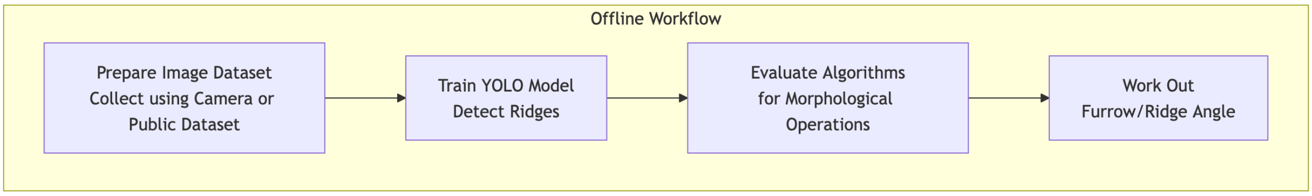

Figure 6: Offline Workflow

the calculated ridge angle. This navigation angle is then used to control the robot's motors, enabling it to steer and move along the furrow.

The offline and online workflows of the process are illustrated in the following diagrams:

Data Collection

The team conducted several experiments to verify the feasibility of the workflows. The first experiment focused on the offline workflow, in which the team collected and preprocessed an image dataset by collecting data using images taken from a handheld camera in one of Varfell's daffodil fields.

Note on images: The research was conducted in the summer months so the ridges contain the dead daffodil plants which appear as a light straw colour. In the winter these areas will contain green growing daffodil shoots and brown earth in the furrows.

The YOLO model was trained on the collected data, and algorithms for morphological operations were evaluated. The results indicated that the YOLO model can detect furrows and ridges in images of sufficient quality (see **Figure 3**). The algorithms for morphological operations were also tested and found to be effective and robust in cleaning masks and detecting ridge lines.

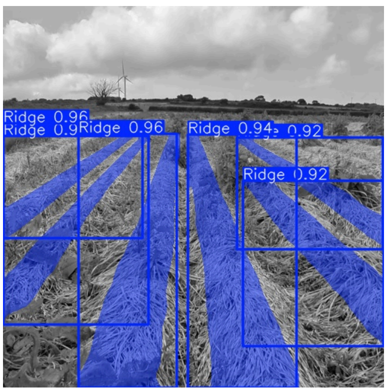

Figure 7: caption

Once the robot had been assembled and tested, the team undertook several field trips to the collaborator's daffodil farm in Falmouth to collect data using the mounted camera and to test the system in real-world conditions. The collected data comprised captured images of furrows, ridges, and backgrounds while the robot was operating on the farm. The YOLO model was subsequently trained on this real-world dataset, and the algorithms for morphological operations were evaluated.

Although the YOLO model is capable of processing images at a high frame rate, the robot's steering does not require such a rate. The camera was therefore configured to capture images at a rate of 5 frames per second, representing an effective compromise between frame rate and storage requirements.

The camera was mounted centrally at the front of the robot, which was manually controlled via a joystick. After two visits to the farm, the team compiled a dataset of 1,500 images, which was then divided into multiple subsets for manual tagging prior to training the YOLO model. The experimental results are presented in the following sections.

Results

The resolution of the images captured by the camera is 640×480 pixels, whereas the public dataset used in previous experiments contains images of 512×512 pixels. This section presents the results of experiments conducted with both the public dataset and the real-world data.

Manual Dataset Experiment

As shown in **Figure 5**, the YOLO model can accurately detect ridges in the public dataset. If multiple ridges are detected, post-processing steps are required to retain only the two central ridges, which guide the robot's movement.

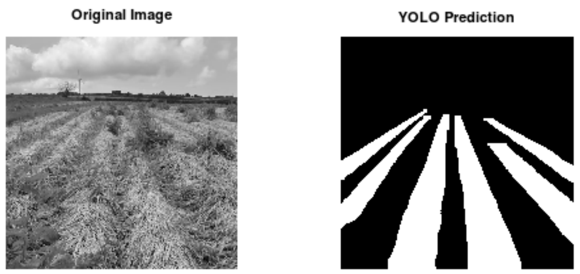

Figure 8: YOLO predicted masks of ridges

The masks corresponding to the two central ridges typically contain more white pixels than those of the other four ridges on either side. The positions of these masks are also taken into account to enhance the robustness of the algorithm for retaining only the central ridges, as shown in **Figure 6**.

