Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 119–129 DOI: 10.15439/2025F9629 ISSN 2300-5963 ACSIS, Vol. 44

Detection and Classification of Rumex Weeds in Grasslands Using YOLOv11

Jorid Holmen, Weria Khaksar
Norwegian University of Life Sciences, Ås, Norway
Email: jorid.holmen@nmbu.no; weria.khaksar@nmbu.no

***Abstract*—This paper explores the use of YOLOv11 and BoT-SORT for detecting and tracking *Rumex obtusifolius* and *Rumex crispus* in grasslands. Two models were developed: Model A trained on the RumexWeeds dataset, and Model B, trained using transfer learning with additional datasets. While Model A performed well on its training data, it struggled in unseen environments. Model B showed improved generalisation, achieving higher performance across diverse conditions and successfully detecting *Rumex longifolius* in Norwegian grasslands.**

Both models were integrated with BoT-SORT and achieved high tracking metrics, supporting GPS-based mapping. Real-time field testing confirmed feasibility, although detection was affected by shadows, terrain, and camera placement.

The results highlight the importance of diverse training data for robust weed detection. Future work should focus on expanding datasets, tuning hyperparameters, and improving hardware for reliable real-world deployment.

Keywords: Weed detection, AI, YOLO, Precision farming, digital agriculture

I. Introduction

THE NEED for sustainable agricultural practices has become increasingly urgent due to environmental challenges, rising input costs, and labour shortages. Traditional weed control methods, especially herbicide use, pose significant ecological risks such as biodiversity loss and water contamination [1], [2], and reducing chemical input is a central objective in EU-wide sustainability strategies [3].

Two very problematic weeds in European grasslands are *Rumex obtusifolius* and *Rumex crispus*, which degrade pasture quality and can negatively affect livestock health [4]. In Norway and other Northern regions, *Rumex longifolius* is also widespread, but remains understudied and absent from open-access datasets [5].

The introduction of deep learning has significantly advanced the field of automated weed detection in agriculture [6]. Several studies have demonstrated promising results using CNNs and YOLO-based models, with applications ranging from UAV mapping to close-range robotic systems [7], [8], [9], [10], [11]. However, these systems often face challenges in generalising across environments, due to variation in lighting, scale, background conditions, and the high cost of collecting annotated training data [6]. Despite these limitations, UAVs and ground robots are becoming increasingly relevant for precision weed control, with successful demonstrations of real-time detection, herbicide application, and object tracking in field settings [12], [13], [14].

Machine learning has enabled progress in automatic dock detection using UAVs and ground robots. For example, Anken et al. [15] used CNNs to detect 90% of *R. obtusifolius*, while Valente et al. [16] achieved reliable UAV-based detection. Güldenring et al. [17] demonstrated successful detection of *R. obtusifolius* and *R. crispus* using YOLOvX. However, models trained on limited datasets often fail to generalise across varying environments, lighting, and species [15], [17].

This paper, part of the SUSDOCK project [18], addresses the lack of data from Northern environments and aims to improve species-specific weed control. The work focuses on detecting dock weeds using deep learning and evaluates generalisation to *R. longifolius* and unseen field conditions.

Main contributions

- Developed a convolutional neural network to detect *R. obtusifolius* and *R. crispus* using open-access datasets.
- Assessed model generalisation to *R. longifolius* and new environments, with and without additional labelled data.
- Explored the use of object tracking and GPS-based mapping to localise dock occurrences.
- Tested the model on a robotic platform to demonstrate feasibility for real-time weed detection.

II. Methodology

This paper follows a structured workflow to ensure a systematic and reproducible approach from data acquisition to analysis. This section outlines the key stages of the process.

The project began by randomly splitting the dataset into training, validation, and test sets. The YOLOv11 object detection model was trained on the training set and validated on the validation set. After training, the model was evaluated on the test set using standard object detection metrics. This model is referred to as *Model A* throughout the remainder of this paper.

To assess generalisation, Model A was also tested on three previously unseen datasets, two of which were annotated. These two labelled datasets were then merged with the original training data and used to retrain the model using the best[1] weights from the initial training. This model will be referred to as *Model B*. This step aimed to explore whether performance could be improved with additional diverse data.

The next stage of the workflow involved tracking and spatial analysis using BoT-SORT, which was applied to dataset

[1]The best weights defined by the Ultralytics implementation during model training.

sequences. This enabled the counting of dock species and mapping of their GPS locations. The tracking performance was then evaluated using established metrics for multi-object tracking. Lastly, Model B was tested on a real-time robotic platform.

A. Software and Hardware

The primary software used in this paper was Python (version 3.9.21) [19], with all scripts written in standard `.py` files.

Due to the computational demands of object detection, local hardware was deemed insufficient. Instead, remote access to the High-Performance Computing (HPC) cluster Orion, provided by NMBU, was used. Orion consists of 1,680 processor cores, 12 terabytes of RAM, and 1 petabyte of storage, accessible via a $10\,\mathrm{Gbit/s}$ network. The operating system is CentOS Linux 7.9. Jobs on Orion were submitted using SLURM (Simple Linux Utility for Resource Management) by creating batch scripts with the `sbatch` command. These scripts define the resource allocation for each job.

B. The Datasets

The primary dataset used to train Model A was the RumexWeeds dataset. Three additional external datasets were used to evaluate the model's ability to generalise to unseen environments. Two of these, the Open Plant Phenotyping Database and the UAV High-Resolution images, were also used for training Model B, to assess if this improved generalisation to new data. An overview of the datasets and their usage is shown in Table I.

RumexWeeds Dataset: The RumexWeeds dataset [17] contains images of *Rumex obtusifolius* and *Rumex crispus*. It consists of 5,510 RGB images with 15,519 manually annotated bounding boxes — 81% for *R. obtusifolius* and 19% for *R. crispus*. Data was collected at three locations in Denmark, with two of them undergoing two recording sessions, resulting in five distinct sessions under varying environmental conditions. The recording sessions took place during August, September, and October. Notably, this dataset does not contain *Rumex longifolius*, Norway's most common dock species.

Images were captured using a robotic platform equipped with an RGB camera mounted $1\,\mathrm{m}$ above the ground at a $75°$ angle. Each image has a resolution of 1920×1200 pixels. The robot also carried GNSS, IMU, and odometry sensors, enabling accurate georeferencing and motion tracking.

Open Plant Phenotyping Database: The Open Plant Phenotyping Database [20] was used to evaluate Model A and for training and evaluation of Model B. This public dataset includes 7,590 RGB images representing 47 plant species, all recorded in Denmark during September and October. Of these, 140 images contain *Rumex crispus*, with 6,672 bounding boxes. The plants were grown in containers designed to mimic natural growth conditions. The *Rumex* samples were photographed 1–3 times daily from seedling emergence to full leaf stage. The camera was positioned directly above the boxes at a height of $1.7\,\mathrm{m}$.

UAV High-Resolution Images: The Unmanned Aerial Vehicle (UAV) High-Resolution Images dataset [16] consists of three images captured in Germany in April using a drone at altitudes of $10\,\mathrm{m}$, $15\,\mathrm{m}$, and $30\,\mathrm{m}$. The image captured at $30\,\mathrm{m}$ was excluded due to insufficient resolution for reliably detecting weeds. The images taken at $10\,\mathrm{m}$ and $15\,\mathrm{m}$ were divided into tiles with a resolution of 640×640 pixels. This process resulted in 316 images, with 610 annotated bounding boxes containing *R. Obtusifolius*. As the Open Plant Phenotyping Dataset, this dataset was used to evaluate Model A and to train and evaluate Model B.

Rumex in Norwegian Grasslands: The last dataset consists of 217 unannotated images captured in Norway's various environments, lighting conditions, and camera angles. This is not an open-access dataset, but is provided for this paper through the SUSDOCK project [18]. The images contain mostly *Rumex longifolius*, the most common dock species in Norway. Although the model was trained on other *Rumex* species, *R. longifolius* shares similar characteristics in natural grassland settings. This dataset was used to visually assess whether the model could detect docks in Norwegian environments. Four images will be focused on that both contain *R. longifolius*.

1) Data Preprocessing: YOLOv11 requires input data in the YOLO format, thus the original formats of the RumexWeeds, Open Plant Phenotyping, and UAV High-Resolution datasets were converted accordingly. A `.yaml` configuration file is also required, defining the paths to the images, label files, and a dictionary of class names.

YOLOv11 expects one label file corresponding to each image in the dataset, containing information about the bounding boxes. One bounding box is represented with the class ID, x- and y-coordinates for the centre of the box, and the width and height. There can be several bounding boxes in each annotation file.

The RumexWeeds dataset was randomly split into training, validation, and testing subsets, with 70%, 10%, and 20% allocated to each, respectively. The class distribution was stratified to ensure it was balanced across all splits. For training Model B with new datasets, the training and validation were combined with 80% of the Open Plant Phenotyping data and 80% of the UAV High-Resolution Images into the training set, and the rest of the RumexWeeds dataset was combined for the test set. This resulted in 80% training data and 20% test data. The reason for this change is the limited data on the Phenotype and UAV datasets.

For Multi-Object Tracking, randomly selected images are not suitable; instead, continuous video sequences are required. Therefore, all the sequences from one recording session of the RumexWeeds dataset were turned into one video for each sequence. The videos were annotated with tracking IDs necessary for the MOT metrics, in MOT16 format [21]. Due to the task's time-consuming nature and limited available time, only one recording session was annotated. A total of 580 annotated images were chronologically sorted, with bounding boxes visually matched to their corresponding objects and

TABLE I: Overview of datasets used, with the number of images, bounding boxes, and their intended usage.

Dataset	Images	Bounding Boxes	Annotated	Usage
RumexWeeds [17]	5,510	15,519	Yes	Train Model A and Model B
Open Plant Phenotyping Database [20]	140	6,672	Yes	Validate Model A, train Model B
UAV High-Resolution Images [16]	323	610	Yes	Validate Model A, train Model B
Rumex in Norwegian Grasslands	217	0	No	Validate Model A and Model B

TABLE II: Modified hyperparameters for YOLOv11 training.

Hyperparameter	Default	Modified Value	Reason
`epochs`	100	150	Allows the model more time to converge and potentially improve performance
`batch`	16	8	Smaller batch size can enhance generalisation and reduce overfitting, especially with limited data
`dfl`	1.5	2	Increases the impact of Focal Loss to better address class imbalance

manually assigned tracking IDs.

C. YOLOv11

For object detection, the YOLOv11 was selected. This YOLO version comes in sizes *nano, small, medium, large* and *x large*. Small was chosen for this paper due to its balance between speed and accuracy. The model was utilised through the Ultralytics implementation, which offers a high-level Python API for training, validation, and inference [22].

By default, the Ultralytics implementation uses pre-trained weights from the COCO (Common Objects in Context) dataset, which contains 80 object classes. These weights help improve training efficiency and accuracy when working with custom data. Another default setting is data augmentation. In addition to regular data augmentation, YOLO implements mosaic augmentation.

The default hyperparameters provided by Ultralytics include preprocessing steps such as image resizing and pixel value scaling. Given that hyperparameter tuning is time-consuming and the YOLOv11 creators have already invested significant effort in optimising the defaults, this paper primarily relied on those standard settings. However, some key parameters were adjusted to better align with the dataset's characteristics, as shown in Table II.

Ultralytics also simplifies evaluation by providing built-in support for standard object detection metrics. For this project, the evaluation metrics were inference speed, precision, recall, mAP50, and mAP50-95.