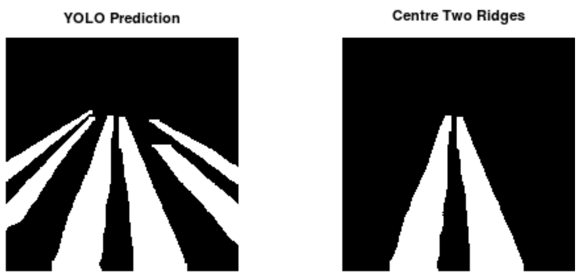

Figure 9: Post-processing to keep the centre two ridges

Once the mask map of the ridges of interest has been created, the distance transform[^1] can be applied to calculate the distance from each pixel to the nearest zero pixel in the source image. According to the documentation for the function **cv::distanceTransform**, the parameters were set to `distanceType=DIST_L2` and `maskSize=5` to achieve a more accurate estimation of distances, as shown in **Figure 7**.

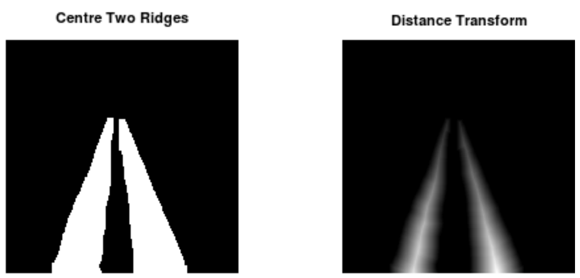

Figure 10: Distance transform of the centre two ridge masks

Subsequently, the centrelines can be extracted as the local maxima in the distance transform result using the **cv::dilate** function, with a structuring element of size 2×3 as the kernel, as shown in **Figure 8**.

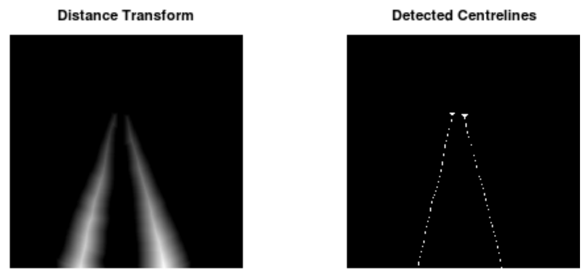

Figure 11: Detected centrelines as the local maxima in the distance transform result

The final step involves applying the Hough Transform (**cv::HoughLines/cv2::HoughLinesP**) to the detected centrelines in order to fit the ridge lines and calculate the ridge angle, as shown in **Figure 9**. The final results, where the detected ridge lines have been fitted and overlaid on the original image, are shown in `strong`

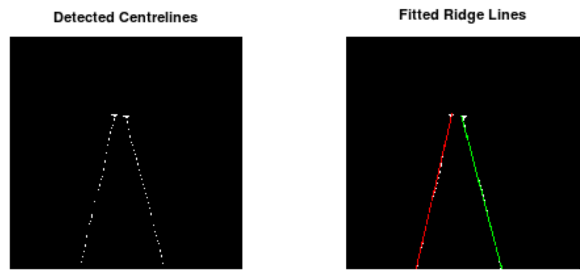

Figure 12: Fitted ridge lines using Hough Transformation

The final results demonstrate that the YOLO model can accurately detect ridges in the public dataset, and that the post-processing steps effectively retain only the two central ridges. The distance transform and Hough Transform were employed to calculate the ridge angle and, consequently, the navigation angle.

Figure 13: Detected ridge lines fitted and overlaid on the original image

Alongside the vector lines angle data was also generated from the model which was available to be subsequently used to control the steering of the robot. Below is a sample data point in JSON:


```
{
  "GoodData\\rawImage_00107_1426": {
    "0x": 0.361328125,
    "0r": 0.4377913036415464,
    "1x": 0.626953125,
```

```
    "1r": -0.1163551652068748
  },
```

`0x` is the normalised position of the first line at the bottom of the image `0r` is the angle (in radians) of the first line at the bottom of the image `1x` is the position of the second line at the bottom of the image and `1r` is the angle (in radians) of the second line at the bottom of the image.