D. BoT-SORT

BoT-SORT was used for object tracking, as it is the default multi-object tracker in the Ultralytics pipeline. BoT-SORT,

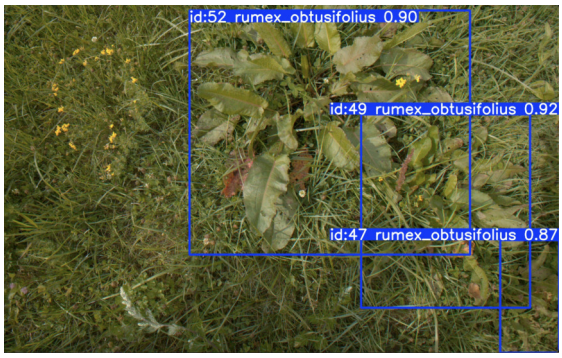


Fig. 1: Extraction from the tracking video, showing a frame with three detected *Rumex obtusifolius* plants, annotated with tracking IDs 47, 49, and 52. The boxes also display class names and detection confidence scores.

with the trained YOLOv11 model as the detection algorithm, was applied to the videos, one for each sequence in the recording session. The output included bounding boxes with unique tracking IDs across frames, forming annotations in MOT16 format and a video visualising the tracked detections. A frame from the tracking video is shown in Figure 1, highlighting how detected objects are assigned consistent tracking IDs.

Tracking IDs were used to associate detected objects with their corresponding GPS coordinates from the RumexWeeds dataset. These locations were visualised using *matplotlib* for static plots and *folium* for interactive maps. The ground truth distribution in the interactive map is shown in Figure 2. When pressing the points in the interactive map, information about what *Rumex* type it is will appear: red points for *R. crispus* and green points for *R. obtusifolius*. In addition, the total number of tracked instances was used to estimate the number of *R. obtusifolius* and *R. crispus* plants.

BoT-SORT was applied to shorter annotated video sequences to evaluate the tracking performance. The output detections in the MOT16 format were compared to the ground truth using the *py-motmetrics* library [23]. A challenge in evaluating tracking is that the tracker may assign different object IDs than those in the ground truth. The evaluation addresses this challenge by mapping the tracking IDs based on Intersection over Union (IoU), requiring a threshold of 0.5 or higher. This ID alignment ensures that metrics such as IDF1 and MOTA accurately reflect tracking performance,

Fig. 2: Ground truth GPS coordinates of dock plants in the RumexWeeds dataset. Each green point represents *R. obtusifolius* and each red point represents *R. crispus*.

Fig. 3: A picture of the robot whilst driving in the field.

rather than being skewed by identity mismatches. The tracking performance was assessed using the three key metrics MOTA, MOTP and IDF1.

E. Real-Time Robotic Platform – A Proof of Concept

To test the feasibility of applying the model in a robotic setting, Model B was selected for deployment. The test was conducted in a field located in Askim, Norway, which contains a high density of *R. longifolius* plants.

The robot was equipped with a Logitech C920s Pro HD webcam, positioned approximately $30\,\mathrm{cm}$ above the ground at an angle of $30°$. The camera has a resolution of 1920×1080 pixels and was connected to a MacBook for simplicity and mobility. A picture of the robot while driving in the field is shown in Figure 3.

The output from the test consisted of a video showing the predicted bounding boxes, along with their confidence scores and assigned tracking IDs. An example of a frame from the video, without any bounding boxes, is shown in Figure 4. Additionally, a text file was generated containing frame-by-frame information, including tracking IDs, bounding box coordinates, and confidence values.

Limitations: Due to limited time and resources, several constraints affected the proof-of-concept test. First, the webcam used was not optimal for field robotics applications, but was selected for its immediate compatibility with the MacBook. Second, the real-time detection code was not fully optimised

Fig. 4: An example frame from the robot during recording.

for the camera settings, leading to performance limitations. Furthermore, the vision system was not physically integrated into the robot's control system, as full hardware integration would have required more development time than the project timeframe allowed. Finally, no GPS module was connected to either the MacBook or the robot, meaning that no spatial localisation data was recorded during the test.

The terrain in the field was bumpy, resulting in the robot's inconsistent driving speed. Due to the camera's mounting position, large portions of the surrounding landscape, including the sky and nearby objects, were captured in many frames. Furthermore, shadows from the robot, the operators, and the low sun position affected the image quality.

However, this is only a proof-of-concept, which means the conditions does not need to be ideal. In spite of these limitations, the tests will still be able to tell the feasibility of the model in a robotic setting.

III. Results and Discussion

A. Object Detection Performance: Model A

Table III shows the evaluation metrics for Model A, trained solely on the RumexWeeds dataset. The model performed well on the training domain, with a high precision of 0.922, a recall of 0.887, and mAP values of 0.949 for mAP50 and 0.703 for mAP50-95. The lower mAP50-95 reflects the model's reduced localisation precision across varying IoU thresholds. On the external Phenotype and UAV datasets, performance declined substantially. While precision remained relatively moderate on the Phenotype data with a value of 0.714, recall dropped significantly to 0.001. The UAV dataset showed poor performance across all metrics. This demonstrates a considerable drop in performance when external data is evaluated. The inference speed is consistent for all three datasets, ranging from $2.761\,\mathrm{ms}$ for the RumexWeeds dataset, $3.450\,\mathrm{ms}$ for the Phenotype dataset and $5.025\,\mathrm{ms}$ for the UAV dataset. The inference was measured on an HPC, which is significantly faster than typical robotic platforms. The Phenotype and UAV datasets showed slower speeds, likely due to more complex images or larger input sizes. These differences should be considered when deploying the model on resource-constrained platforms.

TABLE III: Detection performance of Model A on the validation sets. Results are reported for three datasets: RumexWeeds, Phenotype, and UAV. Metrics include inference speed (ms per image), precision, recall, and mAP50 and mAP50-95.

Dataset	Inference Speed (ms)	Precision	Recall	mAP50	mAP50-95
RumexWeeds	2.761	0.922	0.887	0.949	0.703
Phenotype	3.450	0.714	0.001	0.359	0.198
UAV	5.025	0.015	0.051	0.015	0.009

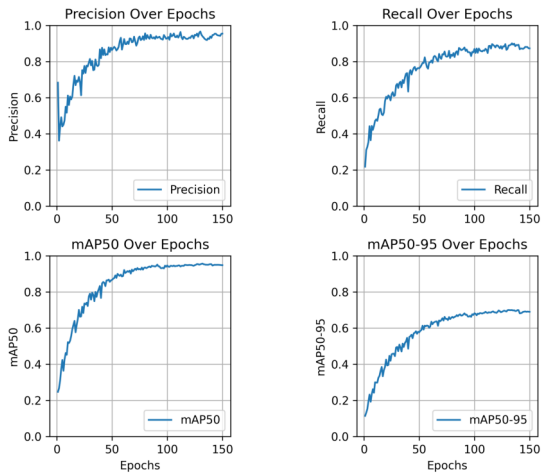
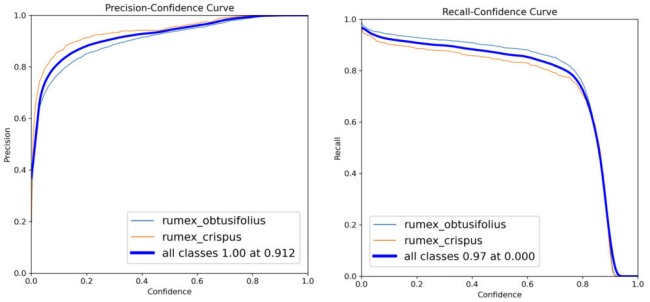


Fig. 5: Training curves for Model A. The plots show the progression of precision, recall, mAP50, and mAP50-95 over 150 epochs on the RumexWeeds dataset. The model converged steadily across all metrics.

Figure 5 presents the training curves for Model A over 150 epochs. The plots illustrate the progression of precision, recall, mAP50, and mAP50-95 throughout training on the RumexWeeds dataset. All four metrics showed a rapid increase during the initial epochs, particularly up to around epoch 50, followed by a more gradual improvement and eventual stabilisation near epoch 100. The curves began at moderate values, with precision, recall, and mAP50 starting between 0.2 and 0.4 suggesting that COCO pretraining provided a strong foundation, while mAP50-95 starts lower, around 0.1. Some fluctuations are observed, likely due to the small batch size, but overall, the trends indicate convergence.

Figure 6a and Figure 6b present the precision– and recall–confidence curves for Model A. The model demonstrated consistently high precision across a broad range of confidence thresholds for both classes, though slightly higher for *R. crispus*. In contrast, recall values were initially high but declined more sharply as confidence increased. This matches the observation of a lower mAp50-95 score, meaning the model prioritised accurate predictions over broader detection coverage, leading to missed detections or less precise bounding boxes at stricter thresholds. Fine-tuning the confidence threshold could improve the balance between recall and precision. The curves followed similar trends for both *R. obtusifolius* and *R. crispus*, with slightly lower recall observed for *R. crispus*, unlike precision.



(a) Precision-confidence curves for Model A.

(b) Recall-confidence curves for Model A.

Fig. 6: Precision and recall confidence curves for Model A, showing performance for *R. obtusifolius*, *R. crispus*, and combined class scores.

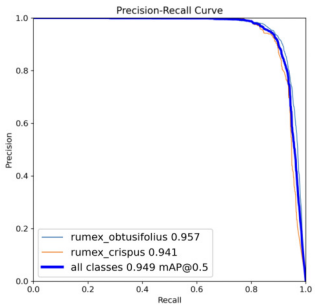


Fig. 7: Precision–recall curve for Model A. The model achieved a high average precision for both *R. obtusifolius* (0.957) and *R. crispus* (0.941), with a combined mAP50 of 0.949.

The corresponding precision–recall curve is shown in Figure 7. The model achieved a combined mAP50 of 0.949 across both target classes. The curve demonstrates that precision remains high as recall increases, particularly for *R. obtusifolius*, which achieved a slightly higher average precision than *R. crispus*, at 0.957 and 0.941 respectively. The curves for both classes followed a similar shape, with minimal divergence across most recall values. The different performances on *R. obtusifolius* and *R. crispus*, is likely due to dataset imbalance, where 81% of annotations were *R. obtusifolius* vs. 19% *R. crispus*. Although focal loss was used to mitigate this, it did not fully offset the imbalance. To improve this, more *R. crispus* images should be annotated, and targeted data augmentation may also help.

While performance on RumexWeeds was strong, Model A's performance dropped significantly on the Phenotype and

UAV datasets. This is likely due to visual domain differences: RumexWeeds images were collected under consistent, ground-based conditions, whereas the external datasets varied in angle, scale, lighting, background, and plant stage. These unfamiliar conditions reduced generalisation. The limited number of *R. crispus* examples further hindered generalisation to new conditions. These results reflect a common deep learning issue: strong performance on training data does not guarantee robustness in new settings. Fine-tuning the model with images better matching target deployment conditions could improve generalisation.

B. Object Detection Performance: Model B

Table IV presents the detection performance of Model B, which was trained using transfer learning on a combination of three datasets. On the combined validation set, the model achieved an inference speed of 2.335 ms, a precision of 0.932, a recall of 0.873, an mAP50 of 0.930, and an mAP50-95 of 0.688. Performance on the RumexWeeds dataset remained strong, with precision, recall, and mAP values comparable to those of Model A. Notably, Model B showed substantial improvements on the external datasets. For example, the Phenotype dataset reached a precision of 0.934, a marked increase compared to Model A. However, when compared to the RumexWeeds dataset, the two new datasets exhibited slightly lower values for recall, mAP50, and mAP50-95, and a higher inference speed.

Model B achieved slightly better mAP50 and mAP50–95 on RumexWeeds than Model A, suggesting that base performance was maintained or improved, partly due to extended training. Still, mAP50–95 scores lagged behind mAP50 across all datasets, indicating that precise localisation remains a challenge.

The inference speed of the combined dataset were similar to the RumexWeeds, likely due to the large proportion of RumexWeeds images. The Phenotype and UAV datasets ran slower at $4.444\,\mathrm{ms}$ and $3.500\,\mathrm{ms}$, respectively. As with Model A, slower speeds may be due to increased image complexity or resolution. Interestingly, UAV was faster than Phenotype in Model B, reversing the pattern from Model A, possibly due to retraining effects or dataset changes.

Figure 8 shows the training curves for Model B over 150 epochs. As with Model A, the plots display the progression of precision, recall, mAP50, and mAP50-95. The values increased rapidly during the early stages of training and stabilised after approximately 50 epochs. The curves started at relatively high values, with precision, recall, and mAP50 beginning between 0.75 and 0.9, while mAP50-95 starts lower, around 0.6. This is typical in transfer learning, where early CNN layers retain useful low-level features. The consistent structure of dock weeds across datasets helped the model learn new features efficiently.