Embodied Data Experiment

The same workflow described above was applied to the real-world data collected from the daffodil field from a camera embodied in the robot itself, which is intended to enable the robot to learn from the environment and navigate the field using the calculated ridge angle.

Figure 14: Robot gathering data in test field

Figure 15: Robot gathering data in test field

Dataset

Throttle and **angle** data was also gathered alongside images. In this sample JSON file you can see how this was formatted.

```
{
    "throttle": 7.333333333333333,
    "angle": -0.00164794921875,
    "image_path": "./data_1426/
rawImage_00118_1426.png"
}
```

The throttle indicates the amount of power applied to the motors, and the angle represents the steering input from the manual Xbox controller at the moment when the image is captured (every 500 milliseconds).

At the end of **2 separate runs** in the field with the robot the team collected **2000 images** of which approximately 1200 were usable to train the model.

Issues

Data collection

However, several issues were identified in the real-world data, as shown in **Figure 13** and **14**.

By comparing the results from the manual handheld dataset and the real-world data, the following issues were identified:

- The real-world images are blurry, suggesting that the camera may not be properly calibrated or that the auto-focus mechanism causes the camera to adjust focus continuously.
- The real-world data contains more noise and exhibits varying lighting conditions, which can lead the YOLO model to mis-classify the ridges.
- The ridge line detection lacks robustness, as the lines are not always well-defined when large gaps exist between segments.

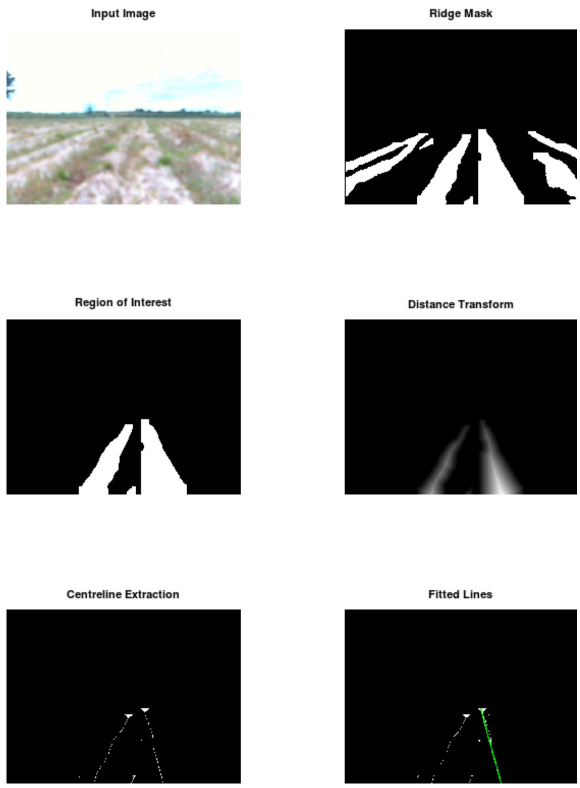

Figure 16: Experiment steps for the Real-world data collected from the daffodil farm

Figure 17: Fitted ridge lines on the Real-world data collected from the daffodil farm

Robot Function

Robustness is a consideration for future development. The robot performed 2 runs of the field and on both occasions it failed to traverse the whole row of the field due to mechanical or electrical failure:

1. First run — a 3D printed part of the drive assembly snapped. This was resolved with a redesign and a denser print.
2. Second run — a short circuit occurred possibly due to a current overload (see Figure 15). This resulted in the destruction of a motor controller. This was despite a circuit breaker designed and implemented to detect these issues and protect vital equipment.

Figure 18: Short circuit on the robot during a run

Each of these were system critical malfunctions that meant that the experiment had to be stopped.