Figures 9a and 9b show the confidence-based precision and recall curves for Model B. In the precision curve, precision remained consistently high across the entire confidence range for both *R. obtusifolius* and *R. crispus*. The recall curve

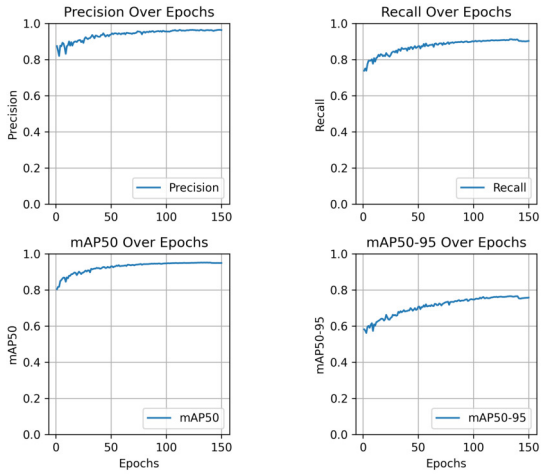
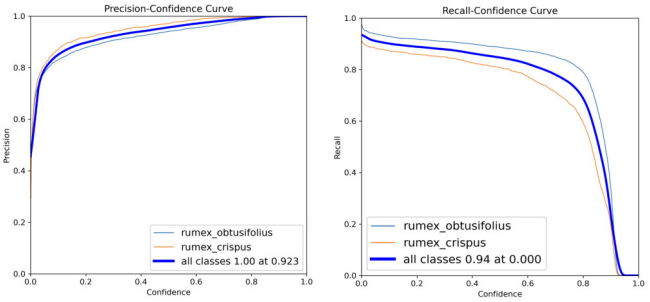


Fig. 8: Training curves for Model B, which was trained using transfer learning on a combined dataset (RumexWeeds, Phenotype, and UAV). The plots show the evolution of precision, recall, mAP50, and mAP50-95 over 150 epochs.



(a) Precision-confidence curves for Model B.

(b) Recall-confidence curves for Model B.

Fig. 9: Precision and recall confidence curves for Model B, showing performance for *R. obtusifolius*, *R. crispus*, and combined class scores.

showed that recall is highest at lower confidence thresholds and decreases steadily as the confidence increases. Recall for *R. crispus* drops more rapidly than for *R. obtusifolius*.

The precision–recall curve in Figure 10 shows that Model B achieves an mAP50 of 0.930. Average precision for *R. obtusifolius* is 0.953, while *R. crispus* reaches 0.907. The class-wise curves follow a similar shape, with high precision across most recall levels. These patterns closely mirror those observed for Model A.

Model B was trained using transfer learning from Model A, with additional labelled data from the Phenotype and UAV datasets. It showed strong detection performance on the combined dataset and improved results on the external datasets compared to Model A. As shown in Table IV, precision, recall, and mAP scores increased significantly on both external datasets, reflecting greater robustness to varied image conditions. This improvement stems from the added data diversity

TABLE IV: Detection performance of Model B on the validation sets. Model B was trained using transfer learning with data from RumexWeeds, Phenotype, and UAV datasets. Metrics include inference speed (ms per image), precision, recall, and mAP50 and mAP50-95.

Dataset	Inference Speed (ms)	Precision	Recall	mAP50	mAP50-95
Combined Data	2.335	0.932	0.873	0.930	0.688
RumexWeeds	2.259	0.946	0.888	0.959	0.733
Phenotype	4.444	0.934	0.765	0.836	0.607
UAV	3.500	0.879	0.775	0.836	0.560

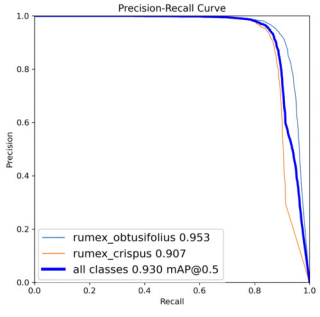


Fig. 10: Precision–recall curve for Model B, trained using transfer learning. The model achieved strong performance on *R. obtusifolius* (0.953) and slightly lower average precision on *R. crispus* (0.907), with a combined mAP50 of 0.930.

and the benefits of transfer learning, where Model A's weights provided a solid starting point.

C. Generalisation to Norwegian Grasslands

Detection results were visualised on images collected from Norwegian grasslands containing mostly *Rumex longifolius* to evaluate how well the models generalise to unseen environments and species. Four images were selected, each shown with predicted bounding boxes from both Model A and Model B.

In the examples, both models identified dock plants in varied settings, including dense vegetation and challenging lighting conditions. Some variation in the number and classification of detections can be observed between the two models. Predictions included both *R. obtusifolius* and *R. crispus* labels.

Figures 11 and 12 show detection results in scenes with visual complexity. These images contain background distractions such as shoes, camera equipment, and uneven lighting, making the detection task more difficult. The *Rumex longifolius* plants are not immediately noticeable even to the human eye. In Figure 11, Model A produced a single prediction in a bright area near a camera leg. In contrast, Model B identified a *R. crispus* leaf, though the prediction has low confidence and is accompanied by a duplicated bounding box. In figure 12 Model A detected a central plant as *R. crispus* with a confidence of 0.69. Model B also identified this plant, but with slightly lower confidence. Additionally, Model B predicted two extra detections with low confidence in areas where no dock plants are visible. It also detected a plant in the upper left with 0.54 confidence, which Model A missed entirely.

TABLE V: BoT-SORT tracking metrics for Model A and Model B. Metrics include Multiple Object Tracking Accuracy (MOTA), Precision (MOTP), and IDF1.

Model	MOTA	MOTP	IDF1
Model A	0.894	0.893	0.883
Model B	0.898	0.89	0.883

The qualitative results from the Norwegian grasslands dataset provide insight into how well the models generalise to completely unseen environments and species. Neither Model A nor Model B was trained on images of *Rumex longifolius*, yet both produced detections on the unlabelled Norwegian images. Model B showed a better overall result. However, both models also displayed false positives, including misclassifications of sunlit areas, plant residues, and patches of grass. This suggests that although the models are capable of transferring some learned features to unfamiliar conditions, their ability to distinguish *R. longifolius* from the background remains limited. The improved responsiveness of Model B indicates that additional training data from varied domains contributes to broader generalisation, but the presence of misclassifications also highlights the need for further adaptation or fine-tuning when deploying such models in new and different environments.

D. Tracking Evaluation Using BoT-SORT

Tracking performance for Model A and Model B was evaluated using the BoT-SORT tracking algorithm. Table V presents the resulting scores across three standard multi-object tracking metrics: MOTA, MOTP and IDF1. The results showed that both models achieved similar performance, with only minor differences observed in MOTA and MOTP, with values between 0.89 and 0.90. The IDF1 score remained identical at 0.883 for both.

To further assess how well the models perform in tracking dock plants over time, the predicted number of detections was compared to the manually annotated ground truth. As shown in Table VI, both models correctly detect five instances of *R. crispus*, while both overestimate the number of *R. obtusifolius* plants by ten. In addition, both models produced a distribution that closely matched the expected locations. Most predictions were concentrated along a path.

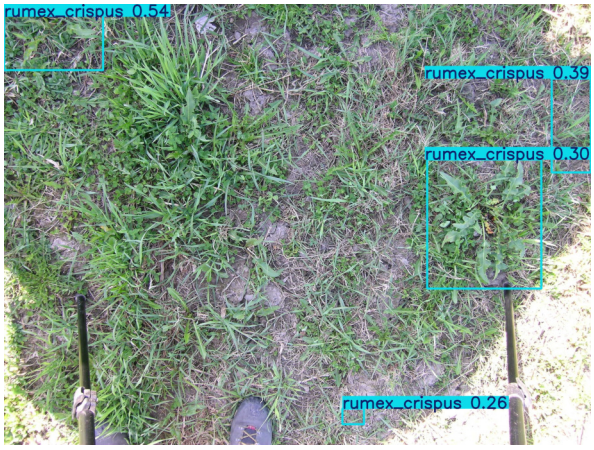
The tracking results using BoT-SORT show that both Model A and Model B maintained high tracking performance across video sequences. This suggests that as long as the object detector provides consistent and confident detections, the tracking algorithm is able to assign and maintain identities effectively.

(a) Model A

(b) Model B

Fig. 11: Detection results in a sparse vegetation scene with visual distractions such as camera equipment and bright lighting. Model A (top) produced one detection near the camera leg, while Model B (bottom) detected a dock leaf with low confidence and overlapping boxes.

(a) Model A



(b) Model B

Fig. 12: Detection results in a visually cluttered scene. Model A (top) identified one dock plant with high confidence. Model B (bottom) detected the same plant and additional low-confidence detections, some of which appear to be false positives.

TABLE VI: Number of dock plants detected by Model A and Model B compared to the manually annotated ground truth.

	Rumex Obtusifolius	Rumex Crispus
Ground truth	41	5
Model A	51	5
Model B	51	5

When comparing the number of tracked detections with the ground truth, both models correctly identified all instances of *R. crispus*, but overestimated the number of *R. obtusifolius*. This overcount likely results from multiple detections on the same plant across frames or slightly offset bounding boxes being treated as separate objects. This observation coincides with the low recall and mAP50-95 values of both Model A and Model B on high confidence thresholds. Since bounding box offset is a contributing factor, this points to potential improvements in the tracking pipeline. Despite these minor inaccuracies, the spatial distribution of tracked detections closely matched the expected GPS coordinates, indicating that the pipeline is suitable for mapping dock presence in the field.

This demonstrates the potential of the combined detection and tracking pipeline for supporting automated weed monitoring and management in real-world farming environments.

E. Real-Time Robotic Platform Performance

Model B was tested in a real-world field environment, resulting in six different video sequences with corresponding text files containing detection information. Table VII summarises the results from each sequence, including the number of frames, the number of unique tracking IDs, the number of unique tracking IDs with average confidence above 0.50 and the number of actual *R. longifolius* plants present in the sequences. The tracking ID number does not correspond well with the ground truth number, due to several false positives.

TABLE VII: Statistics from the six sequences showing information about the number of frames, tracking IDs, and ground truth counts.

Sequence	Frames	Tracking IDs	Average confidence > 0.50	Ground Truth Docks
1	433	28	13	2
2	715	60	34	2
3	607	41	19	3
4	762	17	11	2
5	637	81	53	3
6	708	31	16	8

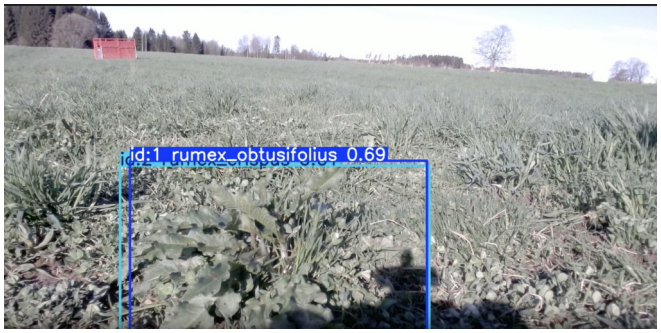


Fig. 13: Frame from sequence 1 showing multiple bounding boxes around a dock and background elements.

In addition to this information, the text files contained a line saying “Coordinates: Location not available" for each detection, meaning it tried to collect the GPS information, but was not able to since there was no GPS module connected.

As illustrated in Figure 13, sequence 1 shows a *R. longifolius* plant with two relatively confident bounding boxes. As the robot moved, an additional bounding box appeared around the same dock. Significant background content, such as the sky, trees, and red farming equipment, is also visible, likely leading to false positives where background objects were misclassified as *Rumex* in later frames. Figure 14 provides an example where a non-*Rumex* object was confidently classified as a dock.

Figure 15 shows a cropped frame from sequence 3, where a *R. longifolius* appears very close to the camera. Only part of the dock is detected, with relatively low confidence. A similar situation is visible in Figure 16 from sequence 5, where the same dock is divided into multiple bounding boxes across different leaves, each with varying confidence levels.