Future Work

As a proof-of-concept, short-term research project, the team successfully built an agricultural robot for the daffodil farm with an AI/ML-powered computer vision system. Given the limited time and resources, the team prioritised tasks to focus on the most critical ones.

Robotics

The robot's dimensions and components are suitable for the environment and the steering and power is sufficient for it to navigate the terrain, However issues need to be addressed to provide sufficient robustness for more tests and to gather sufficient data to train the model and to deploy to the same robot.

- Version 2 will have **4 wheel drive** (8 motors) that will ensure it manoeuvres more effectively, and that it can deliver more torque as all 4 wheels will provide drive to the robot.
- Due to power issues with the original batteries taken from version 1, version 1.2 had new **batteries** mounted on the top and in the centre (see figure 1). This was due to an increase in size of the newer batteries which meant they couldn't be placed in their original battery holder locations. This raised the centre of the gravity and made the robot top heavy. In version 2 this will be addressed by moving the position of the batteries to either side near the wheels lowering the centre of gravity and making the robot more stable.
- **4 Wheel synchro** will provide more nimble and flexible movement allowing the robot to effectively move laterally to the next row in the field.
- The **camera** used has an auto focus and image stabilisation feature which resulted in blurry images (see figure 12 for the change in focus over time). The solution to disable this feature off has been identified for future data collection.
- A reappraisal of **power circuitry** to prevent power surges effecting performance and damaging equipment.

Notes on version 2 progress — Time constraints prevented this parallel track of version 2 development being completed. However, all parts have either been received or have been ordered. Fabrication costs have been added to a purchase order with a local fabrication company. All these purchases have been made within the initial project cost. Time is now needed to finalise the design taking into account the above factors.

Machine Learning

Since the YOLO model has been demonstrated to perform well on the manual dataset, the team will focus on improving model performance on the real-world data. The following steps can be taken to enhance model performance:

- Improve the quality of the collected data, for example by using a stabilised camera and disabling auto-focus.

- Collect additional data under varying lighting conditions and angles.
- Investigate more advanced deep learning techniques, such as transfer learning and data augmentation.
- Explore advanced image processing techniques, including image segmentation and feature extraction.
- Experiment with different Hough Transform parameters, such as the minimum line length and the maximum gap between lines, to improve the accuracy of the ridge line fitting process.

The team will also continue to improve data fusion with other sensors, such as motor data (including throttle and angle) paired with the images. The final results will be evaluated and discussed with the collaborator at the daffodil farm to further enhance the system.

Conclusion

The daffodil robot project has demonstrated the feasibility of integrating AI/ML-powered computer vision in agricultural robotics. The robot did manage to move as expected in the operating environment but robustness was identified as a limiting factor for progressing the project and steps have been identified to resolve these outstanding issues in version 2 of the robot.

The YOLO model, combined with post-processing using distance transform and Hough Transform, was effective in detecting ridges and calculating navigation angles on datasets with good image quality.

Challenges were identified in real-world deployment, including image blur, noise, and variable lighting, which affected model performance. Improving data quality, applying advanced deep learning techniques, and exploring sensor fusion are key steps for enhancing reliability and accuracy.

Only partial proof of hypothesis 1 was possible and hypothesis 2 and 3 still need to be tested.

Overall, the project provides a proof-of-concept framework for autonomous navigation on the daffodil farm and establishes a modular system for further development and testing in agricultural robotics.

References

[1] Pedro Felzenszwalb and Daniel Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004.

[2]D. King-Hele, 'Erasmus Darwin's Improved Design for Steering Carriages–And Cars', Notes and Records of the Royal Society of London, vol. 56, no. 1, pp. 41–62, 2002.

[3] 'ODrive', ODrive. Accessed: Aug. 19, 2025. [Online]. Available: https://odriverobotics.com

[4] H. Goldstein, C. Poole, and J. Safko, Classical Mechanics. San Francisco Munich: Pearson, 2008.

[5] 'Webcam with Adaptive Light Sensor - Razer Kiyo Pro | Razer United Kingdom', Razer. Accessed: Aug. 19, 2025. [Online]. Available: https://www.razer.com/gb-en/streaming-cameras/razer-kiyo-pro