Figure 17 shows two frames from sequence 4. In this situation, the sun is shining directly into the camera, causing strong image diffusion. As a result, the two visible *R. longifolius* plants were not detected at all. A similar issue occurred in sequence 6, where sunlight again affected the camera’s visibility. According to Table VII, sequence 6 generated 31 unique tracking IDs, but only two out of eight actual docks were detected. This pattern, where most docks were missed, is unique to sequences 4 and 6. In contrast, in the other sequences, all docks were detected in some form, although

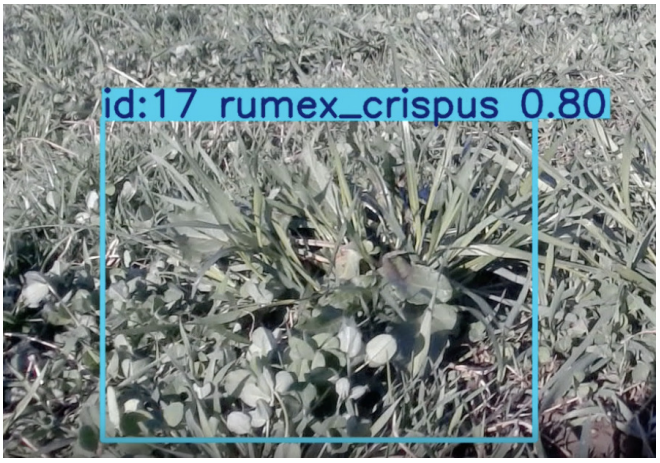


Fig. 14: Frame from sequence 1 showing a non-*Rumex* object incorrectly classified as *Rumex*.

Fig. 15: Frame from sequence 3 showing a close-up of *R. longifolius* with partial and low-confidence detection.

with too many, too few, or poorly placed bounding boxes.

The robotic platform test served as a proof-of-concept to assess whether it would be possible to detect *R. longifolius* in the field using a robot. The overall results were not ideal. However, in cases where the conditions were favourable, such as in Figure 13, the platform successfully detected the dock plants, though with several bounding boxes. This suggests that under improved conditions, the system has the potential to perform significantly better.

Several factors could have contributed to the false positives observed during the field test. The camera on the robot was positioned relatively low, and a higher mounting position would likely have captured a more complete view of the scene. Additionally, tilting the camera further towards the ground could reduce background noise, such as trees and the sky.

Fig. 16: Frame from sequence 5 showing a *R. longifolius* with multiple overlapping bounding boxes.

Fig. 17: Both images show a *R. longifolius* that has not been detected. The frames have become diffused due to the sun.

Güldenring et al. [17] used a camera height of approximately $1\,\mathrm{m}$ and an angle of $75°$, which appeared to be more effective. Another challenge was that the test was conducted when the sun was relatively low in the sky, causing strong shadows and uneven lighting. Capturing images closer to midday would likely improve lighting conditions. Finally, the use of a non-specialised camera and detection code that was not fully optimised for the hardware may also have contributed to the reduced detection performance.

In addition to false positives in detection, a high number of unique tracking IDs were observed. The ground surface was uneven and textured, causing the robot to move unpredictably across the grassland. The BoT-SORT algorithm predicts object movement to maintain consistent tracking IDs. However, the irregular movement of the camera likely made it difficult for the tracking algorithm to generate stable and meaningful tracking results. In future applications, using a larger or wider robot platform could help reduce camera instability and improve tracking accuracy.

Lastly, the absence of GPS coordinates meant that mapping dock occurrences in the field was not possible. However, the proof-of-concept demonstrated that it would be feasible to collect GPS data alongside detection results if such data were available. This indicates a promising potential for mapping dock occurrences in future applications.

IV. Conclusion and Further Work

This paper explored the use of YOLOv11 and BoT-SORT for detecting and tracking dock weeds in grasslands, focusing on improving generalisation across different environments and species. Two models were trained and tested: Model A, trained only on the RumexWeeds dataset, and Model B, which used transfer learning with additional datasets to improve robustness.

The results showed that Model A performed very well on the RumexWeeds dataset but struggled to generalise to new environments, such as the Open Plant Phenotyping Database and UAV High-Resolution Images. Model B, trained with additional data, improved performance on these external datasets while maintaining high accuracy on the original RumexWeeds data. Both models detected *R. longifolius* in images from Norwegian grasslands, with Model B performing slightly better. These findings demonstrate that adding more diverse training data is an effective way to improve the generalisation of deep learning models for weed detection.

The tracking results showed that both Model A and Model B achieved high scores across all evaluated tracking metrics. This indicates that the combined detection and tracking system worked reliably for counting and mapping dock weeds. However, challenges such as slightly inaccurate bounding boxes and overcounting suggest that further improvements to detection precision and tracking stability are needed.

Testing the system in real-time using a robotic platform showed that it is possible to detect *R. longifolius* plants under field conditions, although the results were not ideal. Factors such as strong shadows, a low camera angle, background distractions, and an uneven ground surface likely affected detection accuracy. These results highlight that hardware setup and environmental conditions are critical factors when applying the model outside controlled environments. Despite these challenges, the proof-of-concept showed promising potential for real-time robotic weed detection in future applications.

Further Work: The promising results of this paper show that the system has strong potential and should be developed further. Based on the findings discussed above, several specific areas for improvement have been identified that could further strengthen the system.

First, the project would benefit greatly from expanding the training datasets. If the focus remains on *R. obtusifolius* and *R. crispus*, it would be essential to collect additional images of *R. crispus* to better balance the class distribution. In addition, given the emphasis on Norwegian grasslands, creating a large, open-access dataset specifically for *R. longifolius* would be highly valuable. Another idea worth exploring is training *R. crispus* and *R. obtusifolius* as a single class, as done by Güldenring et al. [17]. Since both species are targeted for removal in the same way, merging them into one detection class could simplify the classification task and possibly improve the model's ability to detect *R. longifolius* as well.

Model B was trained using the same hyperparameters as Model A for simplicity. Future work could investigate tuning the hyperparameters specifically for Model B, as this may further improve performance, particularly when training on more varied data.

For the robotic platform, it would be beneficial to implement the improvements suggested in the discussion, such as optimising camera position and movement stability. In addition, designing a camera flash solution that provides consistent lighting, similar to the one used by Kilter [13], could help reduce issues caused by varying weather and lighting conditions during field operations.

Acknowledgment

This work is a part of the SUSDOCK project funded by the research council of Norway.

REFERENCES

[1] A. Van Bruggen, M. He, K. Shin, V. Mai, K. Jeong, M. Finckh, and J. Morris, "Environmental and health effects of the herbicide glyphosate," *Science of The Total Environment*, vol. 616-617, pp. 255–268, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0048969717330279

[2] A. Klik and J. Rosner, "Long-term experience with conservation tillage practices in austria: Impacts on soil erosion processes," *Soil and Tillage Research*, vol. 203, p. 104669, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167198720304517

[3] J. Wesseler, "The eu's farm-to-fork strategy: An assessment from the perspective of agricultural economics," *Applied Economic Perspectives and Policy*, vol. 44, no. 4, pp. 1826–1843, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aepp.13239

[4] S. Hejduk and P. Dolezal, "Nutritive value of broad-leaved dock (rumex obtusifolius l.) and its effect on the quality of grass silages," *Czech Journal of Animal Science*, vol. 49, no. 4, pp. 144–150, 2004. [Online]. Available: https://cjas.agriculturejournals.cz/artkey/cjs-200404-0003.php

[5] P. E. Hatcher, L. O. Brandsaeter, G. Davies, A. Lüscher, H. L. Hinz, R. Eschen, and U. Schaffner, "Biological control of rumex species in europe: opportunities and constraints." *CABI*, p. 470–475, 2008. [Online]. Available: https://doi.org/10.1079/9781845935061.0470

[6] J. Zhang, F. Yu, Q. Zhang, M. Wang, J. Yu, and Y. Tan, "Advancements of uav and deep learning technologies for weed management in farmland," *Agronomy*, vol. 14, no. 3, 2024. [Online]. Available: https://www.mdpi.com/2073-4395/14/3/494

[7] J. Zhao, T. W. Berge, and J. Geipel, "Transformer in uav image-based weed mapping," *Remote Sensing*, vol. 15, no. 21, 2023. [Online]. Available: https://www.mdpi.com/2072-4292/15/21/5165

[8] E. C. Tetila, B. L. Moro, G. Astolfi, A. B. da Costa, W. P. Amorim, N. A. de Souza Belete, H. Pistori, and J. G. A. Barbedo, "Real-time detection of weeds by species in soybean using uav images," *Crop Protection*, vol. 184, p. 106846, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0261219424002746

[9] P. Wang, Y. Tang, F. Luo, L. Wang, C. Li, Q. Niu, and H. Li, "Weed25: A deep learning dataset for weed identification," *Frontiers in Plant Science*, vol. Volume 13 - 2022, 2022. [Online]. Available: https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2022.1053329

[10] Y. Mu, R. Feng, R. Ni, J. Li, T. Luo, T. Liu, X. Li, H. Gong, Y. Guo, Y. Sun, Y. Bao, S. Li, Y. Wang, and T. Hu, "A faster r-cnn-based model for the identification of weed seedling," *Agronomy*, vol. 12, no. 11, 2022. [Online]. Available: https://www.mdpi.com/2073-4395/12/11/2867

[11] T. W. Berge, T. Torp, F. Urdal, and M. Vallestad, "Sensor technology for precision weeding in cereals: Evaluation of a novel convolutional neural network to estimate weed cover, crop cover and soil cover in near-ground red-green-blue images," Norwegian Institute of Bioeconomy Research (NIBIO), Ås, Norway, NIBIO Report 8(134), 2022. [Online]. Available: https://nibio.brage.unit.no/nibio-xmlui/handle/11250/3031834

[12] T. Jin, K. Liang, M. Lu, Y. Zhao, and Y. Xu, "Weedssort: A weed tracking-by-detection framework for laser weeding applications within precision agriculture," *Smart Agricultural Technology*, vol. 11, p. 100883, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772375525001169

[13] T. Utstumo, F. Urdal, A. Brevik, J. Dørum, J. Netland, Overskeid, T. Berge, and J. Gravdahl, "Robotic in-row weed control in vegetables," *Computers and Electronics in Agriculture*, vol. 154, pp. 36–45, 11 2018.

[14] Kilter Systems, "Kilter systems – ai-powered agricultural robotics," 2024, accessed: 14 April 2025. [Online]. Available: https://www.kiltersystems.com

[15] T. Anken and A. Latsch, "Characteristics of a spot sprayer for the treatment of rumex obtusifolius in meadows," *agricultural engineering.eu*, vol. 78, no. 3, 2023. [Online]. Available: https://www.agricultural-engineering.eu/landtechnik/article/view/3295

[16] J. Valente, S. Hiremath, M. Ariza-Sentís, M. Doldersum, and L. Kooistra, "Mapping of rumex obtusifolius in nature conservation areas using very high resolution uav imagery and deep learning," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, p. 102864, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843222000668

[17] R. Güldenring, F. K. van Evert, and L. Nalpantidis, "Rumexweeds: A grassland dataset for agricultural robotics," *Journal of Field Robotics*, vol. 40, no. 6, pp. 1639–1656, 2023.

[18] "Susdock: Sustainable control and mapping of dock plants," https://www.ri.se/en/susdock, accessed: 2025-04-22.

[19] Python Software Foundation, *Python Language Reference, version 3.9.21*, 2023. [Online]. Available: https://docs.python.org/3.9/

[20] S. L. Madsen, S. K. Mathiassen, M. Dyrmann, M. S. Laursen, L.-C. Paz, and R. N. Jørgensen, "Open Plant Phenotype Database of Common Weeds in Denmark," *Remote Sensing*, vol. 12, no. 8, p. 1246, Apr. 2020. [Online]. Available: https://www.mdpi.com/691100

[21] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," 2016. [Online]. Available: https://arxiv.org/abs/1603.00831

[22] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," 2024. [Online]. Available: https://arxiv.org/abs/2410.17725

[23] C. Heindl, Toka, and J. Valmadre, "py-motmetrics: Python implementation of metrics for multiple object tracking," https://github.com/cheind/py-motmetrics, 2024, accessed: 2025-03-21.